PUBLIC
SAP HANA Platform 2.0 SPS 08
Document Version: 1.0 – 2025-04-22

SAP HANA Administration Guide for SAP HANA Platform

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

Content

1 SAP HANA Administration Guide. 8

2 Administration Information Map. 9

3 Database Administration Tasks at a Glance. 13

4 SAP HANA System Architecture Overview. 15

4.1 Server Architecture of Tenant Databases. 15

Tenant Databases. 17

The System Database. 17

Administration of Tenant Databases. 18

4.2 Server Architecture of SAP HANA XS Advanced Runtime Platform. 20

4.3 Multiple-Host (Distributed) Systems. 23

Scale-Out Architecture of Tenant Databases. 24

Scale-Out Architecture of SAP HANA XS Advanced Runtime Platform. 26

4.4 Server Components of the SAP HANA Database. 29

4.5 System Limitations. 33

5 SAP HANA Administration Tools. 38

5.1 SAP HANA Cockpit. 41

5.2 SAP HANA Studio. 42

6 SAP HANA Hardware and Cloud Measurement Tools for Tailored Data Center Integration 44

7 SAP Solution Manager for SAP HANA Administration. 45

7.1 Connecting SAP Solution Manager to SAP HANA. 46

Configuring an SAP HANA System to Connect to the System Landscape Directory (SLD). 46

Change the Default SLD Data Supplier Configuration. 53

Configuring SAP HANA for System Replication Technical Scenario in SAP Solution Manager 56

7.2 Central Monitoring and Administration with SAP Solution Manager. 60

7.3 Analyzing the Root Cause of Problems. 60

7.4 Controlling Change. 61

8 System Administration. 62

8.1 Aggregate Monitoring and Administration. 62

8.2 Starting and Stopping SAP HANA Systems. 63

Starting and Stopping Systems with SAPControl. 63

Restart Sequence. 64

8.3 Managing Tenant Databases. . . . 65
Creating and Configuring Tenant Databases. . . . 65
Monitoring and Managing Tenant Databases. . . . 110
Memory and CPU Usage for Tenant Databases. . . . 113
Using SAP Web Dispatcher for Load Balancing with Tenant Databases. . . . 115
8.4 SAP HANA System Properties. . . . 126
Configuring SAP HANA System Properties (INI Files). . . . 126
Database-Specific Configuration Parameters. . . . 130
Console Tool setParameter.py. . . . 133
Configure System Usage Type. . . . 135
Reserve Connections for Administrators. . . . 136
8.5 Managing SAP HANA Licenses. . . . 137
License Keys for SAP HANA Database. . . . 137
8.6 Monitoring the SAP HANA Database. . . . 139
Monitoring in SAP HANA Cockpit. . . . 140
Monitoring in SAP HANA Studio. . . . 140
System and Statistics Views. . . . 141
The Statistics Service. . . . 143
8.7 Managing and Monitoring SAP HANA Performance. . . . 148
Caching. . . . 149
Hints. . . . 158
Managing Bandwidth (File IO Throttle). . . . 164
SQL Analyzer Tool for SAP HANA. . . . 168
8.8 Memory Usage in the SAP HANA Database. . . . 168
Types of Memory: Overview. . . . 169
SAP HANA Used Memory. . . . 171
SAP HANA Memory Usage and the Operating System. . . . 173
Memory Allocator Statistics. . . . 176
Memory Sizing. . . . 177
Allocated Memory Pools and Allocation Limits. . . . 178
Change the Global Memory Allocation Limit. . . . 179
8.9 Persistent Data Storage in the SAP HANA Database. . . . 180
Data and Log Volumes. . . . 181
Partitioning Data Volumes. . . . 184
Persistent Memory. . . . 186
SAP HANA Fast Restart Option. . . . 191
Persistence Consistency Check. . . . 194
8.10 Data Tiering. . . . 195
SAP HANA Native Storage Extension. . . . 196
Extension Node. . . . 243
8.11 Managing Tables. . . . 244

Columnar and Row-Based Data Storage . . . 245
Managing Indexes . . . 248
Locks . . . 250
Temporal Tables . . . 253
The JSON Document Store . . . 258
Basic Table Management in SAP HANA Cockpit . . . 259
Basic Table Management in SAP HANA Studio . . . 260
Importing Data from External Sources . . . 260
Importing and Exporting Data With Cloud Sources . . . 262
Table and Catalog Consistency Checks . . . 282
Row Store Reorganization . . . 291
Memory Management in the Column Store . . . 292
The Delta Merge Operation . . . 301
Data Compression in the Column Store . . . 313
Table Partitioning . . . 317
Table Placement . . . 370
Table Replication . . . 380
Redistributing Tables in a Scaleout SAP HANA System . . . 405
8.12 Workload Management . . . 415
Workload in the Context of SAP HANA . . . 415
Controlling CPU Consumption . . . 421
Controlling Parallel Execution of SQL Statements . . . 428
Setting a Memory Limit for SQL Statements . . . 431
Managing Peak Load (Admission Control) . . . 434
Setting User Parameters for Workload . . . 437
Managing Workload with Workload Classes . . . 439
Example Workload Management Scenarios . . . 457
8.13 Getting Support . . . 460
Diagnostic Files and Logs . . . 461
Traces . . . 462
Troubleshooting an Inaccessible or Unresponsive SAP HANA System . . . 486
Collecting Diagnosis Information for SAP Support . . . 486
Problem Analysis Using hdbcons . . . 494
Open a Support Connection . . . 495
8.14 Scheduling Administrative Tasks . . . 495

9 Security Administration and User Management . . . 498
9.1 Monitoring Critical Security Settings . . . 498
9.2 Auditing Activity in the SAP HANA Database . . . 499
9.3 Managing Data Encryption in SAP HANA . . . 499
Server-Side Data Encryption Services . . . 500
Client-Side Data Encryption . . . 540

Use FIPS 140-2 Certified Cryptographic Kernel in CommonCryptoLib. 541
9.4 Managing Client Certificates. 542
9.5 Managing SAP HANA Users. 543
Database Users. 544
Operating System User <sid>adm. 552
Configuring User Authentication and Single-Sign On. 552
User Authorization. 562
Configuring LDAP Integration. 569

10 SAP HANA Lifecycle Management. 582
10.1 SAP HANA Platform Lifecycle Management. 583
About the SAP HANA Database Lifecycle Manager (HDBLCM). 585
10.2 SAP HANA Application Lifecycle Management. 617
Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model. 618
Installing and Updating Products and Software Components in SAP HANA XS Advanced Model 626
Configuring SAP HANA Applications with the Process Engine. 656
10.3 SAP HANA Content. 661
SAP HANA Archive Types. 661
Deploy a Product Archive (*.ZIP). 662
Deploy a Delivery Unit Archive (*.tgz). 663

11 Landscape Management and Network Administration. 664
11.1 Landscape Management. 664
Copying and Moving a System Using Platform LCM Tools. 666
Copying and Moving Tenant Databases. 671
Copying a System Using System Replication 705
Renaming a System. 706
11.2 Network Administration. 714
Configuration of SAP HANA in an IPv6 Environment. 715
Network Zones. 716
Ports and Connections. 718
Host Name Resolution. 743

12 Availability and Scalability 756
12.1 High Availability for SAP HANA. 756
SAP HANA High Availability Support. 757
SAP HANA System Replication. 765
Setting Up Host Auto-Failover. 896
Implementing a HA/DR Provider. 905
12.2 SAP HANA Database Backup and Recovery. 920
Planning Your Backup and Recovery Strategy. 921

Points to Note About Backup and Recovery. . . . 923
Authorizations Needed for Backup and Recovery. . . . 938
SAP HANA Backup Types. . . . 941
Working with Third-Party Backup Tools. . . . 969
SAP HANA Backup. . . . 983
SAP HANA Recovery. . . . 1029
Copying a Database Using Backup and Recovery. . . . 1065
Diagnosis Files for Backup and Recovery. . . . 1070
Reference. . . . 1075
12.3 Scaling SAP HANA. . . . 1079
Aspects of Scalability. . . . 1080
Adding and Removing Hosts. . . . 1081
Configuring Host Roles. . . . 1104
Configuring the Network for Multiple Hosts. . . . 1115
Scaling SAP HANA Extended Application Services, Classic Model. . . . 1123
Starting and Stopping Distributed SAP HANA Systems Using SAPControl. . . . 1124

13 SAP HANA Deployment Infrastructure (DI) Administration. . . . 1125
13.1 SAP HANA DI Administration at a Glance. . . . 1126
13.2 SAP HANA DI Administrator Roles. . . . 1129
13.3 SAP HANA DI Administrator Tasks. . . . 1130
13.4 The SQL API for SAP HANA DI. . . . 1131
13.5 SAP HANA DI Security. . . . 1132

14 Application Run-Time Services. . . . 1134
14.1 Maintaining the SAP HANA XS Classic Model Run Time. . . . 1134
SAP HANA XS Classic Administration Tools. . . . 1135
SAP HANA XS Classic Administration Roles. . . . 1137
SAP HANA XS Classic Configuration Parameters. . . . 1140
Maintaining Application Runtime Configurations. . . . 1146
Managing Trust Relationships. . . . 1160
Maintaining SAML Providers. . . . 1167
Maintaining SMTP Server Configurations. . . . 1176
Maintaining HTTP Access to SAP HANA. . . . 1183
Maintaining Single Sign-On for SAP HANA XS Applications. . . . 1191
Maintaining User Self Service Tools. . . . 1202
Scheduling XS Jobs. . . . 1227
Maintaining Translation Text Strings. . . . 1241
Maintaining HTTP Traces for SAP HANA XS Applications. . . . 1248
14.2 Maintaining the SAP HANA XS Advanced Model Run Time. . . . 1256
XS Advanced Platform Components. . . . 1257
Maintaining the XS Advanced Run-time Environment with a Command-Line Interface. . . . 1261

Maintaining the XS Advanced Run-time Environment with a Graphical User Interface. 1387
XS Advanced User Management. 1469
XS Advanced System Configuration Parameters. 1480
Backup and Recovery in XS Advanced. 1487
Logging and Auditing in XS Advanced. 1490
Platform Sizing in XS Advanced. 1498
Configuring the XS Advanced File-System Service. 1501
Configuring the XS Advanced Platform Router. 1506
Maintaining Single Sign-On for XS Advanced Applications. 1509
Multi-Host Setup with XS Advanced. 1512

15 Data Access. 1518
15.1 SAP HANA Smart Data Access. 1519
Setting Up ODBC Drivers. 1520
Managing Privileges. 1535
Managing Credentials. 1536
Managing Remote Sources. 1550
Using Linked Database. 1624
Managing Virtual Tables. 1628
Virtualizing Parameterized Views. 1641
Virtualizing Table User-Defined Functions. 1644
Monitor Remote Connections and Statements. 1648
Data Type Support. 1649
Functions Pushed Down to Remote Sources. 1652
Synonyms. 1652
Automatic Failover Support. 1653
Safe Mode for ODBC Connections. 1653
Session-Specific Information for Connections. 1655
Smart Data Access System Parameters. 1655
Smart Data Access Performance Tools. 1658
Troubleshooting Smart Data Access. 1676
15.2 SAP HANA Hadoop Integration. 1676

16 Important Disclaimer for Features in SAP HANA. 1677

1 SAP HANA Administration Guide

The SAP HANA Administration Guide is the central operations documentation for the on-premise deployment of the SAP HANA platform.

Scope of this Document

The *SAP HANA Administration Guide* is the central administration guide which aims to provide background information and details of procedures to operate SAP HANA. However, there are several administration resources and tools available to administrators, including cockpit applications and command line utilities; some of these tools have their own detailed documentation (see SAP HANA Administration Tools for a graphical overview of the tools available). This *SAP HANA Administration Guide* provides links to other documentation sets wherever necessary, in particular, the *SAP HANA SQL Reference Guide* and the *SAP HANA cockpit Administration Guide* are very closely integrated by linking.

The *SAP HANA Administration Guide* does not cover administration tasks related to some additional capabilities that may be installed in the SAP HANA system such as SAP HANA dynamic tiering and SAP HANA streaming analytics. For more information about the administration of these capabilities, see the relevant documentation on SAP Help Portal.

Related Information

SAP HANA Administration Tools [page 38]

2 Administration Information Map

In addition to the SAP HANA Administration Guide, several other documents in the SAP HANA platform documentation set provide administrators with important information.

SAP HANA Administration Guide

This guide is the entry point for all information related to the ongoing operation and maintenance of an on-premise deployment of the SAP HANA platform. It contains information about various administration tasks related to the following main areas:

- Administration and monitoring at the landscape, system, and database level
- Monitoring and configuration of security-related settings
- Landscape management and network administration
- Administration of the SAP HANA XS runtime environment
- Setup and management of the SAP HANA Deployment Infrastructure (HDI)
- High availability (including backup and recovery) and scalability
- Data access and integration with SAP HANA data provisioning tools and technologies

The *SAP HANA Administration Guide* also includes information on using the following native SAP HANA administration tools:

- SAP HANA cockpit

  > → Tip
  >
  > For the documentation of the latest SAP HANA cockpit support package (SP), see https://help.sap.com/viewer/p/SAP_HANA_COCKPIT

- SAP HANA studio
- SAP HANA database lifecycle manager (HDBLCM)
- SAP HANA `hdsql` command line
- SAP HANA XS administration tools
- SAP HANA application lifecycle management

SAP HANA Troubleshooting and Performance Analysis Guide

This guide describes what steps you can take to identify and resolve specific performance issues and what you can do to enhance the performance of your SAP HANA database in areas such as:

- Host resources (CPU, memory, disk)
- Size and growth of data structures
- Transactional problems
- SQL statement performance
- Security, authorization, and licensing
- Configuration

Open the SAP HANA Troubleshooting and Performance Analysis Guide

SAP HANA Tenant Databases Operations Guide

This guide brings together all the information required for the operation of an SAP HANA multitenant system, including:

- Overview of architecture and concepts of multitenant systems
- Creating and configuring tenant databases
- Monitoring and managing tenant databases
- Copying and moving tenant databases

Open the SAP HANA Tenant Databases Operations Guide

SAP HANA Master Guide

This guide is the entry point for planning the installation of your SAP HANA system landscape. It provides you with overview information on the aspects such as:

- Use cases and scenarios that SAP HANA can be used in from an application point of view
- Deployment options for SAP HANA on-premise or in the cloud
- Implementation and operation activities during the lifecycle of SAP HANA

Open the SAP HANA Master Guide

SAP HANA SQL and System Views Reference

The *SAP HANA SQL Reference* describes all SQL data types, predicates, operators, expressions, functions, statements, and error codes. The *SAP HANA System Views Reference* describes all system views. You can use the information in this guide to perform the following typical tasks:

- Monitor the current status of the SAP HANA system and database by querying monitoring views
- Analyze and diagnose historical monitoring data by querying statistics views
- Configure the database using SQL commands

Open the SAP HANA SQL and System Views Reference

Administration Guides for Additional SAP HANA Capabilities

Separate administration information is available for the following additional capabilities that may be installed in your SAP HANA system:

- SAP HANA accelerator for SAP ASE
- SAP HANA data warehousing foundation
- SAP HANA dynamic tiering
- SAP HANA smart data integration and SAP HANA smart data quality
- SAP HANA streaming analytics
- SAP HANA real-time replication with SAP Landscape Transformation Replication Server

For more information, see the relevant documentation on SAP Help Portal.

> **Note**
>
> The topics listed above for each guide are not intended to be exhaustive but representative.

Target Audiences

Document	Target Audience	Content Type
SAP HANA Administration Guide	Technology consultants, system administrators	Task- and role-oriented
SAP HANA Tenant Databases Operations Guide	Technology consultants, system administrators	Concept, task- and role-oriented
SAP HANA Troubleshooting and Performance Analysis Guide	Technology consultants, system administrators	Troubleshooting, root-cause analysis
SAP HANA Master Guide	Technology consultants, security consultants, system administrators	Concept and overview
SAP HANA SQL and System Views Reference	Technology consultants, security consultants, system administrators	Reference

Additional Documentation Resources

Product Documentation

For more information about the SAP HANA landscape, including installation and security, see https://help.sap.com/viewer/p/SAP_HANA_PLATFORM.

SAP Notes

SAP Note	Title
2380229	SAP HANA Platform 2.0 – Central Note
1730928	Using external software in an SAP HANA appliance
1730929	Using external tools in an SAP HANA appliance
1730930	Using anti-virus software in an SAP HANA appliance
1730996	Non-recommended external software and software versions
1730997	Non-recommended versions of anti-virus software
1730998	Non-recommended versions of backup tools
1730999	Configuration changes in SAP HANA appliance
1731000	Non-recommended configuration changes

Other Information

For more information about specific topics, see the quick links in the table below.

Content	SAP Service Marketplace or SDN Quick Link
Related SAP Notes	https://support.sap.com/notes
Released platforms	https://apps.support.sap.com/sap/support/pam

Content	SAP Service Marketplace or SDN Quick Link
SAP Solution Manager community	https://go.sap.com/community/topic/solution-manager.html
SAP NetWeaver community	https://go.sap.com/community/topic/netweaver.html
In-memory computing community	https://go.sap.com/community/topic/hana.html

Related Information

SAP HANA Master Guide
SAP HANA Tenant Databases Operations Guide
SAP HANA Troubleshooting and Performance Analysis Guide
SAP HANA SQL and System Views Reference

3 Database Administration Tasks at a Glance

Overview of key tasks for the ongoing operation and maintenance of the SAP HANA database

Initial Administration Tasks

After the initial setup and initial data load, it is strongly recommended that you perform a full data and file-system backup (including configuration backup). For more information, see the section on database backup and recovery.

> **Note**
>
> In replication scenarios with SAP Landscape Transformation Replication Server, do not switch off log writing during the initial data load from SAP ERP into the SAP HANA database. There is no system table or log file that records the information that log writing has been switched off, so it is not possible to check whether log writing has been switched on or off.

Regular Administration Tasks

- Monitor the health of your SAP HANA system using, for example, the SAP HANA cockpit. The most important system information to review is:
  - Overall system status
  - Status of database services, for example, name server and index server
  - General system information (software versions and so on)
  - Alerts generated by the statistics service
  - Usage of important system resources: memory, CPU and disk

  For more information, see *Monitoring the SAP HANA Database*. The SAP Note *2400024 - How-To: SAP HANA Administration and Monitoring* also gives details of recommended tasks in the area of administration, monitoring and housekeeping.
- Perform regular data backups, including configuration backups.
  There are no specific guidelines for backup frequency, which depends on the usage scenario, but for general guidelines, see *Planning Your Backup and Recovery Strategy*.
- Avoid the log backup area becoming full by archiving old log backups to a different location.

  > **Caution**
  >
  > Do not delete log segments at the operating system level, as the log area will become unusable and the database may stop working.

  For more information, see *Housekeeping for Backup Catalog and Backup Storage*.

- Monitor disk space used for diagnosis files and delete files that are no longer needed.
- Check for the latest configuration parameter recommendations
  The SAP Note *2600030 - Parameter Recommendations in SAP HANA Environments* is updated regularly with version-specific information covering cases where configuration parameter settings other than the default values are recommended.

On-Demand Administration Tasks

- In the event of problems with the SAP HANA database, you can check log and trace files for errors. You can also activate and configure several trace types.
  For more information, see *Diagnosis Files* and *Configure Traces*.
- Before updating SAP HANA, perform a data backup including configuration files. This allows for the recovery of the system in the event the software update fails.

Recurring Administration Tasks

Both SAP HANA Extended Services classic model and advanced model provide job-scheduling features so that you can run routine administration tasks in the background at planned intervals.

The following job-scheduling features are available:

- SAP HANA XS classic: Use the .xsjob file to run XS JavaScript functions or call an SQLScript procedure
- SAP HANA XS advanced: Use the Job Scheduler service and Job Scheduler Dashboard for administrators

Related Information

SAP HANA Database Backup and Recovery [page 920]
Monitoring the SAP HANA Database [page 139]
Housekeeping: Deleting and Archiving Backups
Planning Your Backup and Recovery Strategy [page 921]
Diagnostic Files and Logs [page 461]
Configure Traces in SAP HANA Studio
Configure Tracing in the SAP HANA Database Explorer
Scheduling XS Jobs [page 1227]
Scheduling Jobs in XS Advanced [page 1439]
SAP Note 2600030
SAP Note 2400024

4 SAP HANA System Architecture Overview

An SAP HANA system comprises multiple isolated databases and may consist of one host or a cluster of several hosts.

An **SAP HANA system** is identified by a single system ID (SID) and contains one or more tenant databases and one system database. Databases are identified by a SID and a database name. From the administration perspective, there is a distinction between tasks performed at system level and those performed at database level. Database clients, such as the SAP HANA cockpit, connect to specific databases.

The **SAP HANA XS advanced application server** is a layer on top of SAP HANA that provides the platform for running SAP HANA-based Web applications. It is an integral part of the SAP HANA system.

A system may consist of one host or a cluster of several hosts. This is referred to as a **muliple-host, distributed system, or scale-out system** and supports scalability and availability.

The following sections provide overview information about these aspects of system architecture.

Related Information

Server Architecture of Tenant Databases [page 15]
Server Architecture of SAP HANA XS Advanced Runtime Platform [page 20]
Multiple-Host (Distributed) Systems [page 23]
Server Components of the SAP HANA Database [page 29]

4.1 Server Architecture of Tenant Databases

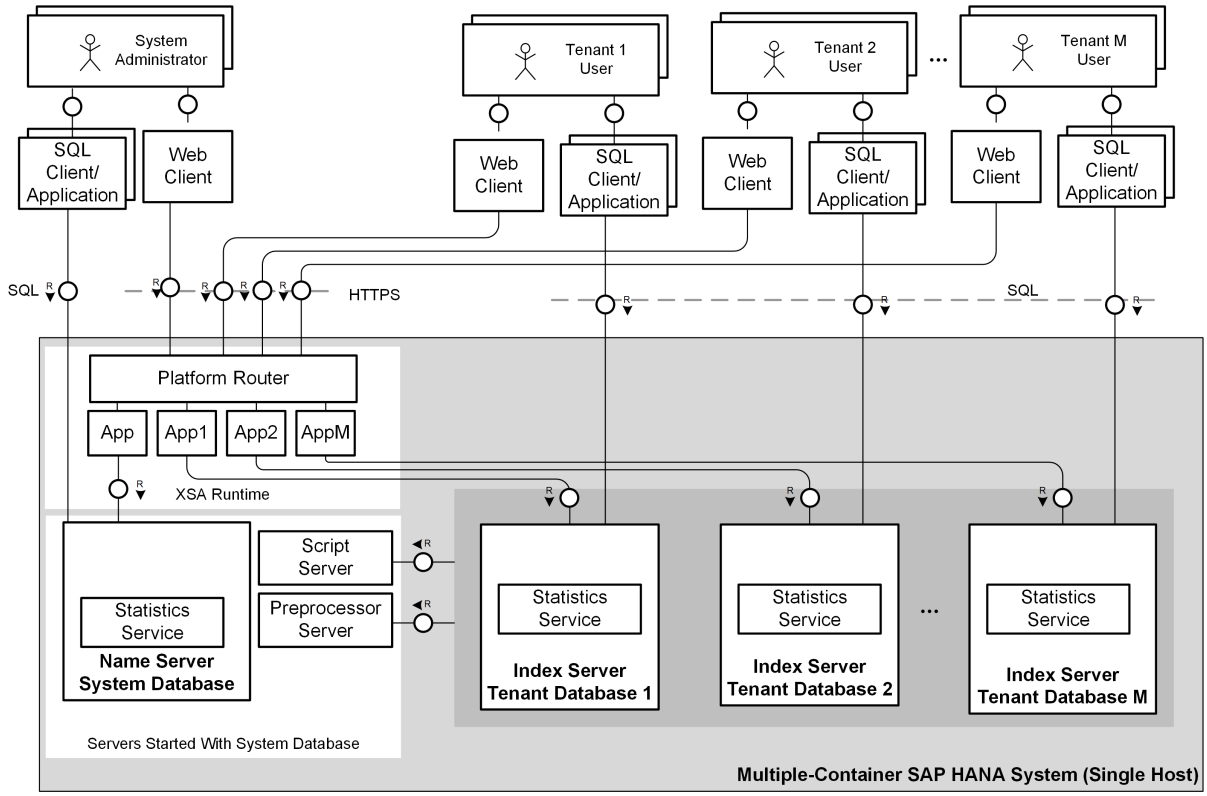
An SAP HANA database consists of multiple servers, for example, name server, index server, preprocessor server, and so on. The databases in an SAP HANA system run different combinations of these servers. The most important server is the index server. It contains the actual data stores and the engines for processing the data and runs in every tenant database.

Only the **system database** runs the name server. The name server contains landscape information about the system as a whole, including which tenant databases exist. It also provides index server functionality for the system database. The name server does not own information about the location of tables and table partitions in tenant databases. Database-related information is stored in the relevant tenant database catalog.

Tenant databases require only an own index server. Servers that do not persist data, such as the compile server and the preprocessor server, run on the system database and serve all databases.

> **Note**
>
> For a full list and description of all SAP HANA servers, see *Server Components of the SAP HANA Database*.

The following figure shows a sample system with three databases (system database and three tenant databases) on a single host.



Single-Host SAP HANA System with Tenant Databases

> **Note**
>
> If the SAP HANA XS classic server is available, it runs embedded in the (master) index server of the tenant database by default, although it can be added as a separate service if necessary. The **SAP Web Dispatcher**, which runs as a separate database service on the host of the system database, is used to route incoming HTTP requests from clients to the correct XS classic server based on virtual host names. This is part of network configuration. In addition to the system-internal Web Dispatcher, you can implement an external Web Dispatcher for load distribution. See the section on using the SAP Web Dispatcher for load balancing with tenant databases.

Related Information

Server Components of the SAP HANA Database [page 29]
Connections from Database Clients and Web Clients to SAP HANA [page 718]
Port Assignment in Tenant Databases [page 725]
Scale-Out Architecture of Tenant Databases [page 24]
Using SAP Web Dispatcher for Load Balancing with Tenant Databases [page 115]

4.1.1 Tenant Databases

SAP HANA supports multiple isolated databases in a single SAP HANA system. These are referred to as tenant databases.

An SAP HANA system is capable of containing more than one tenant database.

A system always has exactly one system database, used for central system administration, and any number of tenant databases (including zero). An SAP HANA system is identified by a single system ID (SID). Databases are identified by a SID and a database name. From the administration perspective, there is a distinction between tasks performed at system level and those performed at database level. Database clients, such as the SAP HANA cockpit, connect to specific databases.

All the databases share the same installation of database system software, the same computing resources, and the same system administration. However, each database is self-contained and fully isolated with its own:

- Set of database users
- Database catalog
- Repository
- Persistence
- Backups
- Traces and logs

Although database objects such as schemas, tables, views, procedures, and so on are local to the database, cross-database SELECT queries are possible. This supports cross-application reporting, for example.

Related Information

Server Architecture of Tenant Databases [page 15]
Scale-Out Architecture of Tenant Databases [page 24]
The System Database [page 17]
Administration of Tenant Databases [page 18]

4.1.2 The System Database

The system database is created during either installation or conversion from a single-container system to a tenant database system. The system database contains information about the system as a whole, as well as all its tenant databases. It is used for central system administration.

A system has exactly one system database. It contains the data and users for system administration. System administration tools, such as the SAP HANA cockpit, can connect to this database. The system database stores overall system landscape information, including knowledge of the tenant databases that exist in the system. However, it doesn't own database-related topology information, that is, information about the location of tables and table partitions in databases. Database-related topology information is stored in the relevant tenant database catalog.

Administration tasks performed in the system database apply to the system as a whole and all of its databases (for example, system-level configuration settings), or can target specific tenant databases (for example, backup of a tenant database). For more information, see *Administration of Tenant Databases*.

Things to Remember About the System Database

- The system database does not have the same functionality as a tenant database.
- The system database is not a database with full SQL support.
- The system database cannot be distributed across multiple hosts, in other words, scale-out is not possible.
- If you need a full-featured SAP HANA database, you always have to create at least one tenant database.
- The system database does not support Application Function Libraries (AFL) and SAP liveCache applications.
- Cross-database access between the system database and a tenant database is not possible. The system database can show monitoring data from tenant databases (views in the schema SYS_DATABASES) but can never show actual content from tenant databases.
- The system database cannot be copied or moved to another host.
- SAP HANA options can only run in tenant databases.
- Tenant-specific configurations cannot be set in the system database. Only global settings are allowed.
- Features can only be restricted or disabled at high level for tenant databases.

Related Information

Administration of Tenant Databases [page 18]
Memory and CPU Usage for Tenant Databases [page 113]
Cross-Database Authorization in Tenant Databases
Restricted Features in Tenant Databases

4.1.3 Administration of Tenant Databases

In SAP HANA systems there is a distinction between administration tasks performed at system level and those performed at database level.

System Versus Database Administration

Tenant database systems have two levels of administration.

Some administration tasks are performed in the system database and apply globally to the system and all its databases. They include for example:

- Starting and stopping the whole system
- Monitoring the system
- Configuring parameters in configuration (*ini) files at system level
- Setting up and configuring tenant databases, for example:
  - Creating and dropping tenant databases
  - Disabling features on tenant databases
  - Configuring system- and database-specific parameters in configuration (*ini) files
  - Scaling out tenant databases by adding services
- Backing up tenant databases
- Recovering tenant databases

Some administration tasks are performed in the tenant database and apply only to that database. They include for example:

- Monitoring the database
- Provisioning database users
- Creating and deleting schemas, tables, and indexes in the database
- Backing up the database
- Configuring database-specific parameters in configuration (*ini) files

Administration Tools

Several tools are available for the administration of SAP HANA. While all tools support database-level administration, system-level administration of tenant databases requires the SAP HANA cockpit (for example, monitoring availability of tenant databases, creating and deleting tenant databases).

For more information about the SAP HANA cockpit and other administration tools, see the section on administration tools in the *SAP HANA Administration Guide*.

Related Information

Tenant Databases [page 17]
The System Database [page 17]
Creating and Configuring Tenant Databases [page 65]
SAP HANA Administration Tools [page 38]
Monitoring and Managing Tenant Databases [page 110]

4.2 Server Architecture of SAP HANA XS Advanced Runtime Platform

SAP HANA extended application services, advanced model (XS advanced for short) provides a comprehensive platform for the development and execution of micro-service oriented applications, taking advantage of SAP HANA's in-memory architecture and parallel execution capabilities.

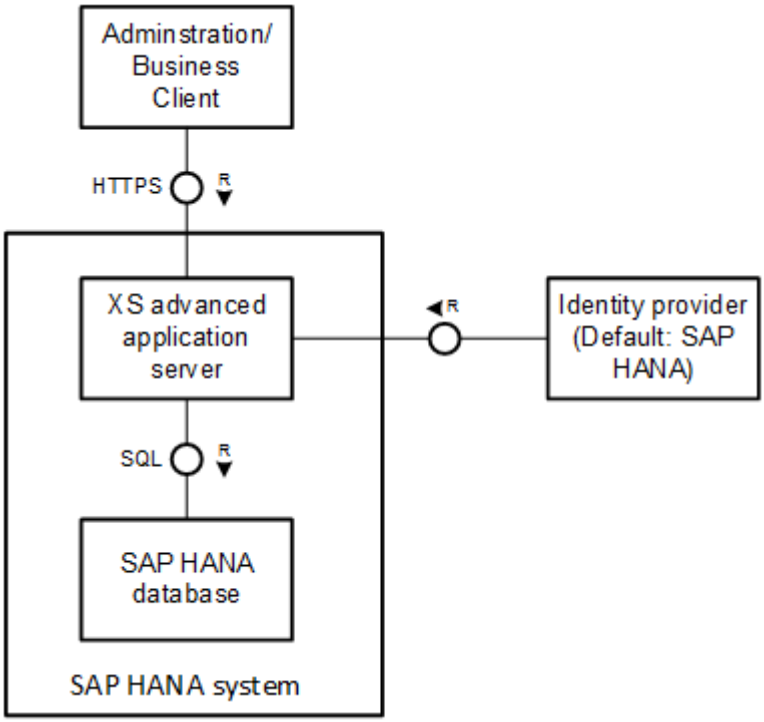
About SAP HANA XS Advanced

SAP HANA XS advanced offers a rich set of embedded services that enable an end-to-end support for web-based applications including lightweight web servers, persistency services, and a configurable identity provider. Furthermore, the platform supports polyglot application development with a core set of pre-deployed runtimes that are accepted as industry standard, for example, node.js or JavaEE.

Although the built-in runtimes come with first-class development and monitoring support, the platform has an open architecture that allows you to add custom runtimes. This high flexibility makes it essential that you put a strong focus on security concepts, not only when configuring and setting up the infrastructure, but also throughout operating the system.

Architecture Overview

As illustrated in the following diagram, the basic system architecture has a classic 3-tier approach:

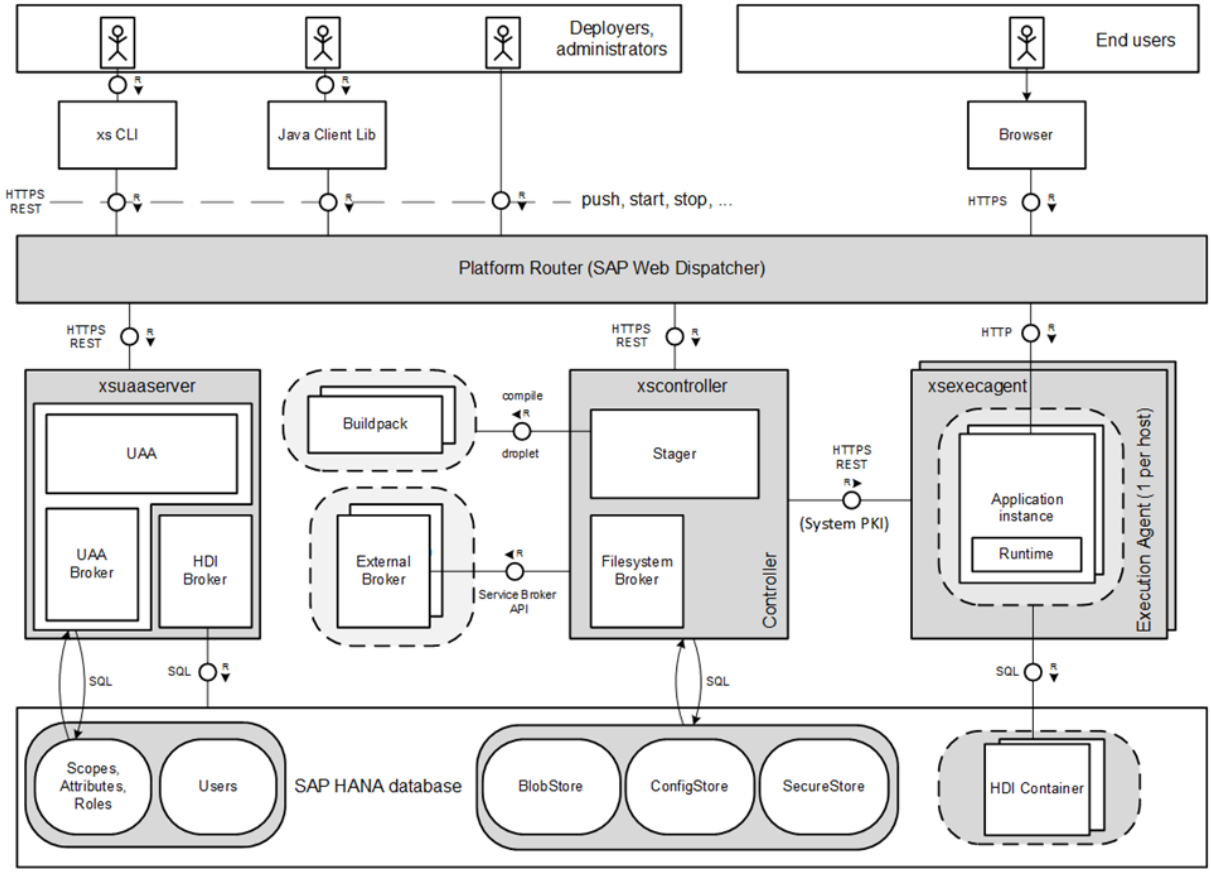


3-Tier Architecture of SAP HANA with XS Advanced

There is a distinction between the overall SAP HANA system and the SAP HANA XS advanced application server. The SAP HANA system refers to the entire SAP HANA platform part of the integrated solution. The SAP HANA XS advanced application server describes only the runtime platform as an integral part of the solution. All services of the SAP HANA system share the same system identifiers (that is, instance number and SID) and are controlled by the `hdbdaemon` service.

The third tier, represented by an SAP HANA database, provides persistency services, that is, data storage. In contrast, the application server components in the middle tier are responsible for deploying, running, and monitoring the applications. Most security-related features such as authentication, authorization, and auditing are primarily enforced in this layer. End users interact on the client layer with system or business users that are authenticated by an identity provider (IdP), which is SAP HANA user management by default. However, both the server components and the applications themselves access the SAP HANA database only through technical database users that the platform generates implicitly. Direct access to the database is only intended for database administration and monitoring purposes.

The following diagram provides a more detailed overview of the technical system landscape of the XS advanced application server. All relevant components and storages used by the application server layer are highlighted with a gray background.



Technical System Landscape of XS Advanced Application Server

The XS advanced application server relies on the following SAP HANA services contributing to the integrated platform solution:

1. `xscontroller` (Controller, FileSystem Broker, Platform Router)
2. `xsexecagent` (Execution Agent)
3. `xsuaaserver` (UAA, UAA Broker and SAP HANA Service Broker)

Administration of the XS Advanced Runtime

A number of administration tools are available to enable you to maintain and manage the various components of the XS advanced runtime environment. For more information, see the section on maintaining the SAP HANA XS advanced model run time.

Related Information

Server Components of the SAP HANA Database [page 29]
Installing an SAP HANA System Including the XS Advanced Runtime
Connections for SAP HANA Extended Application Services, Advanced Model [page 736]

Maintaining the SAP HANA XS Advanced Model Run Time [page 1256]

4.3 Multiple-Host (Distributed) Systems

An SAP HANA system can be distributed across multiple hosts for reasons of scalability and availability.

A multiple-host or distributed SAP HANA system is a system that is installed on more than one host. Otherwise, it is a single-host system.

An SAP HANA system installed on multiple hosts is identified by a single system ID (SID). It is perceived as one unit from the perspective of the administrator, who can install, update, start up, or shut down the system as a whole. The different databases of the system share the same metadata and requests from client applications can be transparently dispatched.

The main reason for distributing a system across multiple hosts is **scale-out**. A multiple-host system can overcome hardware limitations of a single physical server, and it can distribute the load between multiple servers. Distributing a system also supports **failover**. One or more hosts can be configured to work in standby mode, so that if an active hosts fails, a standby host automatically takes its place. The index servers on standby hosts do not contain any data and do not receive any requests.

For more information about hosts, including host roles, fail-over configuration, and storage options, see the *SAP HANA Server Installation and Update Guide*.

Distributing Data

In a multiple-host system each index server is usually assigned to its own host for maximum performance. SAP HANA supports different ways of distributing data across the hosts:

- Different tables can be assigned to different index servers.
- A table can be split, or partitioned, in a way that different rows of the table are stored on different index servers
- A table can be replicated to multiple index servers, for better query and join performance.

When you create new tables or partitions, data is distributed to the available hosts by the system. By default a 'round-robin' distribution method is used, but tables can also be positioned by using table placement rules or by specifying a host and port number with the SQL CREATE TABLE statement in the location clause; this gives complete control over the positioning of individual tables.

Specific applications may have predefined table distribution rules and in some cases configuration files and documentation are available in SAP Notes to help you to set up the necessary partitioning and table placement rules.

For more information, see the sections on table placement, table partitioning, and table replication.

Distributed Execution

Database clients may send their requests to any index server on any host in a distributed system. If the contacted index server does not own all of the data involved, it delegates the execution of some operations to other index servers, collects the result, and returns it to the database client. The SAP HANA client library supports load balancing and minimizes communication overhead by selecting connections based on load data and routing statements based on information about the location of data.

For more information, see the sections on connecting to SAP HANA databases and servers and statement routing in the *SAP HANA Client Interface Programming Reference*.

Related Information

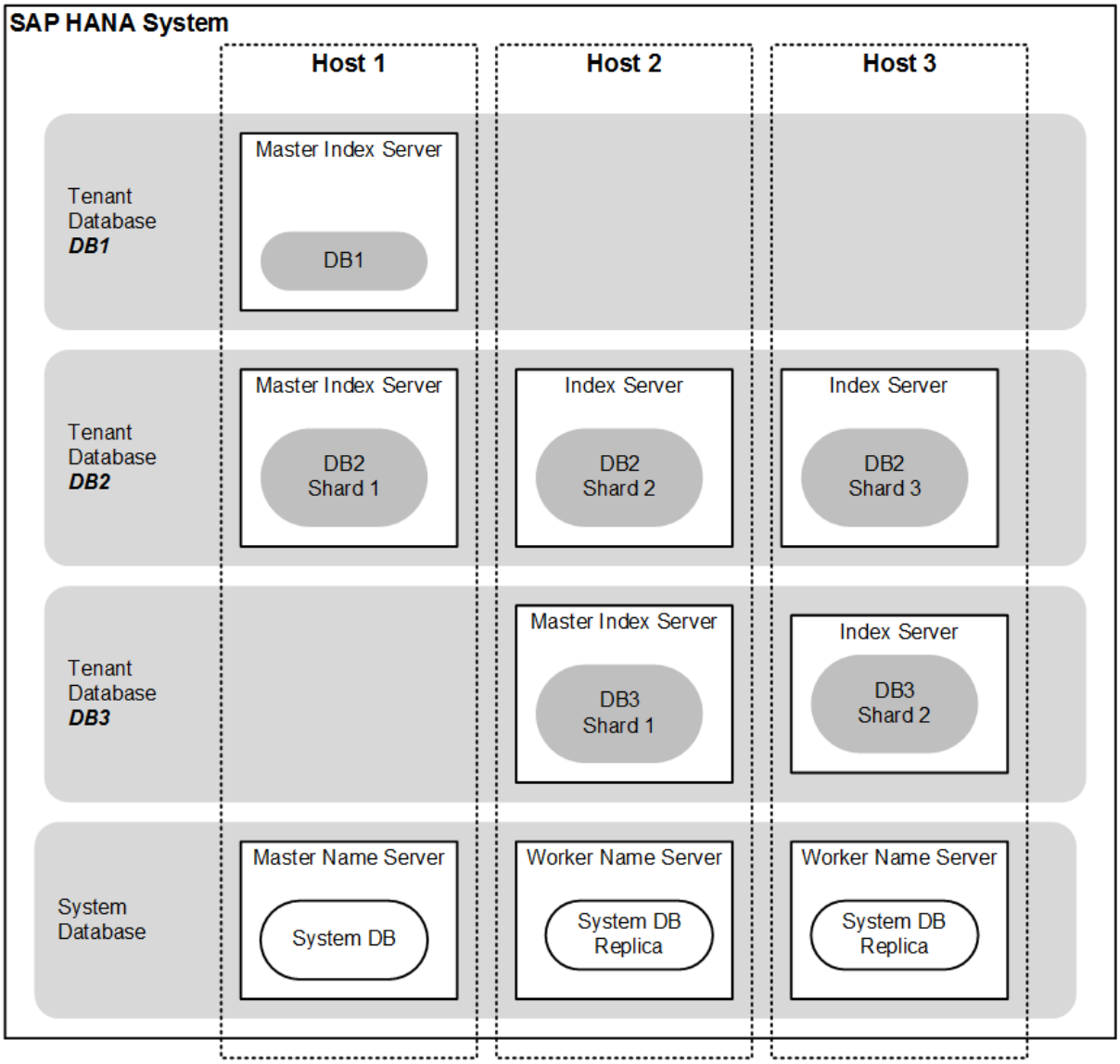
Multiple-Host System Concepts [page 1082]
Connections for Distributed SAP HANA Systems [page 728]
Scale-Out Architecture of Tenant Databases [page 24]
Scale-Out Architecture of SAP HANA XS Advanced Runtime Platform [page 26]
High Availability for SAP HANA [page 756]
Scaling SAP HANA [page 1079]
Table Placement [page 370]
Table Partitioning [page 317]
Table Replication [page 380]
Connecting to SAP HANA Databases and Servers (Client Interface Guide)
Statement Routing (Client Interface Guide)

4.3.1 Scale-Out Architecture of Tenant Databases

Tenant databases can be distributed across several hosts in a multiple-host system.

To ensure system availability, an instance of the system database runs on all hosts (worker and standby) in a single master and multiple workers configuration. Tenant databases can be created on worker hosts and existing databases can be scaled out through the addition of services. If a host fails, the standby instance will fail over all active databases and their services. Like in a single-host system, the master candidate for a failing host is determined. On that host the system database is restarted, if necessary. Up to three hosts can be configured to act as the master host of a system. These three hosts can be set up in the clients with the database name to be reconnected to a tenant database even in the case of a host auto-failover of the master host with the system database.

The following figure shows a tenant database system with three tenant databases distributed across three hosts. Tenant database DB1 has only one index server on host 1, while DB2 and DB3 are distributed across several hosts. Tenant database DB2, for example, is divided into three database shards, each of them with its own index server on a different host. In this context, a database shard is the union of all tables, partitions and replicas of one database that reside on one index server. Tenant database DB3 consists of two shards, one on host 2 and one on host 3. System administrators can specify the host when they create the tenant database, or they can let SAP HANA chose an appropriate host based on load-balancing algorithms.



Multiple-Host System with Tenant Databases

Scale-Out Recommendations

When planning your SAP HANA deployment with tenant databases, various options exist with regard to scale-up versus scale-out.

In general, scaling up offers some performance advantages over scaling out, as memory access is local and minor overhead associated with inter-node network communication is avoided.

Note the following with regard to scale-out:

- It is possible to distribute tenant databases across several hosts in a scale-out system.
- The primary reason to distribute tenant databases generally is when their size is larger than the capacity of a single host. However, other reasons for distributing tenant database may exist, for example, a large SAP Business Warehouse (BW) system requires a scale-out configuration in accordance with its sizing rules.
- If tenant databases are distributed in a scale-out configuration due to sizing requirements, caution is advised when deploying additional tenant databases on the same host as a distributed tenant database

shard. The rationale is this: Workload in distributed scenarios can be somewhat volatile and less predictable. Therefore in many cases, it can be advantageous to dedicate maximum resources of the host to the distributed tenant database shard in order to maintain expected performance.

- In certain cases, more than one distributed tenant database shard may share the same host. In these cases, in order to dedicate maximum resources for a master node (for performance reasons), it is advisable to avoid deploying other tenant databases on the master node. For example, the following deployment should offer performance advantages:
  - Host 1: Master for tenant database 1
  - Host 2: Worker for tenant database 1 and worker for tenant database 2
  - Host 3: Master for tenant database 2
  - Host 4: Standby host for failover

Related Information

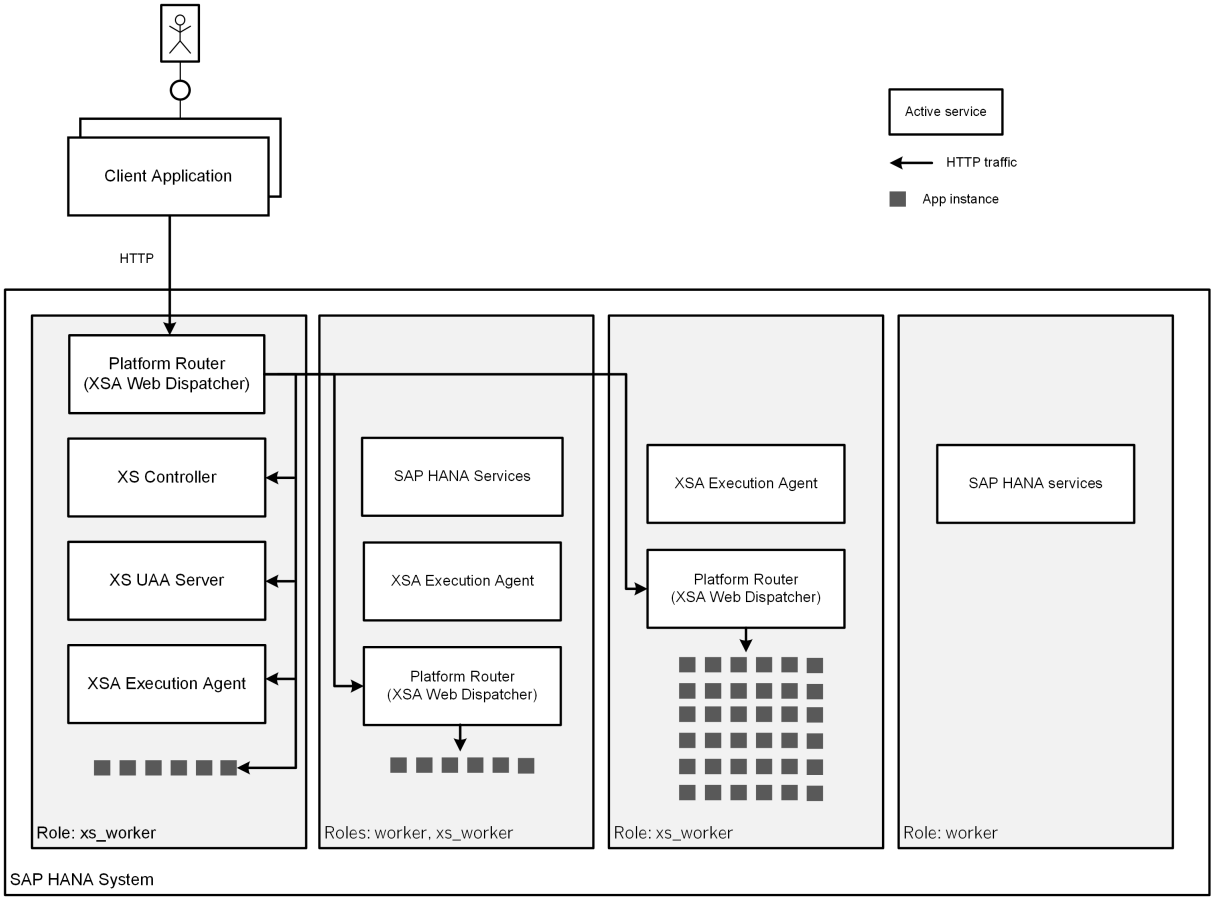
Scaling SAP HANA [page 1079]

4.3.2 Scale-Out Architecture of SAP HANA XS Advanced Runtime Platform

A multiple-host system that includes the SAP HANA XS advanced runtime can be flexibly configured to optimize load balancing and support failover.

During the installation of a multiple-host system with SAP HANA XS advanced, additional host roles are assigned for XS advanced. By default all worker hosts are configured to act as XS worker hosts; that is, they are additionally assigned the role `xs_worker`. However, it is also possible to configure dedicated hosts for XS advanced during installation.

The following figure shows a multiple-host system with two dedicated XS hosts and one shared host.



Multiple-Host System with Dedicated XS Worker Hosts and Shared Hosts

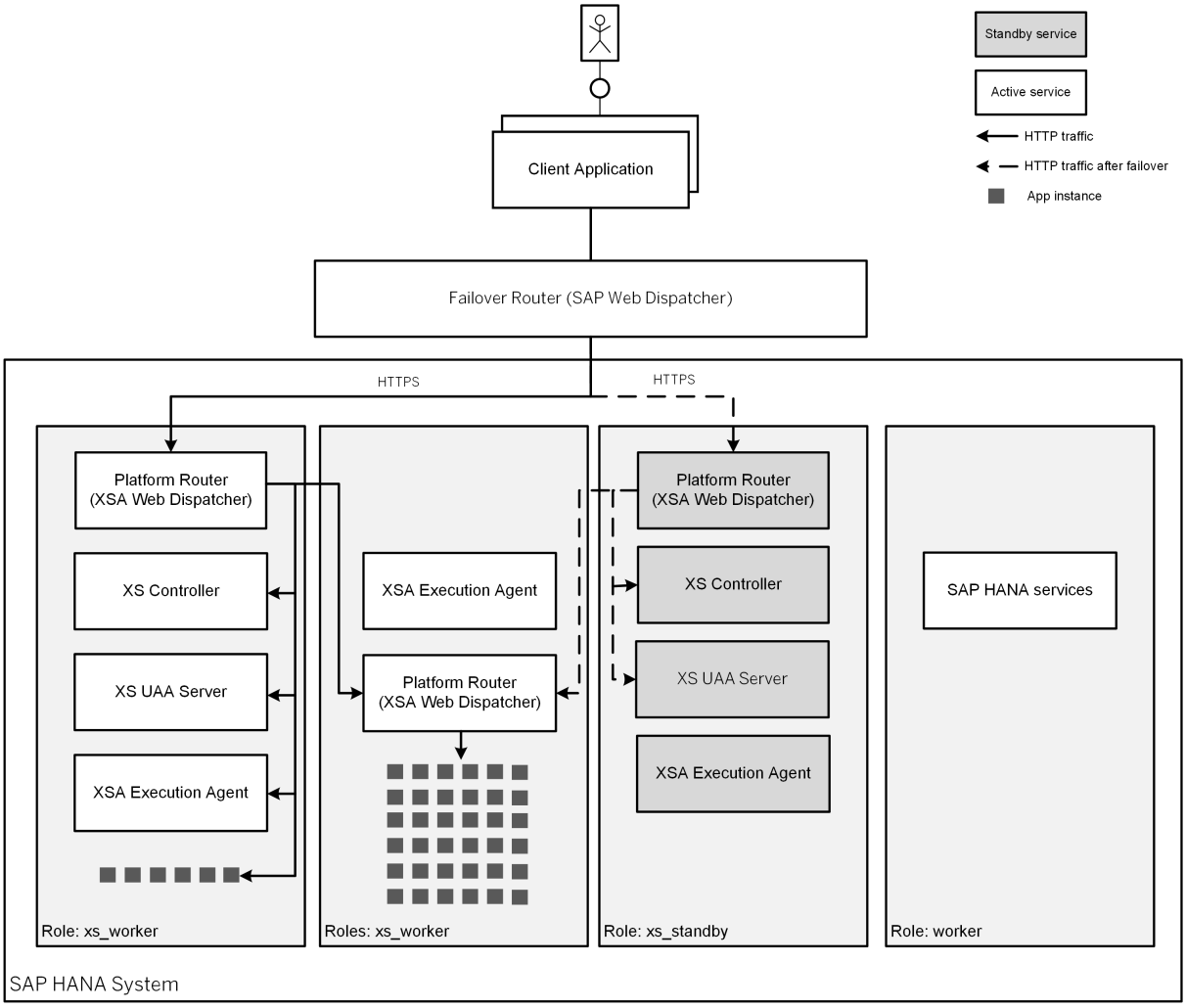
Additionally, the XS advanced runtime can be set up behind a reverse proxy (for example, a load balancer). For more information, see the section on installing the XS advanced runtime in the *SAP HANA Server Installation and Update Guide*.

Failover Configuration

SAP HANA XS advanced integrates into the standard failover mechanisms of SAP HANA: If a worker host fails, a standby host takes over. If a host is both a worker and an XS worker, then the regular standby host takes over. However, a dedicated XS worker host must have a dedicated XS standby host, that is a host with the role `xs_standby`.

In addition, to support failover an external router (for example, SAP Web Dispatcher) must be set up to route requests to the XS advanced hosts active after failover. For more information, see the section on high availability for the XS advanced runtime.

The following figure shows a multiple-host system with two dedicated XS worker hosts, one dedicated XS standby host and SAP Web Dispatcher as failover router.



Multiple-host System Configured for XS Advanced Failover

Related Information

Installing an SAP HANA System Including the XS Advanced Runtime
Setting Up the XS Advanced Runtime Behind a Reverse Proxy
Host Auto-Failover Setup with XS Advanced Run Time [page 902]
Connections for SAP HANA Extended Application Services, Advanced Model [page 736]

4.4 Server Components of the SAP HANA Database

Overview of the most important server components of the SAP HANA database and the corresponding OS processes and services

> → Tip
>
> For more information about the ports used by the components listed here, see the section on ports and connections.

Core Servers

Server Component	OS Process	Service Name	Description
Name server	`hdbnameserver`	`nameserver`	The name server, which runs in the system database only, owns the information about the topology of the SAP HANA system, including knowledge of the tenant databases that exist in the system. Information about the location of tables and table partitions is stored in the relevant tenant database catalog.
Index server	`hdbindexserver`	`indexserver`	The index server, which runs in every tenant database (but not the system database), contains the actual data stores and the engines for processing the data.
Compile server	`hdbcompileserver`	`compileserver`	The compile server performs the compilation of stored procedures and programs, for example, SQLScript procedures. It runs on every host and does not persist data. It runs in the system database and serves all tenant databases.
Preprocessor server	`hdbpreprocessor`	`preprocessor`	The preprocessor server is used by the index server to analyze text data and extract the information on which the text search capabilities are based. It runs in the system database and serves all tenant databases.
SAP Web Dispatcher	`hdbwebdispatcher`	`webdispatcher`	The Web Dispatcher processes inbound HTTP and HTTPS connections to XS classic services.
SAP start service	`sapstartsrv`	`sapstartsrv`	The SAP start service is responsible for starting and stopping the other services in the correct order. It also performs other functions, such as monitoring their runtime state.

Optional Servers

In addition to the core servers mentioned above, the following optional servers may also be running.

Server Component	OS Process	Service Name	Description
Script server	`hdbscriptserver`	`scriptserver`	The script server is used to execute application function libraries written in C++. The script server is optional and must be added manually to the database that requires it. For more information, see the section on adding a service to a tenant database.
Document store server	`hdbdocstore`	`docstore`	This server is required for the document store repository. The document store allows native operations on JSON documents and joins with other column or row store tables. The document store is optional and must be added manually to the database that requires it. For more information, see the section on adding a service to a tenant database.
XS advanced runtime	• `hdbxscontroller` • `hdbxsexeagent` • `hdbxsuaaserver`	• `xscontroller` • `xsexeagent` • `xsuaaserver`	SAP HANA includes a run-time environment for application development: **SAP HANA extended application services, advanced model** (XS advanced). The SAP HANA XS advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of previous SAP HANA extended application services, classic model (XS classic). The SAP HANA XS advanced runtime consists of several processes for platform services and for executing applications. For more information about the individual services, see the table below. The SAP HANA XS advanced runtime runs either on dedicated hosts or together with other SAP HANA components on the same host.
SAP HANA Deployment Infrastructure (HDI) server	`hdbdiserver`	`diserver`	HDI handles the deployment of design-time artifacts into the SAP HANA database. If XS advanced is installed in the system, HDI is already enabled. Otherwise, you must enable it manually. For more information, see the section on enabling HDI in the database.

Server Component	OS Process	Service Name	Description
XS classic server	`hdbxsengine`	`xsengine`	SAP HANA Extended Application Services, classic model (XS, classic) is the application server for native SAP HANA-based web applications. It is installed with the SAP HANA database and allows developers to write and run SAP HANA-based applications without the need to run an additional application server. SAP HANA XS is also used to run web-based tools that come with SAP HANA, for instance for administration, lifecycle management and development. XS classic is the original implementation of SAP HANA XS. The XS classic server can run as a separate server process or embedded within the index server. **Note** SAP HANA XS, classic and the SAP HANA repository are deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.
Extended store server	`hdbesserver`	`esserver`	The extended store server is part of SAP HANA dynamic tiering. It provides a high-performance disk-based column store for very big data up to the petabyte range. For more information, see the documentation for SAP HANA dynamic tiering on SAP Help Portal.
Data provisioning server	`hdbdpserver`	`dpserver`	The data provisioning server is part of SAP HANA smart data integration. It provides capabilities such as data provisioning in real time and batch mode, real-time data transformations, data quality functions, adapters for various types of remote sources, and an adapter SDK for developing additional adapters. For more information, see the documentation for SAP HANA smart data integration on SAP HANA Portal.
Streaming cluster	`hdbstreamingserver`	`streamingserver`	The streaming cluster is part of SAP HANA streaming analytics. Streaming analytics extends SAP HANA with capabilities of SAP Event Stream Processor for consuming data streams and complex event processing. For more information, see the documentation for streaming analytics on SAP Help Portal.

Server Component	OS Process	Service Name	Description
Accelerator for SAP ASE	`hdbetsserver`	`etsserver`	The SAP ASE server is part of SAP HANA accelerator for SAP ASE. It provides SAP Adaptive Server Enterprise (ASE) users the ability to use SAP HANA on SAP ASE data for real-time analytics. For more information, see the documentation for SAP HANA accelerator for SAP ASE on SAP Help Portal.

SAP HANA XS Advanced Services

SAP HANA includes a run-time environment for application development: **SAP HANA extended application services, advanced model** (XS advanced). The SAP HANA XS advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of previous SAP HANA extended application services, classic model (XS classic).

The SAP HANA XS advanced runtime runs either on dedicated hosts or together with other SAP HANA components on the same host. If the runtime platform of SAP HANA XS advanced is installed in your system, the following additional services run in the system database for platform services and for executing applications:

Server Component	OS Process	Service Name	Description
SAP HANA XS Controller	`hdbxscontroller`	`xscontroller`	The Controller is the central management component of SAP HANA XS advanced. For example, it has a view on all deployed and/or running applications, and persists configuration and status information in the database. The Platform Router instance is managed by the `xscontroller` service. The Platform Router, which is realized by an SAP Web Dispatcher instance, exposes the public endpoint for the entire system.
SAP HANA XS Execution Agent	`hdbxsexeagent`	`xsexecagent`	The Execution Agent is responsible for managing processes, that is starting, keeping alive, and stopping tasks.
SAP HANA XS User Authentication and Authorization (UAA)	`hdbxsuaaserver`	`xsuaaserver`	The UAA service manages user logon and logoff requests in SAP HANA XS advanced.

→ Recommendation

SAP recommends that customers and partners who want to develop new applications use SAP HANA XS advanced model. If you want to migrate existing SAP HANA XS classic applications to run in the new SAP HANA XS advanced run-time environment, SAP recommends that you first check the features available with the installed version of SAP HANA XS advanced; if the SAP HANA XS advanced features match the

requirements of the SAP HANA XS classic application you want to migrate, then you can start the migration process. For more information, see the *SAP HANA XS Advanced Migration Guide*.

Related Information

Ports and Connections [page 718]
The JSON Document Store [page 258]
Add Services in a Tenant Database
SAP HANA Deployment Infrastructure
SAP Note 1650957
SAP HANA XS Advanced Migration Guide
SAP Note 2465027
SAP HANA Dynamic Tiering
SAP HANA Streaming Analytics
SAP HANA Accelerator for SAP ASE
SAP HANA Smart Data Integration and SAP HANA Smart Data Quality

4.5 System Limitations

Limitations to take into consideration when administering an SAP HANA database.

Aside from the table below, most system limits can also be viewed by querying the M_SYSTEM_LIMITS system view (`SELECT * FROM M_SYSTEM_LIMITS;`). However, your values might differ depending on the hardware and software configuration your system uses.

For details of sizing and limitations regarding HDI containers, refer to the topic SAP HDI Sizing and Limitations in the guide *SAP HANA Deployment Infrastructure (HDI) Reference for SAP HANA Platform* (see Related Information below).

Limitation Area	Limit	M_SYSTEM_LIMITS view name for the limitation
Database size limit	Row Store: 1,945 GB Column Store: Dependent on size of physical memory	MAXIMUM_SIZE_OF_ROW_STORE
Number of locks	Unlimited for record locks, 16,383 for table locks	MAXIMUM_NUMBER_OF_TABLE_LOCKS
Number of sessions	65,536	MAXIMUM_NUMBER_OF_SESSIONS

Schema Limitations		
Number of schemas per SAP HANA instance	Maximum value of BIGINT data type	
Identifier length	127 bytes	MAXIMUM_LENGTH_OF_IDENTIFIER
Length of an alias name	128 characters	MAXIMUM_LENGTH_OF_ALIAS_NAME
Table name length	Same as Identifier length	MAXIMUM_LENGTH_OF_IDENTIFIER
Column name length	Same as Identifier length	MAXIMUM_LENGTH_OF_IDENTIFIER
Length of a string literal	8 MB	MAXIMUM_LENGTH_OF_STRING_LITERAL
Number of hex characters in a binary literal	8,192 Bytes	MAXIMUM_LENGTH_OF_BINARY_LITERAL

Tables and View Limitations		
Number of columns in a table	64,000 This limit can vary based on context, for example, in the context of virtual tables, SAP HANA may be limited by the capabilities of the remote system and the limit of the other DBMS may apply instead. In cases such as this, the limit that is met first becomes the actual limit.	MAXIMUM_NUMBER_OF_COLUMNS_IN_TABLE
Number of columns in a row table	1,000	MAXIMUM_NUMBER_OF_COLUMNS_IN_ROW_TABLE
Number of columns in a view	64,000	MAXIMUM_NUMBER_OF_COLUMNS_IN_VIEW
Number of rows in each table	Limited by storage size RS: 1,945 GB/ sizeof(row), CS: 2,100,000,000 * number of partitions	
Length of a row	Limited by RS storage size (1,945 GB per index server)	
Size of a non-partitioned table	Limited by RS storage size (1,945 GB per index server)	
Number of partitions in a CS table	16,000	MAXIMUM_NUMBER_OF_PARTITIONS_IN_CSTABLE

Number of triggers per table per DML statement	1,024	MAXIMUM_NUMBER_OF_TRIGGERS_PER_TABLE_PER_DML
Number of records per (non-partitioned) table	2,100,000,000	
Indexes and Constraints		
Number of indexes for a table	1,023	MAXIMUM_NUMBER_OF_INDEXES_IN_TABLE
Number of primary key columns in each table	16	MAXIMUM_NUMBER_OF_COLUMNS_IN_PRIMARY_KEY
Number of primary key columns in each column store table	1,000	MAXIMUM_NUMBER_OF_COLUMNS_IN_PRIMARY_KEY_IN_COLUMN_TABLE
Number of columns in an index	16	MAXIMUM_NUMBER_OF_COLUMNS_IN_INDEX
Number of columns in a UNIQUE constraint	16	MAXIMUM_NUMBER_OF_COLUMNS_IN_UNIQUE_CONSTRAINT
Size of sum of primary key, index, UNIQUE constraint	16,384 Bytes	MAXIMUM_SIZE_OF_KEY_IN_INDEX
Number of indexes in row store	256,000	
Number of unique tokens in a fulltext index	4.294.967.295	
SQL		
Length of an SQL statement	2,147,483,648 Bytes	MAXIMUM_LENGTH_OF_SQL_STATEMENT
Depth of SQL view nesting	128	MAXIMUM_DEPTH_OF_SQL_VIEW_NESTING
Maximum depth of SQL parse tree This limitation does not show in M_SYSTEM_LIMITS unless a limit is configured to something other than 0 (no limit) using the max_parse_tree_depth parameter in `indexerver.ini`.	0 0 (unlimited)	MAXIMUM_DEPTH_OF_SQL_PARSE_TREE

Maximum depth of joins in a statement. This limitation does not show in M_SYSTEM_LIMITS unless a limit is configured to something other than 0 (no limit) using the max_join_depth parameter in `indexerver.ini`.	0	MAXIMUM_DEPTH_OF_JOINS
Number of columns in an ORDER BY	65,535	MAXIMUM_NUMBER_OF_COLUMNS_IN_ORDER_BY
Number of columns in a GROUP BY	65,535	MAXIMUM_NUMBER_OF_COLUMNS_IN_GROUP_BY
Number of elements in IN predicates	65,535	MAXIMUM_NUMBER_OF_COLUMNS_IN_IN_PREDICATE
Number of elements in SELECT clause	65,535	MAXIMUM_NUMBER_OF_OUTPUT_COLUMNS_IN_STATEMENT
Number of tables in a statement. This limitation does not show in M_SYSTEM_LIMITS unless a limit is configured to something other than 0 (no limit) using the max_table_count_in_statement parameter in indexerver.ini.	0	MAXIMUM_NUMBER_OF_TABLES_IN_STATEMENT
LOB Limitations		
Maximum size of an in-memory LOB for a column store table	2 GB	MAXIMUM_SIZE_OF_MEMORY_LOB_IN_COLUMN_STORE
Maximum size of an in-memory LOB for a row store table	2 GB	MAXIMUM_SIZE_OF_MEMORY_LOB_IN_ROW_STORE
Maximum size of a packed LOB	1,013,760 bytes	MAXIMUM_SIZE_OF_PACKED_LOB
Maximum size of a LOB on disk	4,294,967,295 bytes	MAXIMUM_SIZE_OF_DISK_LOB
Procedures		
Size of all stored procedures	1,945 GB	MAXIMUM_SIZE_OF_ALL_STORED_PROCEDURES
Size of a procedure definition	2 GB	MAXIMUM_SIZE_OF_PROCEDURE_DEFINITION

Related Information

M_SYSTEM_LIMITS System View
SAP HDI Sizing and Limitations

5 SAP HANA Administration Tools

Several tools can be used for the administration of SAP HANA.

Native Tools for SAP HANA Administration

SAP HANA Cockpit

The SAP HANA cockpit provides a single point of access to a range of tools for the administration and detailed monitoring of SAP HANA databases. For example, you can use the cockpit to start and stop systems and databases, monitor databases, configure system settings, manage users and authorizations, and do backups.

The cockpit also integrates the SAP HANA database explorer and the SAP HANA SQL analyzer. The database explorer allows you to query information about the database and view information about catalog objects, while the SQL analyzer helps you to understand and analyze the execution plans of queries.

The SAP HANA cockpit is a Web-based HTML5 user interface that you access through a browser. It runs on SAP HANA extended application services, advanced model (XS advanced). It can be used to administer and monitor SAP HANA databases running SAP HANA 1.0 SPS 12 or higher.

> **Note**
>
> An SAP HANA cockpit support pack (SP) is released with every SAP HANA platform support package stack (SPS), but additional cockpit SPs may be released between platform SPSs. The *SAP HANA Administration Guide* contains information only about the cockpit SP delivered with the corresponding platform SPS. If you have a more recent cockpit SP, refer to the documentation available at https://help.sap.com/docs/SAP_HANA_COCKPIT. For more information about the revision and maintenance strategy of the cockpit, see SAP Note 2433181.

More information

SAP HANA Cockpit [page 41]

SAP HANA Cockpit Installation and Update Guide

SAP HANA Studio

As an administrator, you use the SAP HANA studio, for example, to start and stop systems, monitor the system, configure system settings, and manage users and authorizations. Developers can use the SAP HANA studio to create content such as modeled views and stored procedures in the SAP HANA repository.

The SAP HANA studio is the legacy development environment and administration tool for SAP HANA, based on the Eclipse platform.

More information

SAP HANA Studio Administration

SAP HANA hdbsql

SAP HANA hdbsql allows you to execute SQL statements and database procedures, as well as query information about the database and catalog objects, from the command line.

SAP HANA hdbsql is a command line tool installed on the SAP HANA server. It is available at `/usr/sap/<SID>/HDB<instance>/exe`. It can be used to access databases on both local and remote computers.

More information

SAP HANA HDBSQL (Command-Line Reference)

SAP HANA XS Administration Tools

Both the SAP HANA XS classic model and the SAP HANA XS advanced model include Web-based applications to configure and maintain the basic administration-related elements of the application-development process.

The SAP HANA XS advanced cockpit can be accessed from the SAP HANA cockpit.

In addition, the SAP HANA XS advanced model provides command-line tools (`XSA` and `xs` CLI) that you can use to maintain the applications that are deployed to the XS advanced run-time environment, as well as specific elements of the run-time environment itself, for example, the components that enable it, and the users who access and use it.

More information

Maintaining the SAP HANA XS Classic Model Run Time [page 1134]

Maintaining the SAP HANA XS Advanced Model Run Time [page 1256]

SAP HANA Application Lifecycle Management

Separate graphical user interfaces are available to install, update, and uninstall products and software components in the SAP HANA XS advanced and the SAP HANA XS classic environments.

The XS Advanced Application Lifecycle Management GUI can be accessed from the SAP HANA cockpit, while the XS Classic Application Lifecycle Management GUI can be accessed from the SAP HANA studio.

In addition, the `hdbalm` command line tool can be used in the SAP HANA XS classic environment, while the `xs` CLI can be used in the SAP HANA XS advanced environment.

More information

Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]

Installing and Updating Products and Software Components in SAP HANA XS Advanced Model [page 626]

SAP HANA Lifecycle Manager

The SAP HANA database lifecycle manager (HDBLCM) is used to perform SAP HANA platform lifecycle management (LCM) tasks, including installing, updating, and configuring an SAP HANA system.

The SAP HANA HDBLCM program can be run as a graphical user interface, a command-line interface, or as Web user interface in a Web browser.

More information

About the SAP HANA Database Lifecycle Manager (HDBLCM) [page 585]

SAP HANA Hardware and Cloud Measurement Tools

The SAP HANA hardware and cloud measurement tools provide tests and reports for new single host and scale out systems to determine if the hardware you intend to use meets the minimum performance criteria required to run SAP HANA in production use.

More information

SAP HANA Hardware and Cloud Measurement Tools for Tailored Data Center Integration [page 44]

SAP Tools for SAP HANA Administration

SAP Solution Manager

Use SAP Solution Manager to perform end-to-end root cause analysis and unified alert inbox for entire landscape and business process reporting.

SAP Solution Manager is a central alerting and monitoring infrastructure running on SAP NetWeaver AS for ABAP.

More information

SAP Solution Manager for SAP HANA Administration [page 45]

SAP Landscape Manager

SAP Landscape Manager allows you to automate advanced SAP HANA operations and avoiding business downtime during maintenance activities.

It is a central landscape management solution running SAP NetWeaver AS for Java.

More information

SAP Landscape Virtualization Management, enterprise edition

SAP IT Operations Management

SAP IT Operations Analytics (ITOA) lets you get and maintain a holistic, real-time overview of complex datacenter landscapes. You can collect, process, and analyze large volumes of data to find the root causes of datacenter issues and resolve them swiftly, or to predict issues and prevent them from happening in the first place.

ITOA is an SAP HANA XSC application that runs on the SAP HANA platform with the SAP HANA Predictive Analysis Library (SAP HANA PAL).

More information

SAP IT Operations Analytics

5.1 SAP HANA Cockpit

Use the Web-based administration tool SAP HANA cockpit for the administration, monitoring and maintenance of SAP HANA systems.

The SAP HANA cockpit provides tools for the administration and monitoring of SAP HANA databases, and for development capabilities through the SAP HANA database explorer. You can manage multiple databases, each running version SAP HANA 1.0 SPS 12, or later. databases running version SAP HANA 2.0 SPS 01 or later run in multi-container mode, but you can also monitor single-container systems running earlier versions of SAP HANA.

> ⓘ Note
>
> An SAP HANA cockpit support pack (SP) is released with every SAP HANA platform support package stack (SPS), but additional cockpit SPs may be released between platform SPSs. The *SAP HANA Administration Guide* contains information only about the cockpit SP delivered with the corresponding platform SPS. If you have a more recent cockpit SP, refer to the documentation available at https://help.sap.com/docs/SAP_HANA_COCKPIT. For more information about the revision and maintenance strategy of the cockpit, see SAP Note 2433181.

What can I do with the cockpit?

The SAP HANA cockpit provides aggregate, system and database administration features, such as database monitoring, user management, and data backup. You can use the SAP HANA cockpit to start and stop systems or services, monitor the system, configure system settings, and manage users and authorizations.

Cockpit apps that allow you to manage SAP HANA options and capabilities (for example, SAP HANA dynamic tiering) are only available if the option or capability has been installed.

How can I keep an eye on the big picture?

When you first launch the cockpit, you can see system and tenant databases. (The cockpit refers to these as databases). A database is an SAP HANA system (identified by a host name and instance number) which may be a system or tenant database in a tenant (database) container, or a system in a single database container. These databases are organized into database groups - you'll only see databases belonging to the groups to which your cockpit user has been granted access. At a glance, you can see top alerts from more than one database, compare database configurations and monitor the health of multiple databases.

Whenever you like, you can drill down to perform in-depth monitoring on an individual system or tenant. In order to see alerts and other data for this individual database you need to enter database user credentials. These database user credentials must pre-exist (that is, they're already created on the database you're drilling into), and must have the system privilege CATALOG READ and SELECT on _SYS_STATISTICS. For any systems running version SAP HANA 2.0 SPS 01, or later, the cockpit database administrator has the option to enable or enforce single sign-on (SSO).

How do I get access to groups of databases?

A single `COCKPIT_ADMIN` user is created through the cockpit installation process. This user creates other cockpit users through the Cockpit Manager configuration tool, which is launched through a separate URL provided during installation.

The cockpit administrator assigns the role of `Cockpit database Administrator` to at least one cockpit user. The `Cockpit database Administrator` registers databases, again through the Cockpit Manager. When databases are registered, they're added to auto-generated database groups, based on system usage type.

Since the `Cockpit database Administrator` can't grant cockpit users access to an auto-generated database group, they must also create one or more custom database groups. They add registered databases to each group, and grant access to one or more of the cockpit users that were created by the `COCKPIT_ADMIN`. When you launch the cockpit, you're able to see all the registered databases that belong to each of the database groups to which the `Cockpit database Administrator` has granted you access.

Integrated into the cockpit is the SAP HANA database explorer. The database explorer provides the ability to query information about the database using SQL statements, as well as the ability view information about your database's catalog objects.

Related Information

SAP Note 2433181
SAP Note 2380291

5.2 SAP HANA Studio

The SAP HANA studio runs on the Eclipse platform and is both a development environment and administration tool for SAP HANA.

Administrators can use the SAP HANA studio, for example, to start and stop services, to monitor the system, to configure system settings, and to manage users and authorizations. The SAP HANA studio accesses the servers of the SAP HANA database by SQL. Developers can use the SAP HANA studio to create content such as modeled views and stored procedures. These development artifacts are stored in the repository, which is part of the SAP HANA database. The SAP HANA studio is developed in Java and based on the Eclipse platform.

The SAP HANA studio presents its various tools in the form of perspectives which are documented separately:

- Database administration and monitoring features are available primarily within the SAP HANA Administration Console perspective - refer to the guide*SAP HANA Studio*
- For information about the SAP HANA Modeler perspective refer to the *SAP HANA Modeling Guide for HANA Studio*
- For information about the SAP HANA Development perspective refer to the *SAP HANA Developer Guide (For SAP HANA Studio)*

> **Note**
>
> Depending on how you installed the studio, all features may not be available. During installation, you can specify which features you require depending on your role. For system administration, only the feature SAP HANA Studio Administration is necessary. For more information, see *SAP HANA Studio Features* in the *SAP HANA Studio Installation and Update Guide*.

Related Information

SAP HANA Studio Administration
The SAP HANA Development Perspective
Quick Tour: SAP HANA Modeler Perspective

6 SAP HANA Hardware and Cloud Measurement Tools for Tailored Data Center Integration

The SAP HANA hardware and cloud measurement tools allow you to check the interoperability of SAP HANA with your existing enterprise storage, network and server in production environments.

In addition to SAP HANA as standardized and highly optimized appliance, SAP offers the opportunity to run the SAP HANA server with a customer's preferred storage and network solutions. This option enables you to reduce hardware and operational costs through the reuse of existing hardware components and operational processes. Here an SAP HANA server means the exact same bill of material as the certified SAP HANA appliance but without storage. Certified SAP HANA servers that can be used in a TDI deployment are listed in the PAM and the SAP HANA Hardware Directory.

The SAP HANA hardware and cloud measurement tools provide tests and reports for new single host and scale out systems to determine if the hardware you intend to use meets the minimum performance criteria required to run SAP HANA in production use. The tools consist of the SAP HANA hardware and cloud measurement tool, which can be downloaded from the SAP Support Portal, and the SAP HANA hardware and cloud measurement analysis, which is available online. For more information about installing and using the tools, see the *Related Links* section.

> ⚠ Caution
>
> The test should only be used before going into production. Because the tests stress the system to the maximum it is not possible to use the tools in production systems.

Related Information

SAP Note 2493172
SAP HANA Hardware and Cloud Measurement Tools
Product Availability Matrix for SAP HANA
Certified and Supported SAP HANA Hardware Directory

7 SAP Solution Manager for SAP HANA Administration

SAP Solution Manager allows you to manage your business applications throughout their entire lifecycle. You can integrate SAP HANA into an overall operations concept supported through SAP Solution Manager, as of release 7.1, SP05.

SAP HANA is often used in conjunction with other SAP business applications. For example, an SAP ERP system might call accelerators on SAP HANA to speed up business processes, or a product such as SAP Business Warehouse is deployed on the SAP HANA database. If you are using SAP HANA in such a context, then you must manage your business application in addition to administering the in-memory database. This is best done using an integrated approach.

SAP provides you with the SAP Solution Manager application management platform as part of your maintenance agreement. You can use it to manage your business applications throughout their entire lifecycle. As of release 7.1, SP05, SAP Solution Manager supports integration with SAP HANA. You can optimize your operational processes using this combined approach. One example is root cause analysis. Let's assume you have detected a problem in an application that is deployed on SAP HANA or calls an SAP HANA accelerator. In this case, you first have to find out whether the problem is caused by the application or by the SAP HANA database. SAP Solution Manager allows you to trace a process across all included components (from the user interface to the database) to locate the source of the problem. Then, detailed analysis speeds up your resolution process.

Other examples of how SAP HANA and SAP Solution Manager can be valuably integrated in the area of system operation are the processes for monitoring and change control. If your business application is still running on a traditional database, even integrated database administration might be relevant.

Related Information

Controlling Change (example of integration with Solution Manager) [page 61]
Analyzing the Root Cause (example of integration with Solution Manager) [page 60]
Central Monitoring and Administration with SAP Solution Manager [page 60]
Connecting SAP Solution Manager to SAP HANA [page 46]
SAP Help Portal: SAP Solution Manager Documentation
SAP Support Portal: SAP Solution Manager for SAP HANA (various resources including videos)
SAP Support Portal: SAP Solution Manager Early Knowledge Transfer
SAP Community: SAP Solution Manager (wikis, blogs, news)
SAP Community (Technical Operations wiki): SAP HANA Managed System Setup in SAP Solution Manager
SAP Community (Technical Operations): Troubleshooting Guide for SAP Solution Manager
Solution Manager documentation for Landscape Management Database (LMDB)

7.1 Connecting SAP Solution Manager to SAP HANA

Configure a connection to SAP HANA in SAP Solution Manager.

If you want to use capabilities of SAP Solution Manager, you have to make sure that the two systems know each other. Prerequisite for this is the registration of the SAP HANA system in the System Landscape Directory (SLD). From there, SAP Solution Manager gets the information that the SAP HANA system exists. The communication between the systems is based on a central agent infrastructure. The pre-configured agents are delivered by SAP and deployed on the SAP HANA appliance by the hardware partner.

The configuration of the connection itself is done as part of the basic configuration of SAP Solution Manager. In the guided procedure for Managed Systems Configuration you just need to set up the correct connection, assign the right agents, enter some parameters, create required users, and do a few more configurations. After this, you can start the collaboration of SAP HANA and SAP Solution Manager.

Some of the processes in SAP Solution Manager require additional configuration to specify how they should handle the SAP HANA database. For example, you have to specify in system monitoring which metrics you want to control, or you have to define your transport landscape (development system to quality assurance system to production system) for change control.

Related Information

Configuring, Working with and Administering System Landscape Directory (SAP NetWeaver)
SAP Community: System Landscape Directory Overview
Solution Manager documentation for Landscape Management Database (LMDB)
SAP HANA Operations with SAP Solution Manager
SAP HANA Managed System Setup in SAP Solution Manager
SAP Note 1747682
SAP Note 2729787

7.1.1 Configuring an SAP HANA System to Connect to the System Landscape Directory (SLD)

You can use the SAP HANA database lifecycle manager to configure the connection parameters for the central System Landscape Directory (SLD) system.

The System Landscape Directory (SLD) serves as a central information repository for your system landscape. Data suppliers collect and send system data to SLD on a regular basis. The SLD data supplier for SAP HANA systems is implemented within the name server of the SAP HANA system. However, to enable the data collection process for your SAP HANA system, you must first configure the system's connection to the SLD. Note that the SAP HANA database lifecycle manager provides only the functionality to configure the connection to the SLD, the actual registration is performed automatically by the SLD data supplier afterward.

Related Information

Configure SLD Registration Using the Graphical User Interface [page 47]
Configure SLD Registration Using the Command-Line Interface [page 48]
Configure SLD Registration Using the Web User Interface [page 51]
Using the SAP HANA Platform LCM Tools [page 585]

7.1.1.1 Configure SLD Registration Using the Graphical User Interface

You can configure an SAP HANA system to connect to the System Landscape Directory (SLD) using the SAP HANA database lifecycle manager (HDBLCM) resident program in the graphical user interface.

Prerequisites

- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on with the required root user or system administrator user `<sid>adm` credentials.

Context

When an SAP HANA system is connected to the SLD, it can report its status and provide details and information about the system itself.For more information, see SAP Note 1673424 and SAP Note 1649323.

> **Note**
>
> To configure the SLD Proxy, use the command-line interface. See Configure SLD Registration Using the Command-Line Interface [page 48].

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.

2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

```
./hdblcmgui
```

   The SAP HANA database lifecycle manager graphical user interface appears.

> ⓘ Note
>
> To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.

3. Select *Configure System Landscape Directory Registration* from the activity options. Then select *Next*.
4. Define the required parameters. Then select *Next*.

Field Name	Description
SLD Host Name	Specifies the name of the host where the SLD system is installed.
SLD Port	Specifies the standard HTTP access port of the SLD.
SLD User Name	Specifies the user of the SLD system. It must be a user that already exists on the host where the SLD system is running.
SLD Password	Specifies the password for the SLD system.
Use HTTPS	Specifies whether or not to use HTTPS.

5. Review the summary, and select *Run* to finalize the configuration.

Related Information

Using the SAP HANA Platform LCM Tools [page 585]

7.1.1.2 Configure SLD Registration Using the Command-Line Interface

You can configure an SAP HANA system to connect to the System Landscape Directory (SLD) using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface.

Prerequisites

- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on with the required root user or system administrator user `<sid>adm` credentials.

Context

When an SAP HANA system is connected to the SLD, it can report its status and provide details and information about the system itself.For more information, see SAP Note 1673424 and SAP Note 1649323.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:
   - ```
     ./hdblcm
     ```

     and enter the index of the `configure_sld` action, or
   - Start the tool with the `configure_sld` action specified:

     ```
     ./hdblcm --action=configure_sld
     ```
3. Define the required parameters.

Field Name	Description
SLD Host Name	Specifies the name of the host where the SLD system is installed.
SLD Port	Specifies the standard HTTP access port of the SLD.
SLD User Name	Specifies the user of the SLD system. It must be a user that already exists on the host where the SLD system is running.
SLD Password	Specifies the password for the SLD system.
Use HTTPS	Specifies whether or not to use HTTPS.
SLD Proxy	Specifies whether a client proxy for the HTTP connection to the SLD server is used.
SLD Proxy Host	Specifies the SLD proxy host
SLD Proxy Port	Specifies the SLD proxy port number
SLD Proxy User	Specifies the proxy user

   For more information about parameters for SLD configuration, see Related Information.
4. Review the summary, and select *y* to finalize the configuration.

Results

You have configured the SAP HANA system to connect to the System Landscape Directory (SLD). The registration itself is performed by the SLD data supplier service.
     ```

This configuration task can also be performed in batch mode and using a configuration file. For more information about the available configuration methods, see *Using the SAP HANA Platform LCM Tools*.

> **Note**
>
> When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the hdblcm directory of the SAP HANA system, execute the following command:
>
> ```
> ./hdblcm --action=configure_sld --help
> ```

Example

The following example configures the registration in an SLD system interactively. The connection from the SLD Data Supplier service to the SLD is over HTTPS.

```
./hdblcm --action=configure_sld --sld_hostname=mysap --sld_port=50001 --
sld_username=SLDuser --sld_password=SLDpassword --https
--sld_proxy --sld_proxy_host=<sld_proxy_host> --sld_proxy_port=<sld_proxy_port>
--sld_proxy_user=<sld_proxy_user> --sld_proxy_password=<sld_proxy_user_password>
```

Related Information

https
sld_hostname
sld_port
sld_username
timeouts
sld_proxy
sld_proxy_port
sld_proxy_host
sld_proxy_user

7.1.1.3 Configure SLD Registration Using the Web User Interface

A connection to the System Landscape Directory (SLD) can be configured using the SAP HANA database lifecycle manager Web user interface.

Prerequisites

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

  > **Note**
  >
  > For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.
- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Context

When an SAP HANA system is connected to the SLD, it can report its status and provide details and information about the system itself.For more information, see SAP Note 1673424 and SAP Note 1649323.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://<hostname>:1129/lmsl/HDBLCM/<SID>/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

2. Select the *Configure System Landscape Directory Registration* tile.
3. Specify values for the following fields:

Field Name	Description
SLD Host Name	Specifies the name of the host where the SLD system is installed.
SLD Port	Specifies the standard HTTP access port of the SLD.
SLD User Name	Specifies the user of the SLD system. It must be a user that already exists on the host where the SLD system is running.
SLD Password	Specifies the password for the SLD system.
Use HTTPS Connection	Specifies whether or not to use HTTPS.

4. Review the summary, and select *Run* to finalize the configuration.

Related Information

SAPUI5 Developer Guide
SAP Note 1673424
SAP Note 1649323
Add an SAP HANA System
Using the SAP HANA Platform LCM Tools [page 585]

7.1.2 Change the Default SLD Data Supplier Configuration

The System Landscape Directory (SLD) is the central directory of system landscape information relevant for the management of your software lifecycle. Data suppliers collect and send system data to SLD on a regular basis. The SLD data supplier for SAP HANA systems is implemented within the name server.

Prerequisites

- The SLD is configured.
  For more information, see SAP Note 1018839 and the introductory section *Configuring an SAP HANA System to Connect to the System Landscape Directory*.
- You have the system privilege INIFILE ADMIN.

Context

For SAP HANA systems, the name server contains the SLD data supplier. It is configured by default to automatically transfer data to the SLD on a regular basis using the `sldreg` executable. Data is transferred in XML format in a file named `sldreg.xml`. You can change the default settings if required by modifying the `nameserver.ini` configuration file; for example, it may not be necessary to send data to the SLD frequently if your landscape is stable, or you may need to change the default save locations of the configuration and log files.

You can edit configuration files in the SAP HANA cockpit or SAP HANA studio.

> **Note**
>
> Links to additional resources related to SAP Solution Manager are listed under Related Links.

Procedure

1. Navigate to the `nameserver.ini`.
2. Add a new section **`sld`** and add those parameters whose default value you want to change.

   The following table lists the possible parameters and their default values.

   > **Note**
   >
   > Under normal circumstances, you will not need to change the default values. It should only be necessary, for example, for testing purposes or if requested as part of a support inquiry.

Key	Meaning	Default Value	Note
`enable`	Activates or deactivates the SLD data supplier	true	Allowed values are true, false. Re-enabling this parameter triggers a new generation of `sldreg.xml` and sending to the SLD system.
`enable_virtdbhome`	In a system replication landscape, enables the SAP_IdenticalDatabaseSystem association in SLD between the virtual database and the physical database configured with the parameters `sldvirtdbhome` and `sldsystemhome` in the `system_landscape_hostname_virtualization` section of the `global.ini` file	Undefined	Allowed values are true and false. This parameter facilitates the registration of an SAP HANA system replication landscape in SAP Solution Manager. With the `sr_register` command, this parameter is set to **`true`** in the primary system and to **`false`** in the secondary system(s). In the event of a takeover (`sr_takeover`), the value of this parameter is set to **`true`** in the new primary system and to **`false`** in the original primary system, thus changing the association of virtual databases to the physical databases in the new primary. For more information, see *Configuring SAP HANA for System Replication Technical Scenario in SAP Solution Manager*.
`Interval`	Specifies the frequency (in seconds) with which the `sldreg.xml` file is generated. If a newly-generated document is the same as the previous one, it is not sent to the SLD.	300	It does not make sense to enter small positive values or negative values. If you enter 0 or a negative value, data is transferred to the SLD only once. Enter a value without a "1000 separator" (for example, 1899, not 1,899 or 1.899), otherwise it is interpreted as 0.
`force_interval`	Specifies how often (in seconds) the `sldreg.xml` file must be sent to the SLD, even if the file has not changed.	43200	
`configpath`	Specifies the location of the folder that contains the configuration file `slddest.cfg` This file is a parameter for the call to `sldreg`.	`/usr/sap/<sid>/SYS/global`	Example: `/usr/sap/MPW/SYS/global`

Key	Meaning	Default Value	Note
`xmlpath`	Specifies where the file `sldreg.xml` is generated and where the `sldreg.log` log file is written `sldreg.log` is the log file of `sldreg`, and both files are parameters for the call to sldreg.	`/usr/sap/<sid>/HDB<id>/<currenthost>/trace`	Example: `/usr/sap/LRH/HDB42/velberlcm1/trace`
`profilepath`	Specifies the location of the folder that contains the `sldreg` profile `SLDREG.PFL`. This file is a parameter for the call to `sldreg`.	Undefined	Example: `/usr/sap/HGB/profile`
`timeout_connect`	Specifies the network connect timeout in seconds. This file is a parameter for the call to `sldreg`.	0 means infinite.	
`timeout_send`	Specifies the send timeout in seconds in seconds. This file is a parameter for the call to sldreg.	0 means infinite.	
`timeout_receive`	Specifies the response timeout in seconds. This file is a parameter for the call to sldreg.	0 means infinite.	

Results

The transfer of data will take place in line with your new settings.

> **Note**
>
> If errors occur in the transfer of data to the SLD, you can check the log file `sldreg.log` and the database trace for the name server with trace components SLDCollect, SLDConfig, and SLDSend.

Related Information

Configuring SAP HANA System Properties (INI Files) [page 126]
Database Trace (Basic, User-Specific, and End-to-End) [page 464]
Configuring SAP HANA for System Replication Technical Scenario in SAP Solution Manager [page 56]
SAP Note 1018839
SAP Note 2082466

7.1.3 Configuring SAP HANA for System Replication Technical Scenario in SAP Solution Manager

To model an SAP HANA system replication landscape in SAP Solution Manager as a technical scenario, you must configure SAP HANA to send the correct landscape data to the Landscape Management Database (LMDB) via the System Landscape Directory (SLD). SAP Solution Manager uses the system information managed in the LMDB.

System replication is a mechanism for ensuring the high availability of SAP HANA systems, as well as disaster recovery. Through the continuous replication of data from a primary to a secondary system, system replication facilitates rapid failover in the event of a disaster. Production operations can be resumed with minimal downtime. With multitier system replication, a third system is attached to the first secondary making it a replication chain of three systems. For more information about system replication, see the section on availability and scalability in this guide.

Such a system replication landscape can be modeled as a technical scenario in SAP Solution Manager. The SAP HANA SLD supplier sends the required system landscape information to the LMDB based on the following landscape configuration settings in SAP HANA.

> **Note**
>
> The following configuration must be in place before you enable system replication on the primary system and register secondary systems.

Landscape Configuration Settings

Define Virtual Databases (`sldvirtdbhome`)

A virtual database is the logical host name under which an SAP HANA database can always be reached for the purposes of **takeover**. This ensures that application systems remain associated with the correct database in SLD when a secondary system takes over from the primary system.

In each SAP HANA system in the system replication landscape, configure a virtual database at the system level, or if desired for each individual tenant database, by setting the parameter `[system_landscape_hostname_virtualization] sldvirtdbhome` in the `global.ini` file of the system database. This will result in each virtual database being registered in SLD as an `SAP_HDBSystem` object (`<database_name>.DBTypeForSAP.HDB.SystemHome.<sldvirtdbhome>`).

Note the following:

- If you don't configure a value in the `global.ini` of each tenant database, the value configured in the `global.ini` of the SYSTEMDB applies. If you don't configure a value in the `global.ini` of the SYSTEMDB, the value of the `SLDVIRTDBHOME` parameter in the default system profile (`hana/shared/<sid>/profile/DEFAULT.PFL`) is used; or if this value hasn't been set, the host name from the landscape properties file (`hana/shared/<sid>/lm_structure/landscapeVariables.properties`).
- Important: Configure a virtual database name for either all tenants or none. If some tenants are configured and some are not, the fallback values mentioned above are not used for those tenants without a configured value. In this case, the data supplier does not deliver this tenant completely and an error is written to the indexserver trace: `"e SLDCollect logical database hostname = <empty>"`.

- Changing the `sldvirtdbhome` parameter in the `global.ini` file does not require a restart, whereas changing the default profile does.

Define Physical Databases (`sldsystemhome`)

A physical database is the logical host name used to identify an SAP HANA database independently of internal host names for the purposes of **auto-host failover**. Since the host names of master and standby hosts are different, configuring a single physical database ensures that the same database is registered in SLD after a failover as before.

In each SAP HANA system in the system replication landscape, you configure a unique physical database at the system level, or if desired for each individual tenant database, by setting the parameter `[system_landscape_hostname_virtualization] sldsystemhome` in the `global.ini` file of the system database. This will result in each physical database being registered in SLD as an `SAP_HDBSystem` object (`<database_name>.DBTypeForSAP.HDB.SystemHome.<sldsystemhome>`).

Note the following:

- The LMDB does not support fully qualified domain names (FQDNs). We therefore recommend specifying physical databases without domain names. The data supplier will remove domains before sending physical databases to SLD.
- If you don't configure a value in the `global.ini` of each tenant database, the value configured the `global.ini` of the SYSTEMDB applies. If you don't configure a value in the `global.ini` of the SYSTEMDB, the value of the `SLDSYSTEMHOME` parameter in the default system profile (`hana/shared/<sid>/profile/DEFAULT.PFL`) is used; or if this value hasn't been set, the host name from the landscape properties file (`hana/shared/<sid>/lm_structure/landscapeVariables.properties`).
- Important: Configure a physical database for either all tenants or none. If some tenants are configured and some are not, the fallback values mentioned above are not used for those tenants without a configured value. In this case, the data supplier does not deliver this tenant completely and an error is written to the indexserver trace: `"e SLDCollect logical computer hostname = <empty>"`.
- Changing the `sldsystemhome` parameter in the `global.ini` file does not require a restart, whereas changing the default profile does.

Configure the Dependency Chain

In a system replication landscape, communication between primary and secondary systems is based on internal host names. This ensures that each site can resolve the host name of other sites and that the replication chain can switch seamlessly in the event of a takeover. To model the dependency relationship between databases in the replication chain in SLD, the physical databases must be mapped to the internal host names of the systems to which they belong.

In the `system_landscape_hostname_resolution` section of the `global.ini` file of the system database, create an entry for all internal host names in the system replication landscape and as the value, set the physical database available on the host. We recommend you do this in all systems.

The dependency relationship between databases will be modeled in SLD with the association `SAP_DatabaseReplicationDependency`.

> **Note**
>
> For more information about mapping internal host names between primary and secondary systems, see the section on host name resolution for system replication.

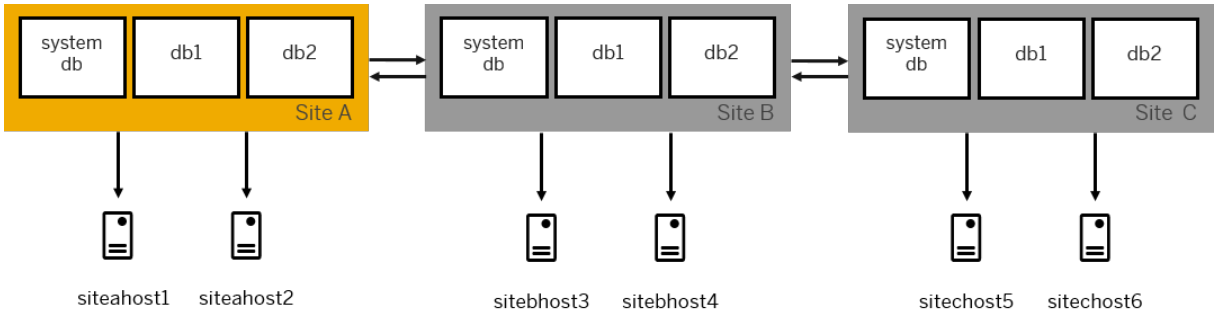
Association of Virtual and Physical Names (`enable_virtdbhome`)

Once the above configuration is in place, the virtual and physical databases must be associated. This takes place implicitly when the secondary system is registered with the primary system. With the `sr_register` command, the parameter `[sld] enable_virtdbhome` in the `nameserver.ini` file is set to **`true`** in the primary system and to **`false`** in the secondary system(s). This results in the virtual databases being associated with the physical databases of the primary system in SLD with the association `SAP_IdenticalDatabaseSystem`.

In the event of a takeover (`sr_takeover`), the value of this parameter is set to **`true`** in the new primary system and to **`false`** in the original primary system, thus changing the association of virtual databases to the physical databases in the new primary.

Example Configuration

The following figure shows a multitier system replication landscape. Each system in the chained setup is installed on two hosts for auto-host failover.



To send the required system landscape information to SLD, make the following landscape configuration settings in SAP HANA:

In all systems, configure virtual databases in the system database `global.ini` (section `system_landscape_hostname_virtualization`) as follows:

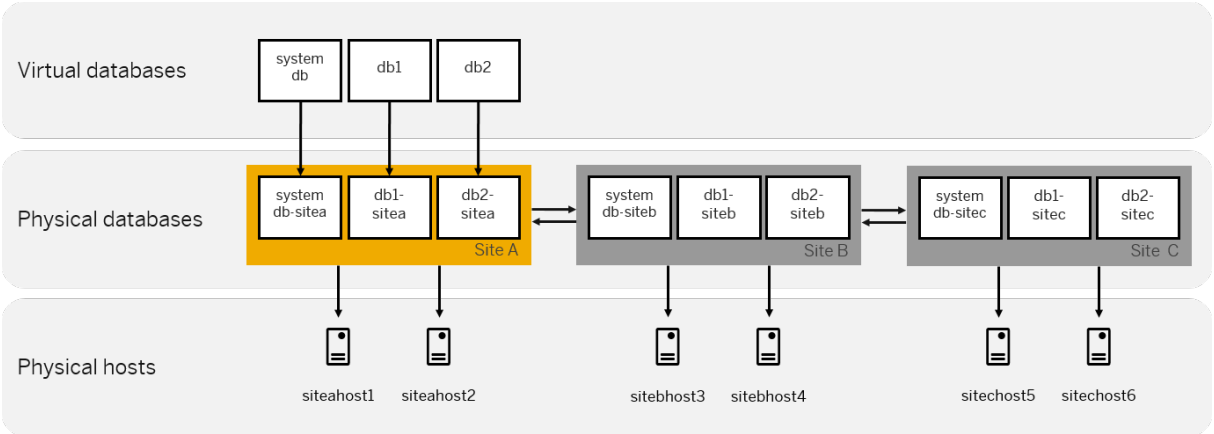
System	sldvirtdbhome Value for System	sldvirtdbhome Value for Tenant 1	sldvirtdbhome Value for Tenant 2
Primary system	systemDB	db1	db2
Tier 1 secondary system	systemDB	db1	db2
Tier 2 secondary system	systemDB	db1	db2

In all systems, configure the physical databases in the system database `global.ini` (section `system_landscape_hostname_virtualization`) as follows:

System	sldsystemhome Value for System	sldsystemhome Value for Tenant 1	sldsystemhome Value for Tenant 2
Primary system	systemDB-sitea	db1-sitea	db2-sitea
Tier 1 secondary system	systemDB-siteb	db1-siteb	db2-siteb
Tier 2 secondary system	systemDB-sitec	db1-sitec	db2-sitec

In all systems, configure the same host name resolution information in the system database `global.ini` (section `system_landscape_hostname_resolution`) as follows:

Parameter Name	Value for System	Value for Tenant 1	Value for Tenant 2
`siteahost1`	systemDB-sitea	db1-sitea	db2-sitea
`siteahost2`	systemDB-sitea	db1-sitea	db2-sitea
`sitebhost3`	systemDB-siteb	db1-siteb	db2-siteb
`sitebhost4`	systemDB-siteb	db1-siteb	db2-siteb
`sitechost5`	systemDB-sitec	db1-sitec	db2-sitec
`sitechost6`	systemDB-sitec	db1-sitec	db2-sitec



Related Information

High Availability for SAP HANA [page 756]
Host Name Resolution for System Replication [page 879]
SLD Registration for SAP HANA SR Systems

7.2 Central Monitoring and Administration with SAP Solution Manager

The monitoring and alerting infrastructure of SAP Solution Manager is based on a central agent infrastructure.

SAP delivers pre-configured agents for SAP HANA. If these agents have been deployed on SAP HANA and have been connected to SAP Solution Manager, SAP Solution Manager receives all alerts from the SAP HANA database. These alerts are displayed in the unified alert inbox.

SAP Solution Manager also provides an automated regular status check of your SAP solutions: SAP Early Watch Alert (EWA). This tool monitors the essential administrative areas of SAP components and keeps you up to date on their performance and stability. If you have any ABAP-based system in addition or running on SAP HANA, you can include the SAP HANA database information in the EWA report of this system: availability of services; size and growth; performance; important alerts; and correct setting of configuration parameters.

In addition to basic administration tasks, SAP provides specific task lists for SAP HANA, SAP Business Warehouse (BW) on SAP HANA, and SAP LT (Landscape Transformation) Replication Server. These lists describe additional activities for administering these components and provide best practices for individual tasks.

7.3 Analyzing the Root Cause of Problems

You can use SAP Solution Manager to analyze the root cause of problems in your system landscape.

Often, SAP HANA systems are integrated with business applications that either call application accelerators in the SAP HANA database or are deployed on the database. If a problem occurs in this setup, you have to take a systematic approach to identify the precise cause of the fault. This standardized process must first broadly identify the components concerned and then analyze them more closely to arrive at the cause of the fault by process of elimination (top-down approach).

End-to-end root cause analysis in SAP Solution Manager provides your support organization with tools and methods that reliably identify the affected component while minimizing the time needed to solve the problem. In addition to your business applications, it supports also your SAP HANA database. This is the foundation for resolving problems in a holistic way. The DBA Cockpit is used in SAP Solution Manager to get a detailed insight into the status of the database. This is the same data that you can see in the SAP HANA cockpit for the SAP HANA database. But the DBA Cockpit supports other databases as well. If you have heterogeneous databases in your environment because your business applications still run on traditional databases, the DBA Cockpit enables you to use the same tool for the different databases.

Trace analysis records performance-related and functional faults in a specific user activity from the user's browser to the data stored in memory or on a storage sub-system. The measurement is triggered in the user interface and automatically activates recording of the relevant traces on every component processed by the user query.

After the root cause of a problem has been located, you can use detailed analysis to find a resolution. **Workload analysis** comprises server-related workload statistics for the connected systems. **Exception analysis** allows you to centrally analyze all exceptions from connected systems, such as serious error messages in logs or dumps. From here, you can start component-specific tools. And change analysis creates transparency for all

changes (technical configuration, code, content) that have been made in the connected system landscape. This is particularly useful in the event of faults that occur once changes have been made to the production landscape.

Related Information

DBA Cockpit for SAP HANA

7.4 Controlling Change

In integrated system landscapes, SAP Solution Manager allows you to manage all changes centrally.

It is important to manage all changes in system landscapes using a central system. This is the only way to execute changes that affect more than one production system at the same time. For example, if you use SAP HANA application-specific accelerators (for example, CO-PA) in combination with your SAP ERP back-end, you might need to synchronize changes of the data models on SAP HANA with changes for accessing the changed models from the transactions on SAP ERP. SAP Solution Manager provides central transport functions for the entire system landscape as well as additional support for quality management.

The process for transporting changes from a SAP HANA development system to a quality assurance and further on to the production system starts in the SAP HANA studio. There, you retrieve content from the SAP HANA source system and export it to enhanced Change and Transport System (CTS+) in SAP Solution Manager. The exported delivery units are attached automatically to a transport request of CTS+. After you have released this transport request, CTS+ triggers the automatic activation of the change in the SAP HANA repository on the target system (quality assurance or production).

In addition to supporting the transport and activation of new or changed data models, views, stored procedures, or analytic privileges, SAP Solution Manager offers additional possibilities to improve the quality of changes. You can use change analysis and reporting for getting information about the current status and history of changes. Configuration validation checks whether the systems of your system landscape (for example, development, quality assurance, and production) are configured consistently and according to the requirements. Quality Gate Management provides an additional quality inspection for projects and ensures changes are transported correctly and synchronously to the production systems. And Change Request Management within SAP Solution Manager controls the entire change execution with detailed process and workflow control. Out of these options, you can select those for controlling changes on SAP HANA that fit best to the quality strategy of your organization.

Related Information

CTS+ How To Guides on SCN

8 System Administration

As a database administrator you are responsible for operational tasks related to the administration, monitoring, and maintenance of your SAP HANA systems.

Related Information

Aggregate Monitoring and Administration [page 62]
Starting and Stopping SAP HANA Systems [page 63]
Managing Tenant Databases [page 65]
Configuring SAP HANA System Properties (INI Files) [page 126]
Managing SAP HANA Licenses [page 137]
Monitoring the SAP HANA Database [page 139]
Managing and Monitoring SAP HANA Performance [page 148]
Managing Tables [page 244]
Workload Management [page 415]
Getting Support [page 460]

8.1 Aggregate Monitoring and Administration

Use the SAP HANA cockpit or SAP HANA studio to view high-level information about the status, availability, performance, capacity, and alert counts of all resources in your SAP HANA landscape before drilling down for database-level monitoring and administration.

Topics covered in SAP HANA cockpit in this area also include configuration of ini parameters and the SAP Early Watch Alert Service.

In SAP HANA Studio this area of administration focuses on using and customizing the System Monitor.

Related Information

Links to SAP HANA Cockpit guide:
Landscape Monitoring and Administration
Configurations and Configuration Templates
SAP EarlyWatch Alert Service
Links to SAP HANA Studio guide:
Monitoring Multiple Systems in SAP HANA Studio

8.2 Starting and Stopping SAP HANA Systems

As the operating system administrator you may need to stop, start, and restart an SAP HANA system.

The operating system administrator (<sid>adm user) can stop, start, and restart an SAP HANA system using any of the following options:

- SAP HANA cockpit from the database directory page
- SAP HANA studio in the Systems view
- Using the SAPControl program to start the system or all hosts in a scaled-out SAP HANA system

When the system restarts individual services start one by one; details of the restart sequence are given in a separate topic below. Starting the system database also starts all tenant databases though tenant databases can also be managed individually.

Related Information

Starting and Stopping Systems in SAP HANA Studio
See Services section in SAP HANA Cockpit Guide
Starting and Stopping Systems with SAPControl [page 63]
Starting and Stopping Distributed SAP HANA Systems Using SAPControl [page 1124]
Restart Sequence [page 64]
Start a Tenant Database [page 87]

8.2.1 Starting and Stopping Systems with SAPControl

You can use the SAPControl program to start or stop SAP HANA system from the command line.

> **Note**
>
> You must be logged on to the SAP system host as user <sid>adm or as a user with root permissions.

Action	Command
Start the system	`/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> -function StartSystem HDB`
Stop the system	`/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> -function StopSystem HDB`
Query current status of all hosts in the system	`/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> -function GetSystemInstanceList`

8.2.2 Restart Sequence

The SAP HANA system restart sequence restores the system to a fully operational state quickly.

When you restart an SAP HANA system, the following activities are executed by the restart agent of the persistence layer.

1. The data volume of each service is accessed in order to read and load the restart record.
2. The list of open transactions is read into memory.
3. Row tables are loaded into memory.
4. Open transactions are processed using the redo log:
   1. Write transactions that were open when the database was stopped are rolled back.
   2. Changes of committed transactions that were not written to the data area are rolled forward.
      The first column tables start being reloaded into memory as they are accessed for roll forward.

   > **Note**
   >
   > Since a regular or "soft" shutdown writes a savepoint, there are no replay log entries to be processed in this case.

   After this step, the database is technically available and logon is possible.
5. Aborted transactions are determined and rolled back.
6. A savepoint is performed with the restored consistent state of the database.
7. Column tables that are marked for preload and their attributes are asynchronously loaded in the background (if they have not already been loaded as part of log replay).
   The preload parameter is configured in the metadata of the table. This feature is useful for example to make certain tables and columns used by important business processes available more quickly.
8. Column tables that were loaded before restart and their attributes start reloading asynchronously in the background (if they have not already been loaded as part of log replay or because they are marked for preload).
   During normal operation, the system tracks the tables currently in use. This list is used as basis for reloading tables after a restart.

Reloading column tables as described in steps 7 and 8 restores the database to a fully operational state more quickly. However, it does create performance overhead and may not be necessary in non-production systems. You can deactivate the reload feature in the `indexserver.ini` file by setting the `reload_tables` parameter in the `sql` section to **`false`**. In addition, you can configure the number of tables whose attributes are loaded in parallel using the `tables_preloaded_in_parallel` parameter in the `parallel` section of `indexserver.ini`. This parameter also determines the number of tables that are preloaded in parallel.

Further Information

An SAP Community blog *SAP HANA Startup* gives further details of the restart sequence with example trace file entries.

Related Information

SAP Community Blog: SAP HANA Startup

8.3 Managing Tenant Databases

As the administrator of a tenant database system, you are responsible for creating and configuring new tenant databases, subsequently monitoring the availability and performance of databases, as well as performing certain database administration tasks.

> **Note**
>
> Administration of tenant databases is possible using the SAP HANA cockpit. However, command-line tools are required for some tasks.

> **Note**
>
> If you have SAP HANA options installed, review the section about tenant databases in the administration guide of the corresponding option for additional information before proceeding. Be aware that you need additional licenses for SAP HANA options and capabilities. For more information, see *Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities*.

Managing Audit Policies

You can change or delete audit policies for one or more tenant databases.

For more information, see *Manage Audit Policies for Tenant Databases* in *Security Administration and User Management*.

Related Information

Creating and Configuring Tenant Databases [page 65]
Monitoring and Managing Tenant Databases [page 110]
Memory and CPU Usage for Tenant Databases [page 113]
Important Disclaimer for Features in SAP HANA [page 1677]
Manage Audit Policies for Tenant Databases

8.3.1 Creating and Configuring Tenant Databases

You create tenant databases after installation if no initial tenant was created, after conversion from a single-container system to a multiple-container system, or anytime a new database is needed.

As a system administrator, you create tenant databases from the system database. You can then configure the new databases as required:

- Increase the database isolation level
- Disable certain features that are not required in tenant databases (for example, backup operations). Disabled features in tenant databases can still be accessed through the system database.

- Enable and configure cross-database access if read-only queries between tenant databases is required
- Edit the configuration change blacklist so that critical system properties cannot be changed by tenant database administrators
- Configure the SAP Web Dispatcher if tenant databases will be accessed by HTTP clients via the SAP HANA XS classic server

> **Note**
>
> Administration of tenant databases is possible using the SAP HANA cockpit. However, command-line tools are required for some tasks.

Related Information

Converting an SAP HANA System to Support Tenant Databases [page 66]
Increase the System Isolation Level [page 78]
Create a Tenant Database
Disable Features on a Tenant Database [page 90]
Enable and Configure Cross-Database Access [page 92]
Prevent Changes to System Properties in Tenant Databases [page 98]
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]
Administration of Tenant Databases [page 18]

8.3.1.1 Converting an SAP HANA System to Support Tenant Databases

You can convert an SAP HANA system to support tenant databases using the SAP HANA database lifecycle manager (HDBLCM) resident program. Converting an SAP HANA system to a tenant database system is permanent and cannot be reversed.

If your system was installed in single-container mode, you can still implement tenant databases by converting the system to a tenant database system. During the conversion process, the system database and one tenant database are created. The tenant database contains all the data of the original system, including users, system configuration, connection properties (port configuration), and system license. However, it does **not** contain the backup history.

After conversion, you can create and configure further tenant databases as needed.

> **Note**
>
> After conversion, a port offset value of 100 is used to reserve ports for system replication communication. A port offset that you defined before the conversion is not changed.

Related Information

Convert to Tenant Databases Using the Graphical User Interface [page 67]
Convert to Tenant Databases Using the Command-Line Interface [page 69]
Convert to Tenant Databases Using the Web User Interface [page 71]

8.3.1.1.1 Convert to Tenant Databases Using the Graphical User Interface

You can convert an SAP HANA system to support tenant databases using the SAP HANA database lifecycle manager (HDBLCM) resident program in the graphical user interface. Converting an SAP HANA system to a tenant database system is permanent and cannot be reversed.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

   ```
   ./hdblcmgui
   ```

   The SAP HANA database lifecycle manager graphical user interface appears.

   > **Note**
   >
   > To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.
3. Select *Convert to Tenant Databases* from the activity options. Then select *Next*.

4. Provide the password of the `<sid>adm` user and the `SYSTEM` user of `SYSTEMDB`, then select *Next*.
5. Review the summary, and select *Run* to finalize the configuration.

Results

Your SAP HANA system is a tenant database system with one system database and one tenant database, both of which are running. You can verify this by adding both databases to SAP HANA cockpit and querying the public view M_DATABASES from the system database. The result will look like this:

```
| DATABASE_NAME   |DESCRIPTION                        | ACTIVE_STATUS |
|-----------------|-----------------------------------|---------------|
| SYSTEMDB        | SystemDB-<SID>-<INSTANCE>         | YES           |
| <SID>           | SingleDB-<SID>-<INSTANCE>         | YES           |
```

Note the following about the tenant database:

- It contains all the data (including users, configuration, and connection properties) of the original system (but not the original backup history).
- Configuration files that are tenant-specific (e.g. indexserver.ini, xsengine.ini, etc.) are now stored at the following location: /usr/sap/`<SID>`/SYS/global/hdb/custom/config/DB_`<database_name>`.
- Its trace files are now stored at the following location: `/usr/sap/<SID>/HDB<instance>/<host>/trace/DB_<database_name>`.

> **Note**
>
> Any trace files that were in the trace directory before the system was converted are not moved.

Next Steps

- Create and configure any additionally required tenant databases. For more information, see *Create a Tenant Database*.

> **Note**
>
> If you configured the properties of the index server, script server, or xsengine server in your original system, these settings initially apply to **all** new tenant databases. You must explicitly configure tenant database if required. For more information, see *System Properties in Tenant Database Systems* in the *SAP HANA Administration Guide*.

- If HTTP access via the SAP HANA XS classic server is required, update the configuration of the Web Dispatcher. For more information, see *Configure HTTP Access to Tenant Databases* in the *SAP HANA Administration Guide*.

Related Information

Password Policy Configuration Options
Create a Tenant Database
Deploy a Delivery Unit Archive (*.tgz) [page 663]
Install a Permanent License
Creating Backups
Database-Specific Configuration Parameters
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]

8.3.1.1.2 Convert to Tenant Databases Using the Command-Line Interface

You can convert an SAP HANA system to support tenant databases using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface. Converting an SAP HANA system to a tenant database system is permanent and cannot be reversed.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --action=convert_to_multidb
   ```
3. Provide the password of the `<sid>adm` user and `SYSTEM` user of `SYSTEMDB` user.
4. Review the summary, and select *y* to finalize the configuration.

Results

Your SAP HANA system is a tenant database system with one system database and one tenant database, both of which are running. You can verify this by adding both databases to SAP HANA cockpit and querying the public view M_DATABASES from the system database. The result will look like this:

```
| DATABASE_NAME   |DESCRIPTION                        | ACTIVE_STATUS |
|-----------------|-----------------------------------|---------------|
| SYSTEMDB        | SystemDB-<SID>-<INSTANCE>         | YES           |
| <SID>           | SingleDB-<SID>-<INSTANCE>         | YES           |
```

Note the following about the tenant database:

- It contains all the data (including users, configuration, and connection properties) of the original system (but not the original backup history).
- Configuration files that are tenant-specific (e.g. indexserver.ini, xsengine.ini, etc.) are now stored at the following location: /usr/sap/`<SID>`/SYS/global/hdb/custom/config/DB_`<database_name>`.
- Its trace files are now stored at the following location: `/usr/sap/<SID>/HDB<instance>/<host>/trace/DB_<database_name>`.

> **Note**
>
> Any trace files that were in the trace directory before the system was converted are not moved.

Next Steps

- Create and configure any additionally required tenant databases. For more information, see *Create a Tenant Database*.

  > **Note**
  >
  > If you configured the properties of the index server, script server, or xsengine server in your original system, these settings initially apply to **all** new tenant databases. You must explicitly configure tenant database if required. For more information, see *System Properties in Tenant Database Systems* in the *SAP HANA Administration Guide*.

- If HTTP access via the SAP HANA XS classic server is required, update the configuration of the Web Dispatcher. For more information, see *Configure HTTP Access to Tenant Databases* in the *SAP HANA Administration Guide*.

Related Information

Password Policy Configuration Options
Create a Tenant Database
Deploy a Delivery Unit Archive (*.tgz) [page 663]
Install a Permanent License
Creating Backups

Database-Specific Configuration Parameters
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]
import_content [page 74]
nostart [page 75]
nostart_tenant_db [page 75]
timeouts

8.3.1.1.3 Convert to Tenant Databases Using the Web User Interface

You can convert an SAP HANA system to support tenant databases using the SAP HANA database lifecycle manager Web user interface. Converting an SAP HANA system to a tenant database system is permanent and cannot be reversed.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

> **Note**
>
> For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.

- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://`<hostname>`:1129/lmsl/HDBLCM/`<SID>`/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

2. Select the *Convert to Tenant Databases* tile.
3. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
Import Delivery Units In The System Database	Import Delivery Units In The System Database
Do Not Start Instance After Reconfiguration	Do Not Start Instance After Reconfiguration
Do Not Start Tenant Database After Reconfiguration	Do Not Start Tenant Database After Reconfiguration
Timeouts	Sets customized timeouts (`start_instance`, `stop_instance`).

4. Provide the password of the `<sid>adm` user and `SYSTEM` user of `SYSTEMDB` user, then select *Next*.
5. Review the summary, and select *Run* to finalize the configuration.

Results

Your SAP HANA system is a tenant database system with one system database and one tenant database, both of which are running. You can verify this by adding both databases to SAP HANA cockpit and querying the public view M_DATABASES from the system database. The result will look like this:

```
| DATABASE_NAME   |DESCRIPTION                        | ACTIVE_STATUS |
|-----------------|-----------------------------------|---------------|
| SYSTEMDB        | SystemDB-<SID>-<INSTANCE>         | YES           |
| <SID>           | SingleDB-<SID>-<INSTANCE>         | YES           |
```

Note the following about the tenant database:

- It contains all the data (including users, configuration, and connection properties) of the original system (but not the original backup history).
- Configuration files that are tenant-specific (e.g. indexserver.ini, xsengine.ini, etc.) are now stored at the following location: /usr/sap/`<SID>`/SYS/global/hdb/custom/config/DB_`<database_name>`.
- Its trace files are now stored at the following location: `/usr/sap/<SID>/HDB<instance>/<host>/trace/DB_<database_name>`.

> **Note**
>
> Any trace files that were in the trace directory before the system was converted are not moved.

Next Steps

- Create and configure any additionally required tenant databases. For more information, see *Create a Tenant Database*.

  > **Note**
  >
  > If you configured the properties of the index server, script server, or xsengine server in your original system, these settings initially apply to **all** new tenant databases. You must explicitly configure tenant database if required. For more information, see *System Properties in Tenant Database Systems* in the *SAP HANA Administration Guide*.

- If HTTP access via the SAP HANA XS classic server is required, update the configuration of the Web Dispatcher. For more information, see *Configure HTTP Access to Tenant Databases* in the *SAP HANA Administration Guide*.

Related Information

SAPUI5 Developer Guide
Password Policy Configuration Options
Create a Tenant Database
Deploy a Delivery Unit Archive (*.tgz) [page 663]
Install a Permanent License

Creating Backups
Database-Specific Configuration Parameters
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]

8.3.1.1.4 Parameter Reference: Converting an SAP HANA System to Support Tenant Databases

Parameters can be specified when converting an SAP HANA system to tenant databases in order to customize the configuration task.

The SAP HANA database lifecycle manager convert to multidb action also supports the following parameters:

- `batch`
- `configfile`
- `dump_configfile_template`
- `help`
- `list_systems`
- `read_password_from_stdin`
- `version`

For more information about these parameters, see the *SAP HANA Server Installation and Update Guide*

For a complete list of the parameters, call the help of the convert to multidb task with the following command:

```
./hdblcm --action=convert_to_multidb --help
```

Related Information

batch
configfile
dump_configfile_template
help
list_systems
read_password_from_stdin
version
Specifying Passwords

8.3.1.1.4.1 `import_content`

Imports delivery units.

Syntax

In the command line, the following syntax is used:

```
--import_content[=off]
```

Remarks

The default for this parameter is `--import_content`.

Related Information

SAP HANA Content [page 661]

8.3.1.1.4.2 `nostart`

Prevents the SAP HANA system from being started.

Syntax

In the command line, the following syntax is used:

```
--nostart
```

8.3.1.1.4.3 `nostart_tenant_db`

Prevents the SAP HANA tenant databases from being started.

Syntax

In the command line, the following syntax is used:

```
--nostart_tenant_db
```

8.3.1.1.5 Convert a System Replication Landscape to Support Tenant Databases

An SAP HANA system replication landscape can be converted to support tenant databases in an offline or near-zero downtime approach.

You can convert an SAP HANA single database container system that is part of a system replication configuration to support tenant databases. You have the choice to convert your SAP HANA system after taking all sites offline, or to convert in a near-zero downtime approach to minimize system downtime.

Offline Conversion

To perform an offline conversion all systems at all sites must first be stopped. Then, starting with the primary site, execute the conversion script. Once the primary site is converted and online again, continue the conversion procedure with the next site, following the replication chain.

Near-Zero Downtime Conversion

In order to carry out a near-zero downtime conversion, execute the conversion script on the secondary site. Once the conversion of this site is complete, perform a takeover. After stopping all other systems, execute the conversion script on the primary site and reregister the primary system.

Related Information

Perform an Offline Conversion [page 76]
Perform a Near-Zero Downtime Conversion [page 77]

8.3.1.1.5.1 Perform an Offline Conversion

You can perform an offline conversion of an SAP HANA system replication landscape to support tenant databases. Converting an SAP HANA system to a tenant database system is permanent and cannot be reversed.

Prerequisites

- The statistics server is **not** running as a separate server process (`statisticsserver`), but instead as an embedded service in the master index server. If this is not the case, migrate the statistics server to the embedded statistics service as described in SAP Note 1917938.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- You are logged on as the system administrator user `<sid>adm`.

Procedure

1. Stop all SAP HANA systems on all sites.
2. Run the following command on the master host of the primary system:
   `<sapmnt>/<SID>/hdblcm/hdblcm --action=convert_to_multidb`

   By default, `<sapmnt>` is `/hana/shared`.
3. Specify a new system user password.
4. Wait until the conversion has finished and the system is active again.
5. Create a data backup of the system database.
6. Repeat steps 2 through 4 on all remaining secondary systems, following the replication chain.

Related Information

SAP Note 1917938
Configuring SAP HANA System Replication [page 792]

8.3.1.1.5.2 Perform a Near-Zero Downtime Conversion

You can perform a near-zero downtime conversion of an SAP HANA system replication landscape to support tenant databases. Converting an SAP HANA system to a tenant database system is permanent and cannot be reversed.

Prerequisites

- The statistics server is **not** running as a separate server process (`statisticsserver`), but instead as an embedded service in the master index server. If this is not the case, migrate the statistics server to the embedded statistics service as described in SAP Note 1917938.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- You are logged on as the system administrator user `<sid>adm`.

Procedure

1. Run the following command on master host of the secondary system:
   `<sapmnt>/<SID>/hdblcm/hdblcm --action=convert_to_multidb`

By default, `<sapmnt>` is `/hana/shared`.

2. Specify a new system user password.
3. Wait until the conversion has finished and all systems are in sync again.
4. Stop all systems except the one you just converted.
5. Perform a takeover by the converted system.
6. Run the following command on master host of the primary system:
   `<sapmnt>/<SID>/hdblcm/hdblcm --action=convert_to_multidb --nostart`
7. Reregister the original primary system as the new secondary system.

Related Information

SAP Note 1917938
Configuring SAP HANA System Replication [page 792]

8.3.1.2 Increase the System Isolation Level

You can increase the isolation level of an existing system from low (default) to high. With high isolation, the processes of individual tenant databases run under dedicated operating system (OS) users belonging to dedicated (OS) groups and internal communication is secured.

Prerequisites

- You have root access to the SAP HANA system.
- You're logged on to the system database.
- You have the system privilege `DATABASE ADMIN`.
- Internal SAP HANA communication has been appropriately configured for TLS/SSL. `[communication] ssl` in the `global.ini` file must have the value **`systemPKI`**.

  > **Note**
  >
  > For more information, see *Secure Internal Communication* and *Server-Side TLS/SSL Configuration Properties for Internal Communication* in the *SAP HANA Security Guide*.

- If the system is running in an SAP HANA system replication configuration, the system PKI SSFS data file and key file have been copied from the primary system to the same location on the secondary systems:
  - `$DIR_INSTANCE/../global/security/rsecssfs/data/SSFS_<SID>.DAT`
  - `$DIR_INSTANCE/../global/security/rsecssfs/key/SSFS_<SID>.KEY`

Procedure

1. For every tenant database, create a dedicated OS user and group:
   a. As root user, log on to the server on which the name server of the system database is running.
   b. Create new groups for every tenant database:

      ```
      groupadd <groupname>
      ```

   c. Create new users for every tenant database, specifying `sapsys` as the primary group:

      ```
      useradd -g sapsys <username>
      ```

   d. Add every new user to the `sidshm` group and their own group as secondary groups:

      ```
      usermod -G <sid>shm,<usergroup> <username>
      ```

   > **Note**
   >
   > If the system is distributed across multiple hosts, you must create identical users and groups on every host. Users and groups must have the same names and IDs on all hosts.

2. Stop all tenant databases in the system.

   In the system database, in an SQL console, execute:

   ```
   ALTER SYSTEM STOP DATABASE <databasename>;
   ```

   > **→ Tip**
   >
   > You can also stop tenant databases in the *Database Management* app in SAP HANA cockpit.

3. Configure the system for high isolation.

   As the operating system user `<sid>adm`, log on to the server on which the master index server is running and run the following command:

   ```
   python /usr/sap/<SID>/HDB<instance>/exe/python_support/convertMDC.py --
   change=databaseIsolation --isolation=high
   ```

   This command runs the following actions:

   - Stops the system
   - Changes the value of the `[multidb] database_isolation` property in the `global.ini` file to `high`
   - Starts the system

4. Assign every database to their respective OS user and group.

   In the system database, in an SQL console, execute:

   ```
   ALTER DATABASE <databasename> OS USER '<username>' OS GROUP '<groupname>'
   ```

   > **→ Tip**
   >
   > You can also assign OS users and groups in the *Database Management* app of the SAP HANA cockpit.

5. Start all tenant databases.

In the system database, in an SQL console, execute:

```
ALTER SYSTEM START DATABASE <database_name>
```

> → Tip
>
> You can also start tenant databases in the *Database Management* app of the SAP HANA cockpit.

Results

The system is now running in high isolation mode. As a result:

- The processes of individual tenant databases run under dedicated OS users belonging to dedicated OS groups, and the processes of the system database run under the `<sid>adm` user.
- Internal database communication is authenticated using X.509 client certificates. Depending on how SSL for internal communication is configured, data communication within databases may also be encrypted. For more information about secure internal communication, see the *SAP HANA Security Guide*.
- Operations that require operating system access are restricted to users with the correct permissions. For more information, see the section on file and directory permissions with high isolation.
- New tenant databases can only be created if a dedicated OS user and group exist.

Related Information

Database Isolation [page 81]
Start a Tenant Database [page 87]
Create a Tenant Database
Assign the OS User and Group for High Isolation
Secure Internal Communication
File and Directory Permissions with High Isolation [page 83]
Isolation Level High for Backups and Third-Party Backup Tools
Server-Side TLS/SSL Configuration Properties for Internal Communication
SAP HANA Security Guide

8.3.1.2.1 Database Isolation

Every tenant database is self-contained and isolated in terms of users, database catalog, repository, logs, and so on. However, to protect against unauthorized access at the operating system (OS) level, it's possible to increase isolation further through OS user separation and authenticated communication within databases.

OS User Separation

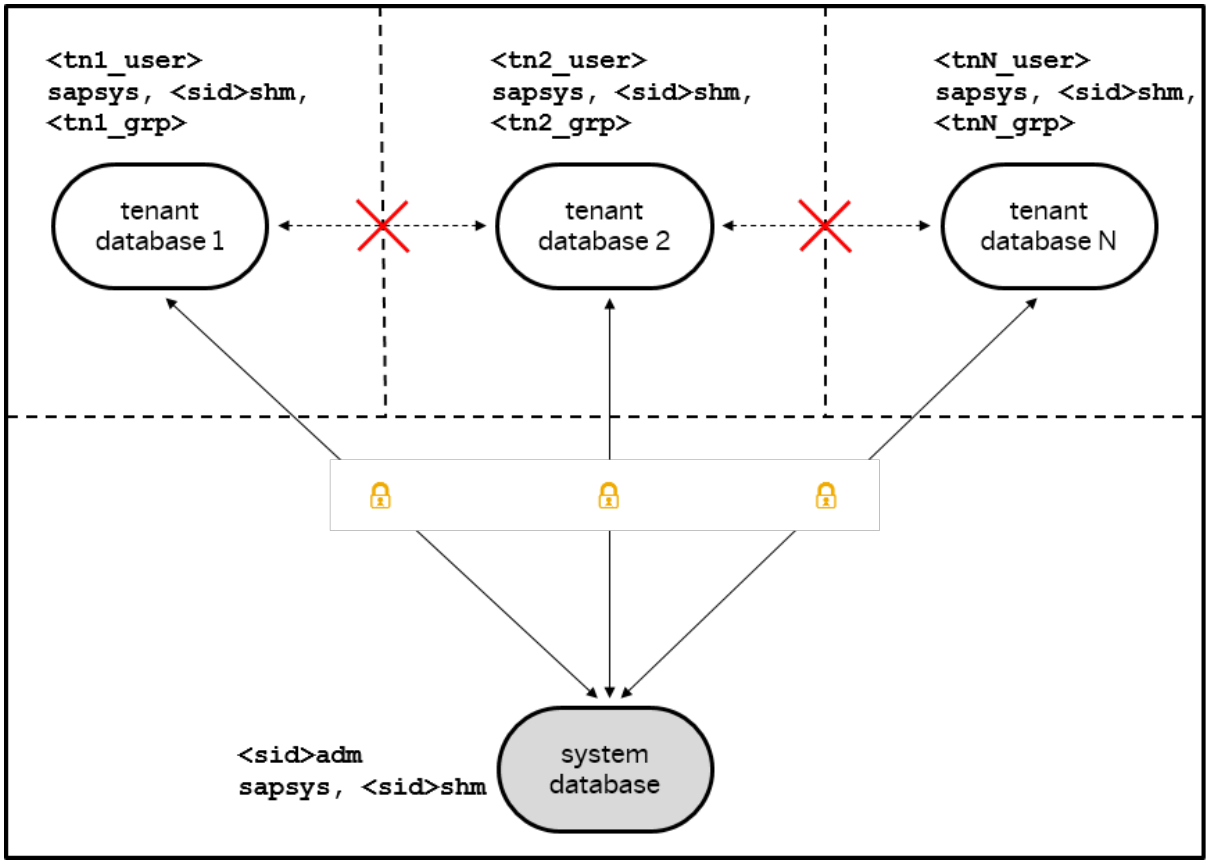
By default, all database processes run under the default OS user `<sid>adm`. If it's important to mitigate against cross-database attacks through OS mechanisms, you can configure the system for high isolation. In this way, the processes of individual tenant databases must run under dedicated OS users belonging to dedicated OS groups, instead of all database processes running under `<sid>adm`. Database-specific data on the file system is then protected using standard OS file and directory permissions.

> ⓘ Note
>
> `<sid>adm` is the OS user for the system database.

Authenticated Communication

In addition, once high isolation has been configured, internal database communication is secured using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol. Certificate-based authentication is used to ensure that only the processes belonging to the same database can communicate with each other. It's also possible to configure internal communication so that all data communication within databases is encrypted.

> ⓘ Note
>
> If cross-database access is enabled, communication between configured tenant databases is allowed.



High Database Isolation

Configuration

You can specify the isolation level of the system during installation. The default isolation level is low. It's also possible to change the isolation level of an existing system (from low to high or from high to low) at any time. For more information, see *Increase the System Isolation Level* in the *SAP HANA Administration Guide*. Once high isolation has been configured, a dedicated OS user and group must exist for every tenant database. Otherwise, it's not possible to create or start a tenant database.

Internal database communication is secured with the same mechanism used for securing other internal SAP HANA communication channels. Once high isolation has been configured, authenticated communication within databases is enabled without any change required to the default TLS/SSL configuration for internal communication. However, encryption of data communication may need to be configured explicitly.

Related Information

File and Directory Permissions with High Isolation [page 83]
Secure Internal Communication
Increase the System Isolation Level [page 78]
SAP HANA Administration Guide

8.3.1.2.2 File and Directory Permissions with High Isolation

In an SAP HANA system configured for high isolation, database-specific data on the file system is protected using standard file and directory permissions. All file and directory permissions are managed by the SAP HANA system and don't need to be set by the administrator.

System Database

The following table shows who has access to which data on the file system:

Files and Directories	Tenant OS User in Tenant OS Group	<sid>adm User
Files in directory containing system configuration files	Read permission (644)	Read permission (644)
Files in trace directory of the system database		Read and write permissions (600)
Directory containing Backint parameter file	Read permission (700)	Read permission (700)
Backint parameter file	Read and write permissions (660)	Read and write permissions (600)

Tenant Database

The following table shows who has access to which data on the file system:

> ### Note
>
> If you want to grant the system administrator access to the tenant database backup files and directories, you need to add the <sid>adm user to each tenant's operating system group.

Files and Directories	Tenant OS User in Tenant OS Group	<sid>adm User
Database-specific directories containing: • Data volumes • Log volumes • Log mirror volumes • Backups	Read, write, and execute permissions (770)	
Database-specific directories containing: • Configuration (*ini) files • Trace files	Read, write, and execute permissions (770)	Read, write, and execute permissions (770)

Files and Directories	Tenant OS User in Tenant OS Group	<sid>adm User
Files in database-specific directory containing: • Configuration (*ini) files • Trace files	Read and write permissions (666)	Read and write permissions (666)
Directory containing Backint parameter file	Read, write, and execute permissions (750)	Read, write, and execute permissions (750)
Backint parameter file	Read and write permissions (640)	Read and write permissions (640)

Related Information

Working with Third-Party Backup Tools
SAP HANA Security Guide

8.3.1.3 Decrease the System Isolation Level

If you configured a system for high isolation during installation or later, you can decrease it back to the default low level if necessary. With low isolation, the processes of all databases run under the default operating system (OS) user <sid>adm.

Prerequisites

- You have root access to the SAP HANA system.
- You're logged on to the system database.
- You have the system privilege DATABASE ADMIN.

Procedure

1. Stop all tenant databases in the system.

   In the system database, execute the SQL statement `ALTER SYSTEM STOP DATABASE <databasename>`.

   > → Tip
   >
   > You can also stop tenant databases in the *Manage Databases* app of the SAP HANA cockpit.

2. Configure the system for low isolation.

   As the operating system user `<sid>adm`, log on to the server on which the master index server is running and run the following command:

   ```
   python /usr/sap/<SID>/HDB<instance>/exe/python_support/convertMDC.py --
   change=databaseIsolation --isolation=low
   ```

   This command runs the following actions:

   - Stops the system
   - Changes the value of the `[multidb] database_isolation` property in the `global.ini` file to `low`
   - Starts the system
3. Clear the assignment of OS users and groups to tenant databases.

   In the system database, in an SQL console, execute

   ```
   ALTER DATABASE <database_name> OS USER '' OS GROUP ''
   ```

   for every tenant database.

   > → Tip
   >
   > You can also clear the OS users and groups in the *Manage Databases* app of the SAP HANA cockpit.
4. Start all tenant databases.

   In the system database, in an SQL console, execute

   ```
   ALTER SYSTEM START DATABASE <database_name>
   ```

   > → Tip
   >
   > You can also start tenant databases in the *Database Management* app of the SAP HANA cockpit.

Results

The system is now running in low isolation mode again.

- The processes of all databases run under `<sid>adm`.
- Internal database communication isn't authenticated.

Related Information

Start a Tenant Database [page 87]
Stop a Tenant Database

8.3.1.4 Create a Tenant Database Using Replication

You can create tenant databases by replicating from existing tenant databases.

Prerequisites

- You have configured tenant replication.
- You've navigated to the *Database Overview* page of the system database of the source or target system. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.
- You have the system privilege `DATABASE ADMIN`.
- You have created a complete system backup for the source tenant database.

  > ⚠ Caution
  >
  > When you use the cockpit to move a tenant, the source database is deleted as part of the process. If the source is running SAP HANA 2.0 SP01 or earlier, its backups are also deleted as part of the process—you can't roll back! Before moving, SAP recommends that you run a backup, then replicate the backup to a new location.

- For the target (the database where you're putting the moved or copied database):
  - You must have the system privileges `DATABASE ADMIN`.
- If the system is configured for high isolation, the operating system (OS) user and group required for the new tenant database already exist. For more information, see *Database Isolation* in the *SAP HANA Administration Guide*.
- You are logged in to the system database (SYSTEMDB).

Procedure

1. At the top of the *Database Overview* page, click *Database Management*.
2. To create a new tenant based on an existing tenant, choose *Create Tenant* > *Create Tenant Using Replication*.
3. To create a copy of an existing tenant database, select *Copy using replication*. To remove the original tenant after the copy has been created, select *Move using replication*.
4. Select the source database and tenant.
5. Enter a name for the new tenant.

   Restrictions apply to tenant names. Alphanumerical string of uppercase alpha characters [A-Z], digits [0-9] are permitted, starting with a letter. Depending on the file system, tenant names with up to 253 characters are supported.
6. (Optional) Specify the number of the internal communication port of the listed services.

   Under *Advanced Settings*, specify the port number for each service. If you don't enter a port, it's assigned automatically based on port number availability. In multihost systems enter host and port of a service. For

more information about port number assignment, see *Connections for Tenant Databases* in the *SAP HANA Master Guide*.

7. (For high isolation system only) Enter a dedicated OS user and group for the source.
8. If configuration changes are required for the replication, warnings appear to indicate the required changes. Select *Approve* to proceed.

   You may see a warning that you do not have the privilege required to check if the databases are fully configured for tenant replication. Please contact your system administrator to make sure that tenant replication has been properly configured.
9. If prompted, enter the SAP Control credentials required for the restart of the system.
10. If a trust relationship has not yet been established between the source system and the target system, select an existing public key certificate or upload a certificate.
11. Review the summary, then choose *Copy Tenant Database* or *Move Tenant Database* to start replicating the tenant database.

Next Steps

Register the new tenant in SAP HANA Cockpit Manager so it can be managed by SAP HANA Cockpit.

Related Information

Linux Kernel Parameters
Delete a Tenant Database [page 88]
Start a Tenant Database [page 87]
Monitoring Tenant Databases in SAP HANA Cockpit
CREATE DATABASE Statement (Tenant Database Management)
SAP HANA Administration Guide
SAP HANA Master Guide
Configure Tenant Replication
Copy or Move a Tenant Database Using Replication

8.3.1.5 Start a Tenant Database

You can start tenant databases either individually, or all at once by starting the whole system.

Prerequisites

- You've navigated to the *Database Overview* page of the database you want to manage. See Getting to the Database Overview Page.

- You are logged in to the system database (SYSTEMDB).
- You have the system privilege DATABASE ADMIN or DATABASE START.

Context

For more information about how to start the whole system, see the sections on stopping and starting a database.

Procedure

1. At the top of the *Database Overview* page, click *Database Management*.
2. Click the *Start* button for the tenant you want to start.

Related Information

Start a Database
ALTER SYSTEM START DATABASE Statement (Tenant Database Management)
M_DATABASES System View
SAP HANA SQL Reference
Prevent the Start of a Tenant Database at System Startup

8.3.1.6 Delete a Tenant Database

You can delete tenant databases that are no longer required.

Prerequisites

- You've navigated to the *Database Overview* page of the database you want to manage. See Getting to the Database Overview Page.
- You are logged in to the system database (SYSTEMDB).
- You have the system privilege `DATABASE ADMIN`.

Context

If you delete a tenant database that is running SAP HANA 2.0 SPS 01 or later, you have the option to keep the backup directories of the deleted tenant. Backups can then only be removed by deleting them from the file system. If you delete a tenant database that is running an earlier version of SAP HANA, the backup directories are deleted automatically. It's therefore recommended that if you want to preserve these backup directories, you relocate them before deleting the database.

> **Note**
>
> If a tenant SAP HANA database is enabled for usage with XS advanced and mapped to an organization/space in XS advanced, then it is recommended not to use cockpit (or the SQL command line interface) to delete the tenant database but to use the XS advanced xs command line interface. This is necessary so that XS advanced is aware that the tenant database is no longer available. Note, too, that if you use the XS advanced xs CLI to delete a tenant database used by XS advanced, all data in the tenant database is lost. See also 'Maintaining Tenant Databases in XS Advanced'.

Procedure

1. At the top of the *Database Overview* page, click *Database Management*.
2. Select the tenant to delete and click *Stop*.

   Once stopped, its status changes to *Stopped*.
3. Click the ⊗ *(Delete)*icon for the tenant.
4. If this database is running SAP HANA 2.0 SPS 01 or later, choose whether to *Keep Backup Directories* or *Delete Directories* and proceed with the database deletion, or *Cancel* the database deletion. If the database is running an earlier version of SAP HANA, choose whether to *Delete Tenant* or *Cancel* the database deletion.

Results

The system commences the process to delete the database. Once deleted, the database disappears from the list. Volumes and trace files are removed.

Next Steps

If you configured the SAP Web Dispatcher to route HTTP requests to the deleted database, you need to update the configuration.

Related Information

ALTER SYSTEM STOP DATABASE Statement (Tenant Database Management)
DROP DATABASE Statement (Tenant Database Management)
Execute SQL Statements in SAP HANA Studio
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic
SAP HANA SQL and System Views Reference
Start a Tenant Database [page 87]
Maintaining Tenant Databases in XS Advanced

8.3.1.7 Disable Features on a Tenant Database

To safeguard and/or customize your system, certain features of the SAP HANA database can be disabled in tenant databases. You can do this in the SAP HANA cockpit.

Prerequisites

- The system database is registered in the SAP HANA cockpit.
- You have the system privilege INIFILE ADMIN.

Context

Some features of the SAP HANA database are not required or desirable in certain environments, in particular features that provide direct access to the file system, the network, or other resources. To maximize your control over the security of your system, you can disable these features in tenant databases, for example import and export operations or the ability to back up the database.

The system view M_CUSTOMIZABLE_FUNCTIONALITIES provides information about those features that can be disabled and their status. This view exists in both the SYS schema of every database, where it contains database-specific information, and in the SYS_DATABASES schema of the system database, where it contains information about the enablement of features in all databases.

For more information about the features that can be disabled and why, see *Restricted Features in SAP HANA Tenant Databases* in the *SAP HANA Security Guide*.

You disable features in tenant databases in the `customizable_functionalities` section of the `global.ini` file.

> **Note**
>
> All features are enabled in the system database and cannot be disabled.

Procedure

1. Determine which feature(s) you want to disable by referring to the view M_CUSTOMIZABLE_FUNCTIONALITIES (SYS) of the system database.
2. On the *Overview* page of the system database in the SAP HANA cockpit, open *Configuration of System Properties* by clicking the corresponding administration link.
3. Select the configuration file `global.ini` file and the section `customizable_functionalities`.
4. Add a new parameter for the feature that you want to disable:
   a. Specify the database on which you want to blacklist the properties.
   b. In the *Key* field, enter the name of feature that you want to disable and set the value to **`false`**.

   > **Note**
   >
   > If you want to disable the feature on all tenant databases (including any that will be created in the future), enter **`false`** as the system layer value.
5. Repeat for further features not required in the tenant database(s).
6. Restart the affected tenant database(s).

Results

The feature is disabled. You can verify this in the view M_CUSTOMIZABLE_FUNCTIONALITIES (SYS_DATABASES).

Tenant database administrators can see which features are enabled in their database using the view M_CUSTOMIZABLE_FUNCTIONALITIES (SYS).

Related Information

Start a Tenant Database [page 87]
System and Statistics Views [page 141]
Restricted Features in Tenant Databases

8.3.1.8 Enable and Configure Cross-Database Access

Read-only queries between tenant databases are supported but not enabled by default. You must first enable this feature for the system in the system database and then configure which databases may communicate with one another. You can do this in the SAP HANA cockpit.

Prerequisites

- The system database is registered in the SAP HANA cockpit.
- You have the system privilege INIFILE ADMIN.

Context

Every tenant database is self-contained with its own isolated set of database users and isolated database catalog. However, to support for example cross-application reporting, cross-database SELECT queries are possible. This means that database objects such as tables and views can be local to one database but be read by users from other databases in the same system.

So, for example, the following query would be possible:

```
SELECT *
FROM schema1.table1 AS tab1, db2.schema2.table2 as tab2
WHERE tab2.column2 = 'foobar'
```

For more information about which object types on remote databases can be accessed using this mechanism and which local object types can access remote database objects, see *Cross-Database Access*.

To allow queries between databases, you must first enable cross-database access and then specify which databases may communicate with one other. You can do this by configuring the `global.ini` configuration file in the SAP HANA cockpit.

Procedure

1. On the *Overview* page of the system database in the SAP HANA cockpit, open *Manage System Configuration*.
2. Locate the parameter `cross_database_access` in the configuration file `global.ini`.
3. Choose *Override Value* to change the parameter value to **`true`**.

   Alternatively, enable cross-database access by executing the following statement in the system database:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'SYSTEM') set
   ('cross_database_access', 'enabled')='true' WITH RECONFIGURE;
   ```
4. Enable communication from one tenant database to one or more other tenant databases by executing the following statement in the system database:

```
ALTER SYSTEM ALTER CONFIGURATION
('global.ini', 'SYSTEM') set ('cross_database_access',
'targets_for_<source_db_name>')='<target_db1>[,<target_db2>...]' WITH
RECONFIGURE;
```

> **Example**
>
> You have two databases DB1 and DB2 and you want to be able to access DB1 from DB2. So you add the parameter with`targets_for_DB2` the value **`DB1`**.

> **Note**
>
> Cross-database access is configured only in one direction. If in the above example you also want DB2 to be able to access DB1, you would have to add the parameter `targets_for_DB1` with the value **`DB2`**.

Results

Cross-database queries are now possible between the configured databases.

Next Steps

Create remote identities for those users who require cross-database access. For more information, see *Cross-Database Authorization in Tenant Databases* in the *SAP HANA Security Guide*.

In order for a user in one database to be able to run a query or create an object that references objects in another database, the user must be mapped to a sufficiently privileged user in the remote database.

Related Information

Cross-Database Access [page 93]
Troubleshooting Error Situations Related to Cross-Database Access [page 95]
Cross-Database Authorization in Tenant Databases

8.3.1.8.1 Cross-Database Access

Read-only queries between tenant databases in the same SAP HANA system are possible. This supports cross-application reporting. Cross-database access must be explicitly enabled.

Every tenant database is self-contained with its own isolated set of database users and isolated database catalog. However, to support for example cross-application reporting, cross-database SELECT queries are

possible. This means that database objects such as tables and views can be local to one database but be read by users from other databases in the same system.

The following object types on remote databases can be accessed using cross-database access:

- Schemas
- Rowstore and columnstore tables (not including virtual tables)
- SQL views (not including monitoring views)
- Graphical calculation views
  - If they only use supported object types as data sources
  - If they don't use procedure-based analytic privileges
- Synonyms

The following object types on the local tenant database can access database objects on the remote tenant database:

- SQL views
- Scripted and graphical calculation views
- Procedures
- Synonyms

The SAP HANA modeler supports modeling of graphical calculation views using tables and other graphical calculation views as data sources from different tenant databases. For more information, see *Modeling Graphical Calculation Views With Tenant Databases* in the *SAP HANA Modeling Guide (For SAP HANA Studio)*.

For more information about how to enable and configure cross-database access, see *Enable and Configure Cross-Database Access*.

Related Information

Enable and Configure Cross-Database Access [page 92]
Cross-Database Authorization in Tenant Databases (SAP HANA Security Guide)
Troubleshooting Error Situations Related to Cross-Database Access [page 95]
Workload Management and Cross-Database Queries [page 94]
Modeling Graphical Calculation Views With Tenant Databases (SAP HANA Modeling Guide)
Import/Export Catalog Objects with Dependencies for Multi-TenantDB (SAP Community Blog)

8.3.1.8.2 Workload Management and Cross-Database Queries

Cross-database queries are executed on one or more databases. The workload management settings of the tenant database executing the query or part of the query are applied.

To balance and manage different types of workload in SAP HANA (OLAP, OLTP, mixed, and internal), it is possible to classify workloads based on user and application context information and apply resource

limitations (for example, a statement memory limit). Workload classes allow SAP HANA to influence dynamic resource consumption at the session or statement level.

The execution of any plan operations of a cross-database query in a remote tenant database is subject to the resource limitations of the workload classes and mappings defined in the remote database. If multiple remote tenant databases are involved in query execution, then different limitations may apply to different portions of the execution plan.

> ⓘ Note
>
> For cross-database queries workload classes in the remote tenant database is the only way of applying resource limitations; in this case only the following set of workload class mapping properties is available:
>
> - APPLICATION USER NAME
> - CLIENT
> - APPLICATION COMPONENT NAME
> - APPLICATION COMPONENT TYPE
> - APPLICATION NAME
> - USER NAME
>
> For cross-database queries it is not possible to control resource allocation by setting user parameters; normally, you can set values for the parameters STATEMENT MEMORY LIMIT, STATEMENT THREAD LIMIT and PRIORITY on user level, but this is not supported in this case. Also global ini file configurations (statement memory limit, statement thread limit) are not supported for cross-database queries.

For more information about workload management using workload classes and workload mappings, see *Workload Management* in the *SAP HANA Administration Guide*.

Related Information

Workload Management [page 415]
Managing Workload with Workload Classes [page 439]

8.3.1.8.3 Troubleshooting Error Situations Related to Cross-Database Access

If you are using cross-database access to query data from other tenant databases in your system, some error situations may arise.

Situation 1

You are creating views, procedures, or synonyms to access objects on other tenant databases in the same system. After dropping and re-creating an object on a remote tenant database, you can no longer access

the view or procedure on the local tenant database. You get error messages such as `invalidated view` or `invalidated procedure`. You also notice that the IS_VALID column in the system views VIEWS and PROCEDURES do not accurately reflect the fact that the view or procedure is invalid. In addition, there are entries missing in the OBJECT_DEPENDENCIES system view for the affected views, procedures, or synonyms.

What's the problem?

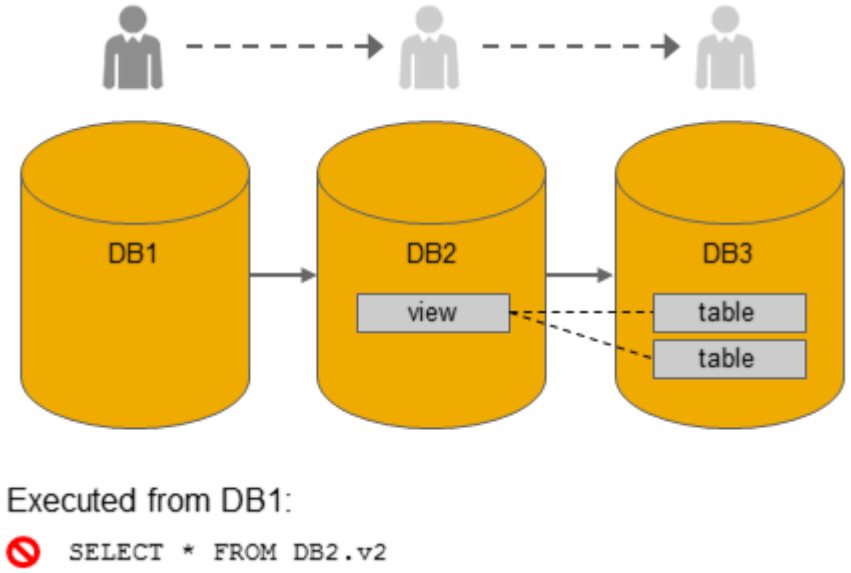
Cross-database access supports only read-only operations. Changes to an object on one tenant database cannot therefore be reflected accurately on other tenant databases that contain objects dependent on the changed object. This affects the validity flag in the relevant system views, as well as the object dependencies. Remote objects may stay valid if they retain their internal object identifier during re-creation and are re-created in an compatible way, but they will become invalid if their internal object identifier changes.

What can I do?

You need to re-create the dependent object in the local tenant database in order for it to become valid again.

Situation 2

You are querying an SQL view or a calculation view on a remote tenant database and the view itself accesses objects on a third tenant database (multi-level cross-database access). You are getting error messages such as `insufficient privilege: not authorized`. Analytic privileges on the third tenant database may be evaluated based on the wrong database user.



What's the problem?

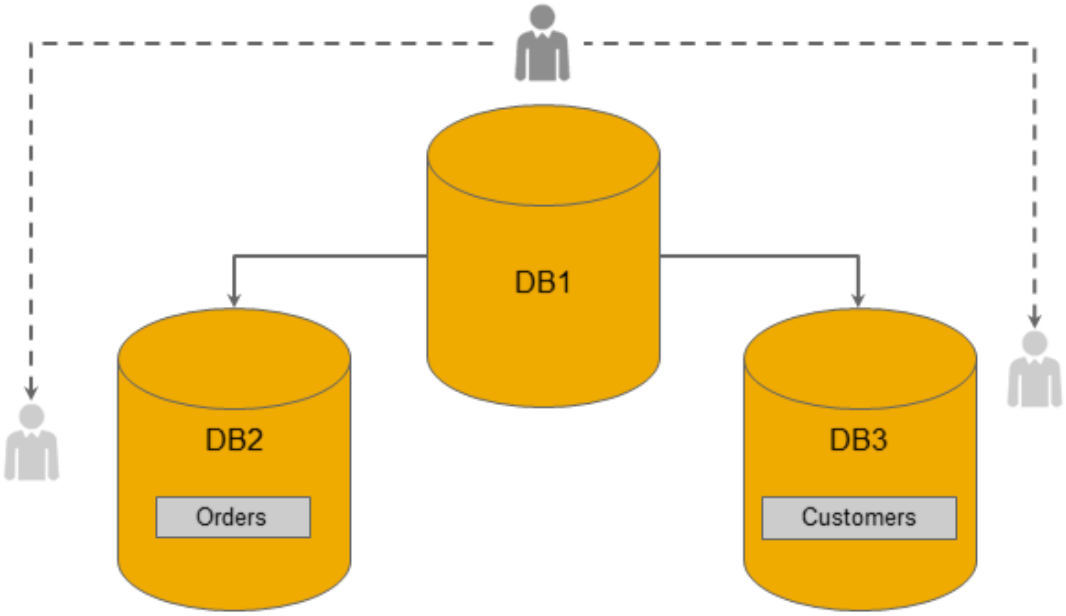
Cross-database queries do not support multiple tenant database levels as part of a view hierarchy, even if communication between databases is enabled (including the required authorized remote users).

What can I do?

Nothing. The feature is not supported.

Situation 3

Your system is running in high isolation mode. Queries that involve more than one remote tenant database run into timeouts. You are getting error messages such as `execution plan aborted` or `current operation canceled by request and transaction rolled back`. Accessing objects on remote tenant databases individually works fine.



Executed from DB1:

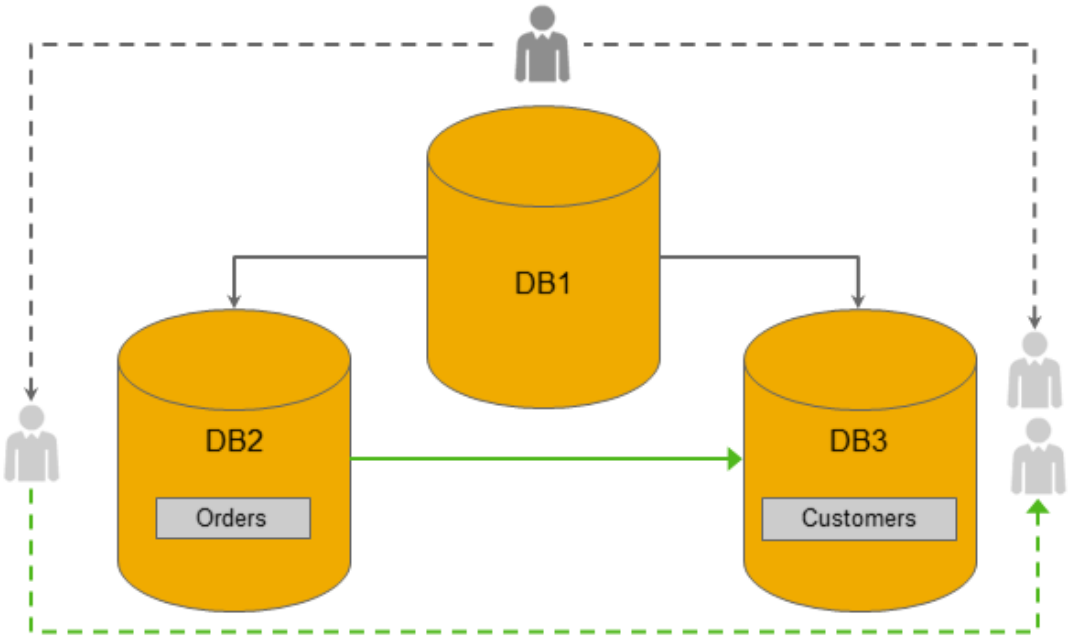
```
SELECT DB2.Orders.OrderID, DB3.Customers.CustomerName
FROM DB2.Orders
INNER JOIN DB3.Customers
ON DB2.Orders.CustomerID=DB3.Customers.CustomerID;
```

What's the problem?

The communication channels that are enabled for cross-database queries are applied in a strict fashion to the underlying network channels as well. This means that one tenant database can only open a network connection to another tenant database if communication between these two databases has been explicitly enabled. The execution plan for a query that involves objects from multiple tenant databases could however lead to direct network connections between any of the tenant databases, even if communication between them has not been explicitly enabled. This specifically applies to joins between tables on two different remote tenant databases.

What can I do?

You need to enable communication between all tenant database pairs that can potentially be involved in a query (including authorized remote users). For more information about how to do this, see *Enable and Configure Cross-Database Access*.



Executed from DB1:

```
SELECT DB2.Orders.OrderID, DB3.Customers.CustomerName
FROM DB2.Orders
INNER JOIN DB3.Customers
ON DB2.Orders.CustomerID=DB3.Customers.CustomerID;
```

Related Information

Enable and Configure Cross-Database Access [page 92]

8.3.1.9 Prevent Changes to System Properties in Tenant Databases

To ensure the stability and performance of the overall system or for security reasons, you can prevent certain system properties from being changed by tenant database administrators, for example, properties related to resource management. A configuration change blacklist is available for this purpose. You configure the blacklist in the SAP HANA cockpit.

Prerequisites

- The system database is registered in the SAP HANA cockpit.
- You have the system privileges INIFILE ADMIN.

Context

System configuration (*.ini) files have a database layer to facilitate the configuration of system properties for individual tenant databases. However, it may be desirable to prevent changes to certain properties being made directly in tenant databases because they could for example affect the performance of the system as a whole (CPU and memory management properties).

For this reason, a dedicated configuration change blacklist, `multidb.ini`, is available. This blacklist contains several critical properties by default. You can customize the default configuration, as well as add further properties by editing the file in the SAP HANA cockpit.

> **Note**
>
> Properties in the blacklist can still be configured at all levels in the system database. For more information about configuring system properties, see *Configuring SAP HANA System Properties (INI Files)*.

Procedure

1. On the *Overview* page of the system database in the SAP HANA cockpit, open *Configuration of System Properties* by clicking the corresponding administration link.
2. Select the configuration file `multidb.ini` and the section `readonly_parameters`.
3. Add a new parameter to the blacklist:
   a. Specify on which layer you want to blacklist the properties.

      You can choose from the following layers:

Layer	Result
System	Configuration not possible in any tenant database.
Database	Configuration not possible in the specified tenant database(s)

      > **Note**
      >
      > Layered configuration is possible. A lower-layer configuration overrides a higher-layer configuration. This also allows you to change the default configuration of the blacklist. The example below shows you how you could do this.

   b. In the *Key* field, enter the ini file section that contains the properties you want to blacklist.

If the section exists in more than one configuration file, you can specify the exact configuration file by entering **<file>/<section>**. If you do not specify a configuration file, the properties will be blacklisted in all files that contain the section.

For example, to specify the `communication` section in all configuration files, enter **communication**. But to specify the `communication` section in the xsengine.ini file only, enter **xsengine.ini/communication**.

c. In the *Value* field, enter the properties that you want to blacklist.

If you want to add all the properties in the section, enter *****. If you want to add all the properties in all sections of a specific file, enter **<filename>/*** (for example, **xsengine.ini/***).

d. Choose *OK*.

e. Add further parameters as required.

Results

Tenant database administrators cannot change the properties in the configuration change blacklist. If they try, they will get the error message: `Change not allowed for tenant database`. System administrators can still change the properties in the system database in all layers.

Example: Layered Configuration

The property `[sql] sql_executors` is blacklisted for all tenant databases in all configuration files by default. You could create a layered configuration for example as follows:

- You change the `sql` entry at the system layer and enter `plan_cache_size` as the value. This overrides the default configuration so that `[sql] plan_cache_size` is blacklisted instead of `[sql] sql_executors`.
- You change the `sql` entry at the system layer and enter `sql_executors` and `plan_cache_size` as the value. This overrides the default configuration so that both `[sql] plan_cache_size` and `[sql] sql_executors` are blacklisted.
- You add a new entry `indexserver.ini/sql` at the system layer with the value `plan_cache_size` as the value. This adds a specific configuration for the `indexserver.ini` file. Here, now only `[sql] plan_cache_size` is blacklisted.

Related Information

Configuring SAP HANA System Properties (INI Files) [page 126]

8.3.1.9.1 Default Blacklisted System Properties in Tenant Databases

In systems that support tenant databases, there's configuration change blacklist `multidb.ini`, which is delivered with a default configuration.

The following table lists the system properties that are included in the `multidb.ini` file by default. So, tenant database administrators can't change these properties. System administrators can still change these properties in the system database in all layers.

You can customize the default configuration change blacklist by changing existing entries in the `multidb.ini` file and adding new ones. For more information about how to prevent changes to specific system properties in tenant databases, see *Prevent Changes to System Properties in Tenant Databases* in the *SAP HANA Administration Guide*.

File/Section	Properties	Description
`auditing configuration`	• `default_audit_trail_type` • `emergency_audit_trail_type` • `alert_audit_trail_type` • `critical_audit_trail_type` • `audit_statement_length`	Prevents configuration of audit trail targets and the maximum audit statement length
`auditing_csvtextfile`	• `max_file_size` • `max_files`	Prevents configuration of the CSV text file audit trail files
`communication`	*	Prevents configuration of default key and trust stores, as well as other critical communication settings
`global.ini/customizable_functionalities`	*	Prevents disabling of restricted features
`global.ini/extended_storage`	*	Prevents configuration of extended storage (SAP HANA dynamic tiering)
`global.ini/persistence`	• `basepath_datavolumes_es` • `basepath_logvolumes_es` • `basepath_databackup_es` • `basepath_logbackup_es`	
`global.ini/system_replication`	• `keep_old_style_alert` • `enable_full_sync` • `operation_mode`	Prevents configuration of certain system replication settings
`global.ini/system_replication_communication`	*	

File/Section	Properties	Description
`global.ini/ system_replication_hostnam e_resolution`	*	
`global.ini/xb_messaging`	*	Prevents configuration of messaging
`multidb.ini/ readonly_parameters`	*	Prevents configuration of the `multidb.ini` file itself
`indexserver.ini/ authentication`	`SapLogonTicketTrustStore`	Prevents configuration of the trust store for user authentication with logon/assertion tickets
`memorymanager`	• `allocationlimit` • `minallocationlimit` • `global_allocation_limit` • `async_free_threshold` • `async_free_target`	Prevents configuration of memory allocation parameters
`execution`	`max_concurrency`	Prevents configuration of threading and parallelization parameters
`session`	• `maximum_connections` • `maximum_external_connec tions`	
`sql`	`sql_executors`	

Related Information

Prevent Changes to System Properties in Tenant Databases
Unlock Blocklisted Parameters
Copy Blocklisted Parameters

8.3.1.10 Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic

To enable Web-based applications to send HTTP(S) requests to tenant databases via the SAP HANA XS classic server, the internal SAP Web Dispatcher must be configured so it knows which requests to dispatch to which database on the basis of DNS alias host names. You do this by specifying the public URL of every tenant database in the `xsengine.ini` configuration file.

Prerequisites

- You are logged on to the system database.
- You have the system privilege INIFILE ADMIN.
- The network administrator has defined an alias hostname in your organization's Domain Name System (DNS) for every tenant database in the SAP HANA system. The alias hostname must refer to the hostname of the machine that is used for HTTP(S) access to the tenant database.
- You have a role based on the role template `sap.hana.xs.wdisp.admin::WebDispatcherAdmin`. This is required to access the SAP HANA Web Dispatcher Administration tool for configuring HTTPS.

Context

The XS classic server allows Web-based applications to access SAP HANA via HTTP(S). The internal Web Dispatcher of the SAP HANA system manages these incoming HTTP(S) requests. To allow applications to send requests to specific databases, every tenant database needs an alias host name. Requests to the alias host name can then be forwarded to the XS server of the corresponding tenant database. Requests with the physical host name in the HTTP host header are forwarded to the XS server running on the system database.

The default HTTP ports are used in all cases, that is, 80`<instance>` (HTTP) and 43`<instance>` (HTTPS). Alias host names are mapped to internal HTTP(S) ports so that incoming requests can be routed to the correct database.

You configure HTTP(S) access to tenant databases by specifying in the `xsengine.ini` file the URLs by which each tenant database is publicly accessible. The system then automatically configures the Web Dispatcher by generating the required profile entries in the `webdispatcher.ini` configuration file. It is not necessary to specify the URL of the system database, this is done automatically.

> **Note**
>
> This automatic configuration of the Web Dispatcher is controlled by the parameter `[profile] wdisp/system_auto_configuration` in the `webdispatcher.ini` configuration file. If this parameter is set to `false`, you need to configure the `webdispatcher.ini` file manually.

For HTTPS access, you must subsequently configure the required client certificates and trust stores using the SAP Web Dispatcher Administration tool. The following approaches are supported:

- Using a single "wildcard" server certificate in a single trust store that covers all databases in the system
  Wildcard certificates are more flexible when tenant databases are frequently added and deleted. However, if you use a wildcard certificate, either the server requires its own sub-domain or you must ensure that the certificate cannot be abused from other servers.

  > **Caution**
  >
  > Do not use a wildcard server certificate if strict isolation between tenant databases is required. If authentication relies on a wildcard certificate and a shared trust store, users of one tenant database will be able to log on to other databases in the system.

- Using individual certificates in individual trust stores for each database

Individual certificates for each database are more suitable in a flat domain structure for individual servers. They also ensure strict isolation between tenant databases. However, they involve more administrative effort to maintain.

Procedure

1. Specify the public URLs of all tenant databases in the xsengine.ini file in one of the following ways:

Option	Description
SAP HANA studio	1. Open the Administration editor and choose the *Configuration* tab. 2. Navigate to the `xsengine.ini` file and expand the `public_urls` section. 3. For each tenant database in the system, add the new properties `http_url` and `https_url` at the **database layer** and enter its public URL as the value: • **`http://<virtual_hostname>:80<instance>`** • **`https://<virtual_hostname>:43<instance>`** **Note** The scheme (`http/https`) must be included in the URL.
SQL	For each tenant database, execute the statements: • ALTER SYSTEM ALTER CONFIGURATION ('xsengine.ini', 'database', '`<tenant_DB_name>`') SET ('public_urls', 'http_url') = 'http://`<virtual_hostname>`:80`<instance>`' WITH RECONFIGURE; • ALTER SYSTEM ALTER CONFIGURATION ('xsengine.ini', 'database', '`<tenant_DB_name>`) SET ('public_urls', 'https_url') = 'https://`<virtual_hostname>`:43`<instance>`' WITH RECONFIGURE;

Note

The following values are set at the **default layer** and represent the URLs of the system database:

- **`http://$(SAPLOCALHOST):80$(SAPSYSTEM)`**
- **`https://$(SAPLOCALHOST):43$(SAPSYSTEM)`**

By default, the system database initially retrieves any request with the port 80`<instance_no>`. However, as soon as you configure the URLs of tenant databases, it is available under http://`<localhost>`:80`<instance>` only, and not the fully qualified domain name (FQDN). The local host is known to SAP HANA without the FQDN.

If you want to change this default behavior and configure a different URL for the system database, you can do so by executing the following statement:

```
ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini', 'system')
SET('public_urls', 'http_url') = 'http://<virtual_hostname>:80<instance>'
WITH RECONFIGURE;
```

New entries are now created in the `webdispatcher.ini` file at the host layer for every database. You can verify this by executing the following statement (from the system database):

```
SELECT KEY, VALUE, LAYER_NAME FROM SYS.M_INIFILE_CONTENTS WHERE FILE_NAME =
'webdispatcher.ini' AND SECTION = 'profile' AND KEY LIKE 'wdisp/system%'
```

This returns the following result for example:

```
KEY         | VALUE                                                | LAYER_NAME
wdisp/system_0     | GENERATED, SID=SYS, EXTSRV=http://localhost:30014,
SRCVHOST='myhost'              | DEFAULT
wdisp/system_1     | GENERATED, SID=MYD, EXTSRV=http://localhost:30042,
SRCVHOST='mydatabase.example.com'    | HOST
```

2. Optional: Secure incoming communication by configuring HTTPS.

Option	Description
Single certificate for all databases	1. Start the SAP HANA Web Dispatcher Administration tool at http://<localhost>:80<instance>/sap/hana/xs/wdisp/admin/. 2. For the default `SAPSSLS.pse` trust store, create a new SSL key pair and certificate request: 1. From the main menu, choose *SSL and Trust Configuration* > *PSE Management*. 2. From the *Manage PSE* menu, choose *SAPSSLS.pse*. 3. Choose *Recreate PSE*. 4. Enter a distinguished name that matches the host name of all tenant databases. **Example** • Physical host name: myhost.example.com • Tenant host name 1: mydatabase1.example.com • Tenant host name 2: mydatabase2.example.com In this case, you specify **`CN=*.example.com`** as the DN, thus creating a server certificate that matches all tenant databases and the system database. 5. Choose *Create*. 6. Create a certificate request and submit to your certificate authority (CA) for signing (*Create CA Response*). 3. Import the signed certificate For more information, see *Configure HTTPS (SSL) for Client Application Access*.
Individual certificates for each database	1. Start the SAP HANA Web Dispatcher Administration tool at http://<localhost>:80<instance>/sap/hana/xs/wdisp/admin/. 2. For each tenant database and the system database, create a new trust store with a unique certificate: 1. From the main menu, choose *SSL and Trust Configuration* > *PSE Management*. 2. On the PSE management screen, choose *Create New PSE*. 3. Enter a file name for the new PSE. **Example** `example.pse` 4. Enter the distinguished name: **`CN=<host name used for the tenant database in the public_urls section of the xsengine.ini file>`** 5. Choose *Create*. 6. For the new PSE, create a certificate request and submit to your CA for signing (*Create CA Response*). 7. Import the signed certificate into the new PSE (*Import CA Response*). 3. Configure the Web Dispatcher to use multiple certificates:

Option	Description
	1. In the `webdispatcher.ini` file, create or change the parameter `[profile] icm/ssl_config_0`, specifying as the value: **`ID=ssl_config_main, CRED=SAPSSLS.pse, SNI_CREDS=<semicolon (';') separated list of database PSE files>`** 2. Add **`,SSLCONFIG=ssl_config_main`** to the value of the `icm/server_port` parameter for the HTTPS port (by default `icm/server_port_1`). Example `icm/server_port_1 = PROT=HTTPS,PORT=4443$(SAPSYSTEM),PROCTIMEOUT=600, SSLCONFIG=ssl_config_main`

Results

You can access the XS server of tenant databases via the configured URLs.

→ Tip

If you experience slow response times when accessing the XS server of a tenant database (for example, Web-based applications running on the tenant database), this indicates that the server is not able to resolve the host name correctly using the DNS and retries repeatedly. If this is the case, contact your network administrator for a detailed problem analysis.

As a workaround, you can manually override virtual host name resolution on the machine where the browser is running by modifying the `/etc/hosts` file on the local machine. In this file, append a new line, starting with the static IP address of the server, followed by the virtual host name of your tenant database, for example, "10.20.30.40 mydatabase.example.com". To edit this file you need admin or root privileges.

Next Steps

Optional: Enable access to Web-based applications from the SAP HANA studio.

Some Web-based tools are accessible from the SAP HANA studio, for example, the SAP HANA cockpit and SAP HANA Lifecycle Management tool. If you want to be able to access these tools from a tenant database registered in the studio, you must specify the alias hostname in the properties. You can do this as follows:

1. In the *Systems* view, right-click the tenant database and choose *Properties*.
2. Open the *XS Properties* page and enter the alias hostname in the *XS Host* field.

Related Information

Configure HTTPS (SSL) for Client Application Access [page 1184]
Using SAP Web Dispatcher for Load Balancing with Tenant Databases [page 115]

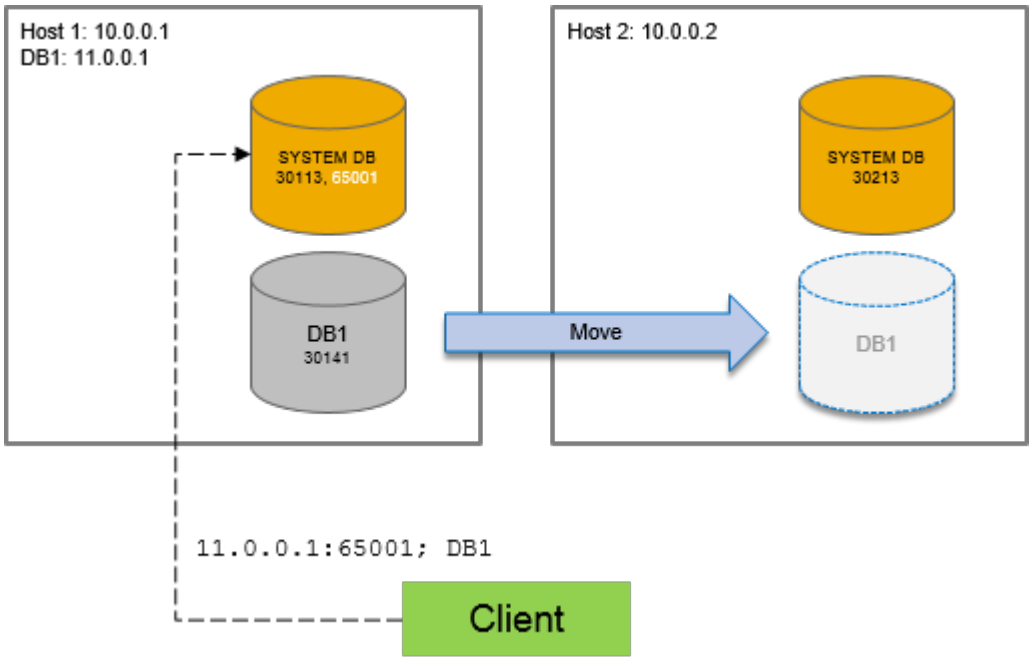
8.3.1.11 Configure Host-Independent Tenant Addresses

You can configure the access to tenant databases to be independent of the SAP HANA system ID number by mapping additional ports to a tenant database.

Context

The client connection to a tenant database is established over port 3<instance_no>13 of the system database. If a tenant database is moved to another system, the instance number of the system and consequently the connection port will change. To establish a connection independent of its current host, you can specify additional port numbers and map them to the tenants.

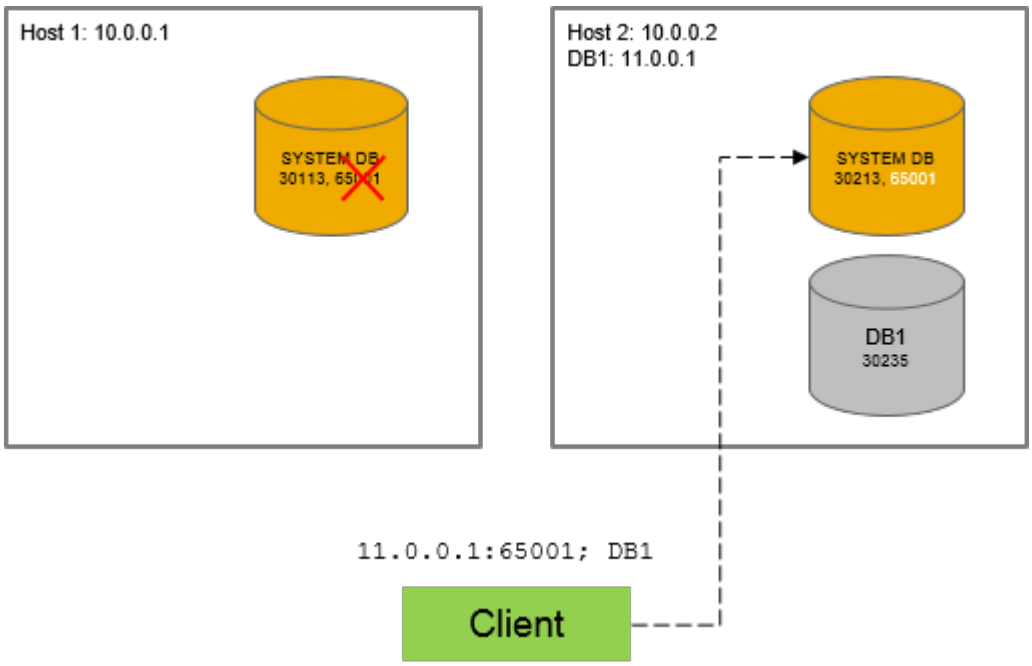
Configure a connection that is independent of the actual host IP address by mapping an IP address to each tenant database at operating system level. Add an additional port on which the system database listens for incoming connections. A client connection is then established by calling the IP address of the tenant, the name of the tenant database and the additional listen port.



Sample Code

```
SERVERNODE=11.0.0.1:65001;UID=db1user;PWD=Aa123456;DATABASENAME=DB1
```

Once the tenant was moved to the target system, the additional listen port has to be removed on the source system and added on the target system. The tenant-specific IP address must be added for the target host at operating system level. A client connection to the tenant database can still be established with the same information as before the move.



Procedure

Configure additional ports on which the system database listens in addition to port 3<instance_no>13. Any available port number except 0 is permitted.

You can do this by executing the following SQL statement:

```
ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini' , 'system') SET ('multidb' ,
'listenports' ) = '<port1>[,<port2>...]' WITH RECONFIGURE;
```

Next Steps

To remove all additional ports for a tenant database, execute the following SQL statement:

```
ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini' , 'system') UNSET
('multidb' , 'listenports' ) WITH RECONFIGURE;
```

Related Information

Copying and Moving Tenant Databases [page 671]

8.3.1.12 Reset the SYSTEM Password of a Tenant Using the Cockpit

If the password of the SYSTEM user in a tenant database is unknown, you can reset it from the system database.

Prerequisites

- You've navigated to the *Database Overview* page of the database you want to manage. See Getting to the Database Overview Page.
- There's no user available with the system privilege USER ADMIN that can reset the SYSTEM user password.

  > **Note**
  >
  > If you can log on as SYSTEM or another user with the system privilege USER ADMIN, don't use the procedure described here to change the password of the SYSTEM user. Instead, change the password using the *User* editor in SAP HANA cockpit

- You're connected to the system database and have the system privilege DATABASE ADMIN.

Procedure

1. At the top of the *Database Overview* page, click *Database Management*.
2. Select the tenant and click *Stop*.

   Once stopped, its status changes to *Not running*.
3. Click *Tenant Actions* and then *Reset SYSTEM Password*.
4. Enter and confirm a new temporary password for the SYSTEM user.
5. Select *Reset Password & Restart*.

Results

- The password for the SYSTEM user is reset and the tenant database is restarted.
- The next time you log on with the SYSTEM user, you will be prompted to change the password in line with the password policy of the tenant database
- If the SYSTEM user was previously deactivated, locked, or expired, it's now activated again. In this case, we recommend that you return it to its deactivated state.
- If auditing is enabled, the password change is automatically logged in both the system and tenant database audit trails.

Related Information

Monitoring Tenant Databases in SAP HANA Cockpit [page 111]
Resetting the SYSTEM User Password

8.3.2 Monitoring and Managing Tenant Databases

To ensure the overall performance and stability of an SAP HANA system, you as the system administrator can monitor all tenant databases in the system using the system database. You also can perform administration tasks such as stopping and starting tenant databases, or adding and removing services.

> **Note**
>
> Administration of tenant databases is possible using the SAP HANA cockpit. However, command-line tools are required for some tasks.

Support Model

The following is the general approach for analyzing and resolving issues in tenant databases:

1. Tenant database administrators analyze issues in their tenant databases using the available diagnosis and trace files.
2. If tenant database administrators discover issues that they cannot analyze using diagnosis and trace files, they contact the system administrator.
3. The system administrator can first check the health of the tenant database in the system database by analyzing the monitoring data available in the SYS_DATABASES schema.
4. If the system administrator cannot see what the problem is from the system database, the tenant database administrator needs to provide him with the necessary privileges to access the tenant database directly so that the system administrator can analyze the issue there.

Related Information

Administration of Tenant Databases [page 18]
Start a Tenant Database [page 87]
Delete a Tenant Database [page 88]
View Diagnosis Files of an Unavailable Tenant Database [page 111]
Add Services in a Tenant Database

8.3.2.1 Monitoring Tenant Databases in SAP HANA Cockpit

As the tenant database administrator, you can monitor the availability, resource usage, and performance of tenant databases in the SAP HANA cockpit from the system database.

Aggregate database information is available on the *Database Overview* page of the system database. Clicking the *Database Management* link displays information about all databases in your system and allows you to monitor and administer your tenant database. You can also create new tenant databases on this page.

You can use the cockpit to monitor and manage more than one database, each running version SAP HANA 1.0 SPS 12 or later. Any database running version SAP HANA 2.0 SPS 01 or later is set in multiple-container mode by default. The cockpit can also monitor single-container systems running earlier versions of SAP HANA. When you access the *Manage Services* application, the operations you have the option to perform depend on whether you are displaying a tenant or a system database.

> **Note**
>
> To perform operations on a tenant database, you have the system privilege DATABASE ADMIN.

Related Information

Using the Database Overview Page to Manage a Database
Database Details
Monitor Alerts for a Tenant Database

8.3.2.2 View Diagnosis Files of an Unavailable Tenant Database

If a tenant database is unavailable, for example because it's stopped or experiencing major performance problems, the tenant database administrator can't access diagnosis files. In this case, you as the system administrator can access the diagnosis files of the tenant database from the system database using the SAP HANA database explorer.

Procedure

Open the *Host Diagnostic Files* folder of your cockpit database, then click the diagnostic file that you want to examine to open it in an editor. The *Host Diagnostic Files* folder contains all diagnostic files that have been configured for the SAP Host Agent.

> **Note**
>
> You cannot open binary trace files (marked with a binary icon) in the database explorer. You can only download binary trace files.

For more information about configuring the SAP Host Agent, see the *SAP Host Agent* documentation.

The cockpit database must have valid SAP Control Credentials set in the cockpit. If the user has not set valid SAP Control Credentials, then an error is returned.

The diagnosis files of the system database are displayed.

Next Steps

If more detailed diagnosis information is required (for example for SAP Support), you can trigger the collection of a full system information dump for tenant databases. For more information, see *Collecting Diagnosis Information for SAP Support* in the *SAP HANA Administration Guide*.

Related Information

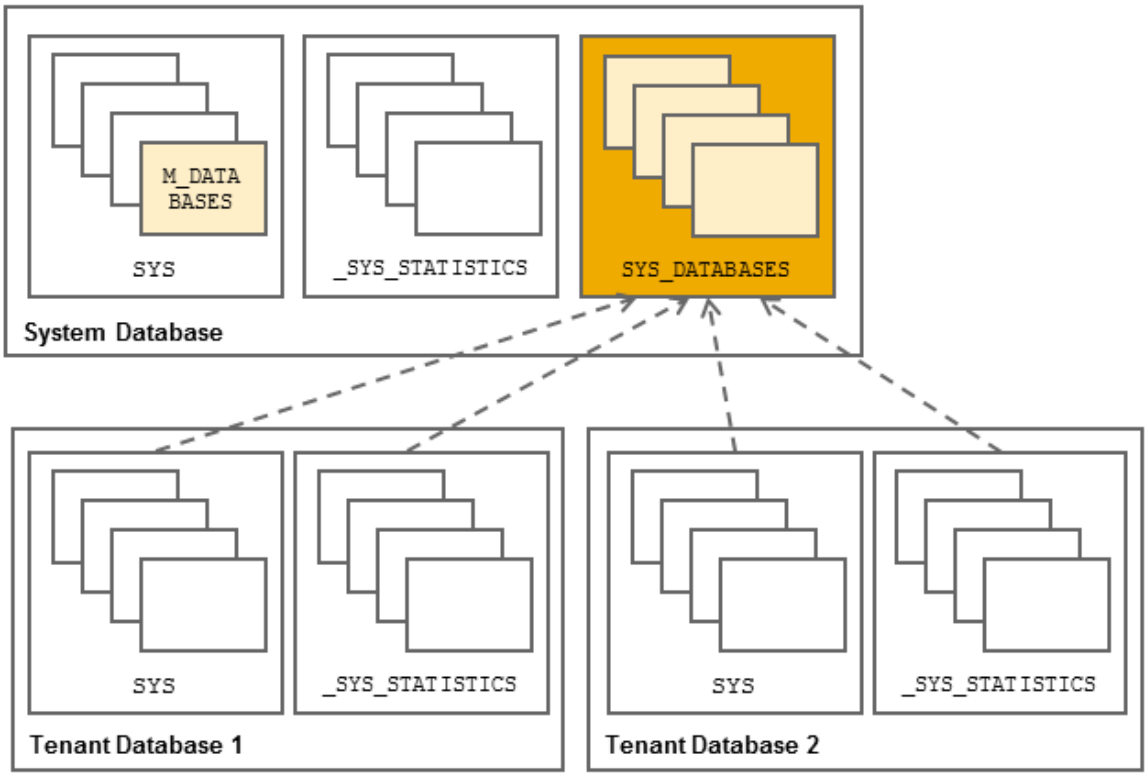
Add a Database to the SAP HANA Database Explorer
View Diagnostic Files in the SAP HANA Database Explorer
SAP Host Agent

8.3.2.3 System and Statistics Views in Tenant Database Systems

Every database has its own SYS and _SYS_STATISTICS schemas that contain information about that database only. For system-level monitoring, additional views are accessible in the system database: the M_DATABASES (SYS) view and the views in the SYS_DATABASES schema.

- M_DATABASES
  This view is available in the SYS schema of the system database of a multiple-container system. It provides an overview of all tenant databases in the system. Only users with the system privilege DATABASE ADMIN can see the contents of this view.
- SYS_DATABASES schema
  The views in the SYS_DATABASES schema provide aggregated information from a **subset** of the views available in the SYS and _SYS_STATISTICS schemas of all tenant databases in the system. These union views have the additional column DATABASE_NAME to make it possible to identify from which database the information is coming refers. The system views in the SYS_DATABASES schema are accessible only from the system database. To be able to view information in these views, you need the system privilege DATABASE ADMIN or CATALOG READ.

Tools such as the SAP HANA cockpit use these views to support system-level monitoring.



System and Statistics Views

8.3.3 Memory and CPU Usage for Tenant Databases

Manage and control the memory and CPU usage of your system by configuring limits for individual tenant databases. If necessary, you can also reserve memory for the system database.

Managing Resource Usage of Tenant Databases

Several system properties allow you to influence the allocation of memory and CPU resources in SAP HANA systems. System properties (INI) files have a database layer to facilitate the configuration of properties for individual tenant databases.

The following properties are useful for influencing the resource consumption of tenant databases.

- `[memorymanager] allocationlimit` in the `global.ini` file
  Use this property to limit the maximum amount of memory (in MB) that can be allocated individually to processes of a tenant database. Each process of a tenant database can allocate the specified value. Setting the allocation limit too low can cause the tenant database to become inaccessible until more memory can be allocated.

> **Example**
>
> Executed from the system database:
>
> ```
> ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE', 'MYDB')
> SET ('memorymanager','allocationlimit') = '8192' WITH RECONFIGURE;
> ```

> **Note**
>
> Memory alignment happens on the fly and can therefore take some time. To make it happen immediately, you can restart the database.

- `[execution] max_concurrency` in the `global.ini` file
  Use this property to influence the maximum number of CPU cores that can be used for each tenant database by limiting the number of concurrently running threads used by the JobExecutor subsystem. A reasonable default value is the number of cores divided by the number of tenant databases. Don't specify a value of `0`. A change of this value takes effect immediately.

  > **Example**
  >
  > Executed from the system database:
  >
  > ```
  > ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE', 'MYDB')
  > SET ('execution', 'max_concurrency') = '4' WITH RECONFIGURE;
  > ```

  > **Note**
  >
  > In NUMA architectures, setting the `max_concurrency` parameter isn't enough to achieve the desired performance gains, so also bind sockets that share memory using the affinity setting. For more information, see *Controlling CPU Consumption*.

Managing Memory Usage of System Database

After installation, the system database contains only data required to monitor and manage the system, as well as statistics data related to itself. The system has an average memory consumption of 15 GB.

If the system database is experiencing performance problems, for example, out-of-memory situations, you can reserve a minimum amount of memory (MB) for the system database by configuring the parameter `[multidb] systemdb_reserved_memory` in the `global.ini` file.

Related Information

Controlling Parallel Execution of SQL Statements
Configuring SAP HANA System Properties (INI Files)

8.3.4 Using SAP Web Dispatcher for Load Balancing with Tenant Databases

If an SAP HANA system has multiple instances of SAP HANA extended services, classic model (SAP HANA XS classic) and is distributed across multiple hosts, you can implement an external SAP Web Dispatcher to distribute the load of inbound HTTP requests and to enure high availability.

The following section describes how to configure an external SAP Web Dispatcher for SAP HANA systems with tenant databases.

Before You Start

Note the following points:

- The external SAP Web Dispatcher is a separate installation and does not form part of the SAP HANA system. It must have a minimum version of **745 Patch Level 21**.
- An SAP Web Dispatcher process also runs on all SAP HANA hosts on which an instance of SAP HANA XS is active. This internal SAP Web Dispatcher is a fixed part of the SAP HANA system. In a system with tenant databases, this internal SAP Web Dispatcher must also be configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Tenant Databases*.
- All information and configuration steps described in SAP Note 1855097 are still valid. In particular, the parameter `wdisp/filter_xs_internal_uri` has to be set to `false` in the `webdispatcher.ini` configuration file of your SAP HANA system.
- The configuration described in the following sections describes access to tenant databases. However, it is also valid for the system database. For the Web Dispatcher, there is no difference between tenant databases and the system database.
- The SAP Web Dispatcher handles only HTTP(S) access to SAP HANA.
- For more information about configuring secure HTTPS access, see *Configure HTTP(S) Access to Tenant Databases* (internal Web Dispatcher configuration) and *Configuring SAP Web Dispatcher to Support SSL* in the SAP HANA Web Dispatcher documentation.

Related Information

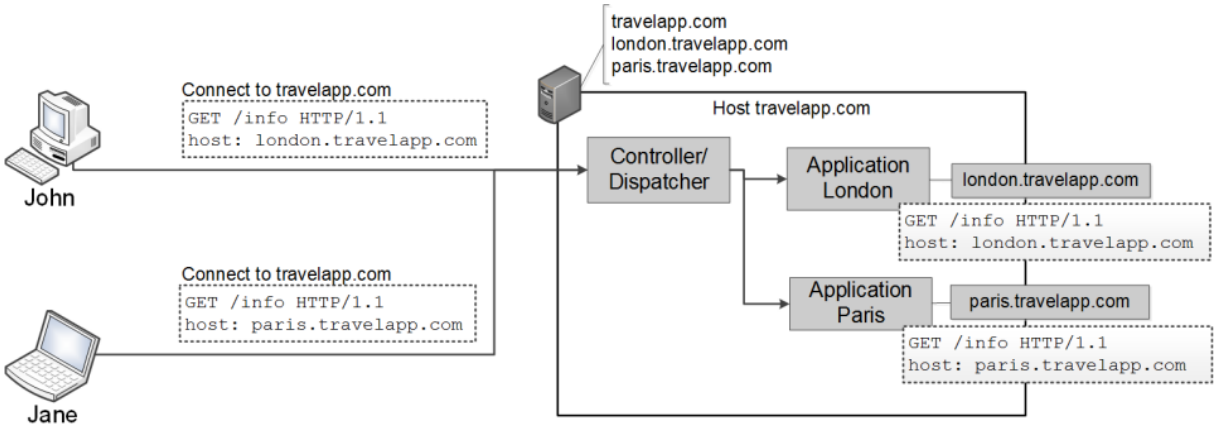
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]
Configuring SAP Web Dispatcher to Support SSL
Virtual-Host-Based Routing [page 115]
Configuring an External SAP Web Dispatcher for Tenant Databases [page 118]

8.3.4.1 Virtual-Host-Based Routing

An example explains the basics of virtual-host-based routing.

The Website `travelapp.com` provides Web-based services for information about popular travel destinations. Services are implemented as separate applications, which run on separate Web servers

on one host (`travelapp.com`). Virtual host names are used to distinguish between the available services: `london.travelapp.com` and `paris.travelapp.com`. Both virtual host names are aliases for `travelapp.com`. This can be illustrated as follows:



Virtual-Host-Based Routing

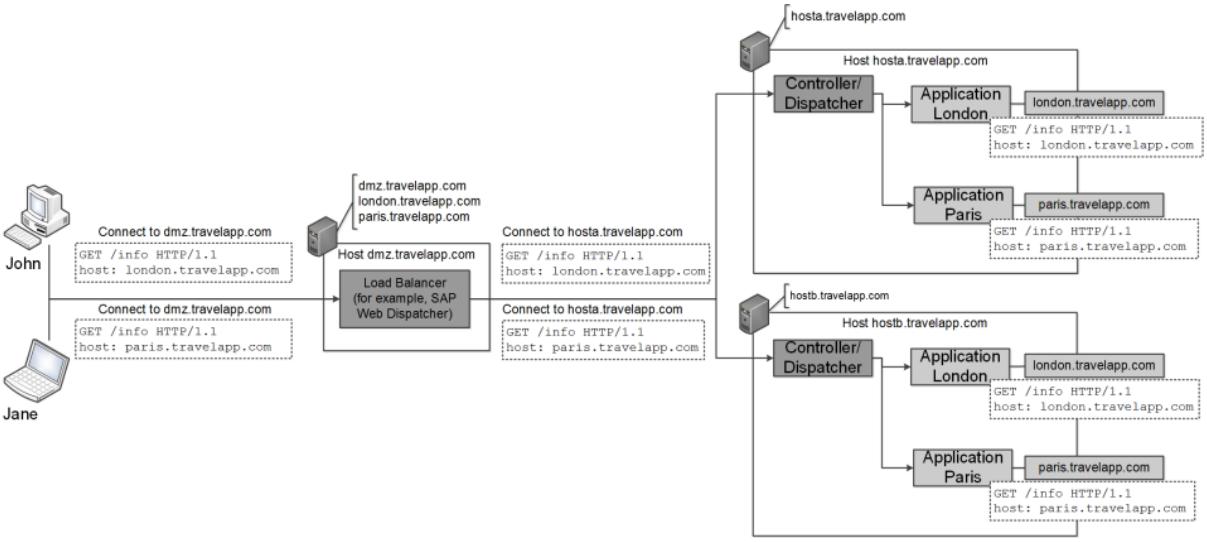
John wants to read information about London. Therefore, he enters `london.travelapp.com` into his browser. As `london.travelapp.com` is an alias for `travelapp.com`, the browser sends the HTTP request to `travelapp.com`, but it uses `london.travelapp.com` as the host header of this request. The request arrives at a controller or dispatcher process on `travelapp.com`. This dispatcher process decides whether to forward the request to the Web server responsible for displaying information about London or the Web server responsible for displaying information about Paris. This decision is made based on the host header, that is the host name that the user originally entered into the browser. `london.travelapp.com` is assigned to the application for London and `paris.travelapp.com` is assigned to the application for Paris.

Jane requires information about Paris and enters `paris.travelapp.com` into her browser. This request also arrives at the dispatcher process and is dispatched to the Paris application based on the host header of the request.

Load Balancing

`travelapp.com` has proved to be a successful service with many users. As a result, one host is no longer sufficient, and the application has been installed on a second host. In addition, a load balancer is needed to distribute requests between the two hosts. The aliases `london.travelapp.com` and

`paris.travelapp.com` have to be changed to point to the host of the load balancer to guarantee that all requests are handled by the load balancer (`dmz.travelapp.com`). This can be illustrated as follows:

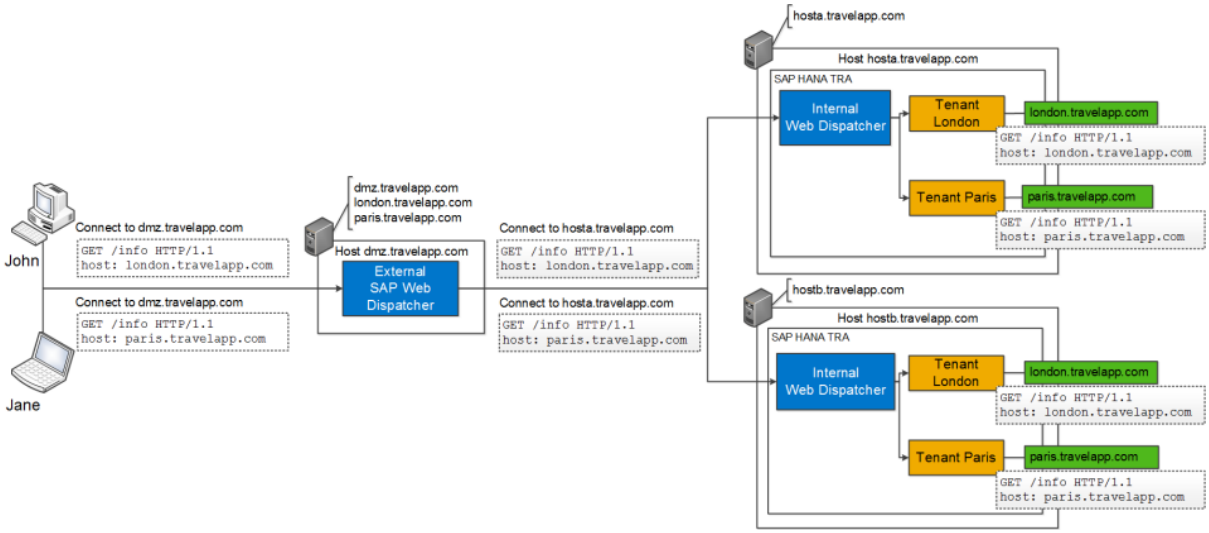


Virtual-Host-Based Routing with Load Balancing

John again wants to read information about London. Therefore, he enters `london.travelapp.com` into his browser. As `london.travelapp.com` is an alias for `dmz.travelapp.com`, the browser sends the HTTP request to `dmz.travelapp.com`, but it uses `london.travelapp.com` as the host header of this request. This request arrives at the load balancer, which simply forwards the request to `hosta.travelapp.com` or `hostb.travelapp.com` based on the current load. It must not change the host header of the request because this request is later necessary in the dispatcher. After that, the dispatcher handles the request as if no load balancer is involved, regardless of the fact that the host name in the host header actually points to another host.

SAP HANA Tenant Databases

Translated to the context of SAP HANA tenant databases, the load balancer is an external SAP Web Dispatcher, and the dispatcher is the system-internal SAP Web Dispatcher, as illustrated in the following figure:



Virtual-Host-Based Routing for SAP HANA Tenant Databases

8.3.4.2 Configuring an External SAP Web Dispatcher for Tenant Databases

Virtual host names for differentiated HTTP access to tenant databases are configured in the system-internal SAP Web Dispatcher. If you're using an external SAP Web Dispatcher for load balancing, you must also configure the external Web Dispatcher. Otherwise, information about the selected virtual hosts can't be transported to the SAP HANA system.

> → Remember
>
> All of the configuration settings mentioned here are done in the **external** Web Dispatcher and not in the internal Web Dispatcher that is part of the SAP HANA system. The external Web Dispatcher is a separate installation and does not form part of the SAP HANA system. Before you can configure the external Web Dispatcher, the internal Web Dispatcher must already have been configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Tenant Databases*.

Single Versus Multiple Tenant Access via External Web Dispatcher

Every tenant database that needs to be accessed through the external Web Dispatcher requires a `wdisp/system_<XX>` parameter entry in the external Web Dispatcher profile (`sapwebdisp.pfl`). The `XSSRV` subparameter specifies the XS server to connect to, and the `XSVHOST` subparameter specifies the virtual host name of the tenant database.

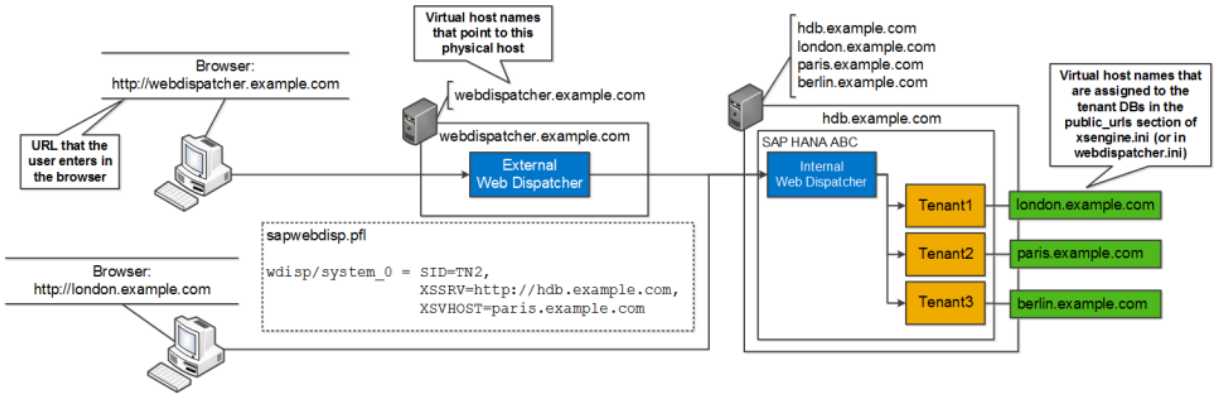
Example

```
wdisp/system_<xx> = SID=<3-digit ID>, XSSRV=http://<physical host name of SAP HANA server>:<port>, XSVHOST=<virtual host name of the tenant>
```

Note

Virtual host names are configured in the `public_urls section` of the `xsengine.ini` configuration file (or in `webdispatcher.ini`). The is part of the **internal** Web Dispatcher configuration. For more information, see *Configure HTTP(S) Access to Tenant Databases*.

If only **one tenant database** needs to be accessed through the external Web Dispatcher, a single `wdisp/system_<XX>` entry for the tenant database with the above configuration is sufficient, as depicted in the following figure:



Note

The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the XSSRV subparameter requires port numbers, and line breaks are not allowed.

Access to a Single Tenant Database

Note

An external Web Dispatcher is not mandatory to access a single tenant. But there are scenarios in which an external Web Dispatcher is required, for example sophisticated applications (for example, some SAP Fiori scenarios) or for security purposes. For more information, see the relevant application documentation and the SAP Web Dispatcher documentation.

Virtual hosts names are used to configure tenant differentiation. Two scenarios are possible:

- **Option 1**: Tenant databases are accessed via HTTP through the external Web Dispatcher only; there is no direct HTTP access to the tenant databases (recommended)
- **Option 2**: Tenants databases are accessed via HTTP both through the external Web Dispatcher and directly, bypassing the external Web Dispatcher
  This configuration requires additional virtual host names and is more complex than option 1. However, this option is useful if the external Web Dispatcher is being added to an existing landscape.

Related Information

SAP Web Dispatcher Documentation on SAP Help Portal
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]
Option 1: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher Only [page 120]
Option 2: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher and Directly [page 123]

8.3.4.2.1 Option 1: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher Only

Use this configuration if you want tenant databases to be accessed through the external Web Dispatcher only. With this configuration, there is no direct HTTP access to the tenant databases.

The main part of this configuration involves setting the virtual host names of tenant databases configured in the external Web Dispatcher profile to point to the host of the external Web Dispatcher, instead of the host of the SAP HANA system. As a result, all requests to the virtual host name of a tenant database first go to the external Web Dispatcher and are then forwarded to the internal Web Dispatcher in the SAP HANA system.

> → Remember
>
> Before you can configure the external Web Dispatcher, the internal Web Dispatcher must already have been configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Tenant Databases*.

Single-Host Systems

The `wdisp/system_<xx>` entry for each tenant database is configured in the external Web Dispatcher profile as follows:

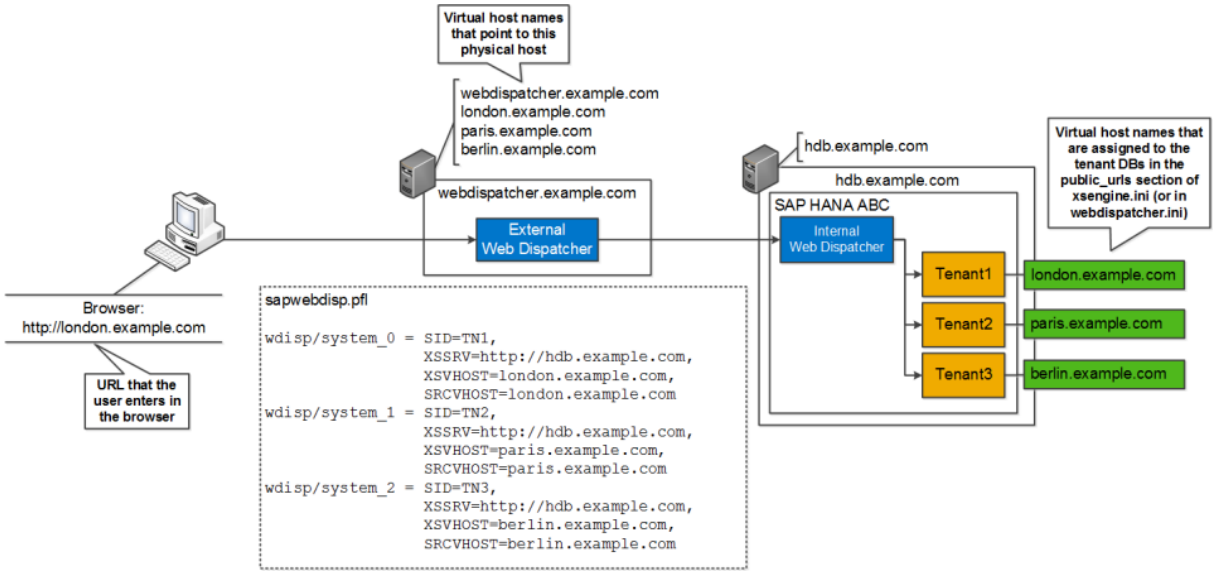
- `XSSRV` specifies the actual physical SAP HANA host name and port to which to requests are sent.
- `XSVHOST` specifies the virtual host name of the tenant database to which requests are sent.

  > ⓘ Note
  >
  > If a tenant database has multiple virtual host names assigned, only one needs to be entered in `XSVHOST`.

- `SRCVHOST` specifies the virtual host name that is used to map incoming HTTP requests to the `wdisp/system` entry that represents a particular tenant.

> ⓘ Note
>
> With this configuration option, `XSVHOST` and `SRCVHOST` are always identical.

The following figure depicts this configuration in a single-host system:



Access to Multiple Tenant Databases (Single Host)

> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the XSSRV subparameter requires port numbers, and line breaks are not allowed.

In the example depicted above, what happens when the user enters **`london.example.com`** into her browser?

1. The browser opens a TCP/IP connection to `webdispatcher.example.com` because `london.example.com` is only an alias name for `webdispatcher.example.com`.
2. The browser sends an HTTP request over this connection. The host header of this HTTP request is `london.example.com`, which is the URL that the user entered.
3. The external Web Dispatcher receives the HTTP request, checks the host header and uses this to map the request to a `wdisp/system` entry. As `london.example.com` is the SRCVHOST value for `wdisp/system_0`, the request is associated with `wdisp/system_0`.
4. The external Web Dispatcher opens a TCP/IP connection to the XSSRV value of `wdisp/system_0` (**`hdb.example.com`**).
5. The external Web Dispatcher sets the destination of the request to the tenant database specified in the XSVHOST subparameter of `wdisp/system_0` (**`london.example.com`**) by injecting a proprietary HTTP header into the request.
6. The internal SAP HANA Web Dispatcher receives the request. Because of the injected HTTP header field, it identifies that the request is destined for tenant database 1 and forwards it to the XS server of tenant database 1.

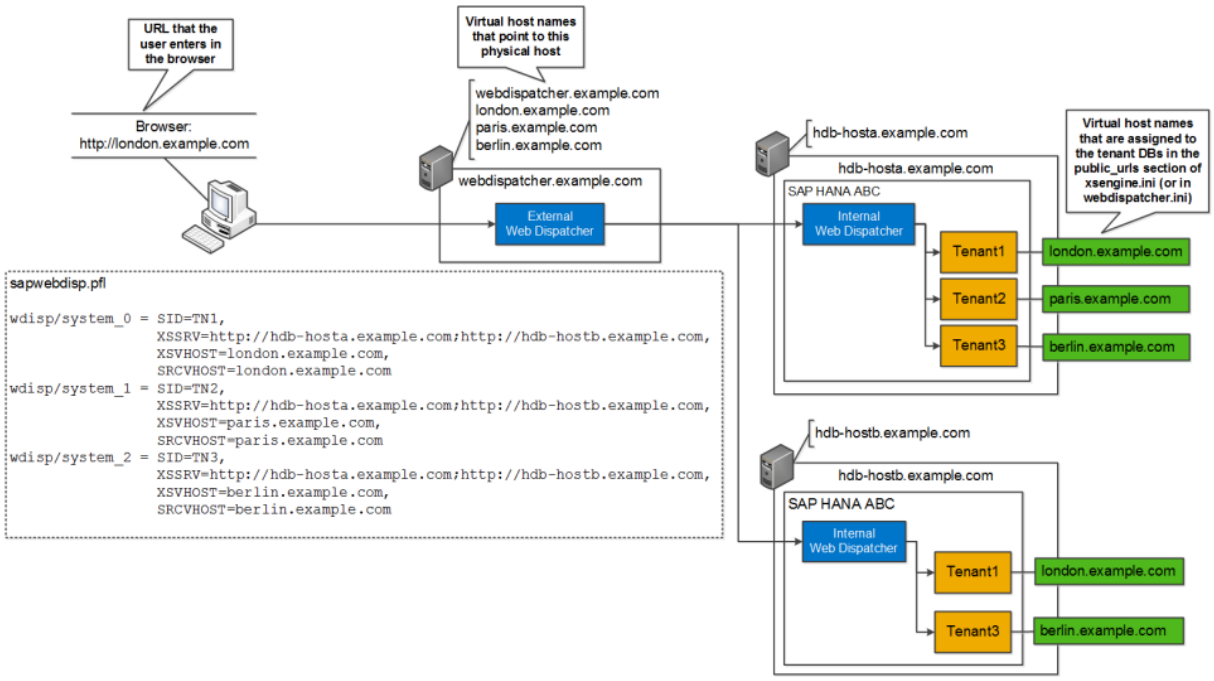
Multiple-Host Systems

In a multiple-host system, the external Web Dispatcher must be configured to connect to all hosts. This means that all hosts with a running XS server (or that may have an XS server in the future) have to be entered as the value for XSSRV as a semi-colon (;) separated list. Even if a tenant database is not running on a host, you should

add the host to the list anyway. This will enable the smooth moving of tenant databases without the need to change the external Web Dispatcher configuration.

> **→ Remember**
>
> In the internal Web Dispatcher configuration, the virtual host name in the `XSVHOST` subparameter must be to assigned to the tenant database on **all** hosts.

The following figure depicts the configuration in a multiple-host system:



> **ⓘ Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

Access to Multiple Tenant Databases (Multiple Hosts)

Now what happens when the user enters **`london.example.com`** into her browser?

The process is identical to the single-host scenario with one exception: The external Web Dispatcher periodically checks which host(s) a tenant database is actually running on. If a tenant database is running on multiple hosts, the external Web Dispatcher performs load balancing between these hosts.

Related Information

Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]

8.3.4.2.2 Option 2: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher and Directly

Use this configuration if you want tenant databases to be accessed both through the external Web Dispatcher and directly, bypassing the external Web Dispatcher.

With this configuration, additional virtual host names are required for each tenant database. These virtual host names point to the physical host name of the external Web Dispatcher. The virtual host names that are assigned to the tenant databases still point to the host of the SAP HANA system.

> → Remember
>
> Before you can configure the external Web Dispatcher, the internal Web Dispatcher must already have been configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Tenant Databases*.

Single-Host Systems

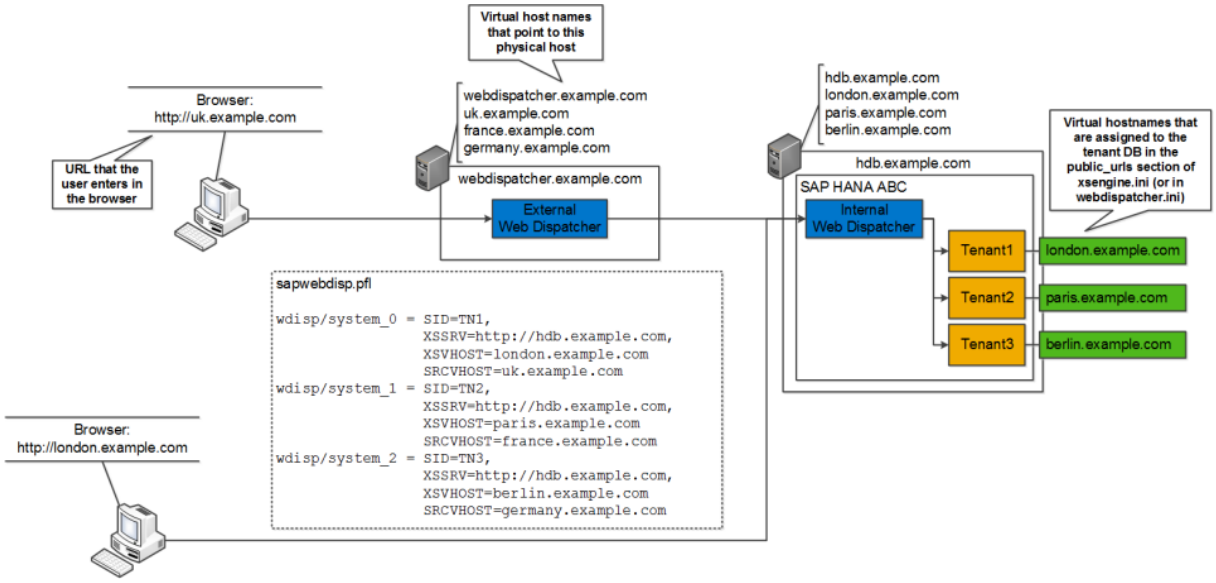
The `wdisp/system_<xx>` entry for each tenant database is then configured in the external Web Dispatcher profile as follows:

- `XSSRV` specifies the actual physical host name and port to which to requests are sent.
- `XSVHOST` specifies the virtual host name of the tenant database to which requests are sent.

  > ⓘ Note
  >
  > If a tenant database has multiple virtual host names assigned, only one needs to be entered in `XSVHOST`.

- `SRCVHOST` specifies the virtual host name that is used to map incoming HTTP requests to the `wdisp/system_<xx>` that represents a particular tenant.

The following figure depicts this configuration in a single-host system:



Access to Multiple Tenant Databases in Single-Host System

> ### Note
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the XSSRV subparameter requires port numbers, and line breaks are not allowed.

In the example depicted above, what happens when the user enters **`uk.example.com`** into his browser?

1. The browser opens a TCP/IP connection to `webdispatcher.example.com` because `uk.example.com` is only an alias name for `webdispatcher.example.com`.
2. The browser sends an HTTP request over this connection. The host header of this HTTP request is `uk.example.com`, which is the URL that the user entered.
3. The external Web Dispatcher receives the HTTP request, checks the host header and uses it to map the request to a `wdisp/system` entry. As `uk.example.com` is the SRCVHOST value for `wdisp/system_0`, the request is associated with `wdisp/system_0`.
4. The external Web Dispatcher opens a TCP/IP connection to the XSSRV value of `wdisp/system_0` (`hdb.example.com`).
5. The external Web Dispatcher sets the destination of the request to the tenant database specified in the XSVHOST parameter of `wdisp/system_0` (`london.example.com`) by injecting a proprietary HTTP header into the request.
6. The internal SAP HANA Web Dispatcher receives the request. Because of the injected HTTP header field, it identifies that the request is destined for tenant database 1 and forwards it to the XS server of tenant database 1.

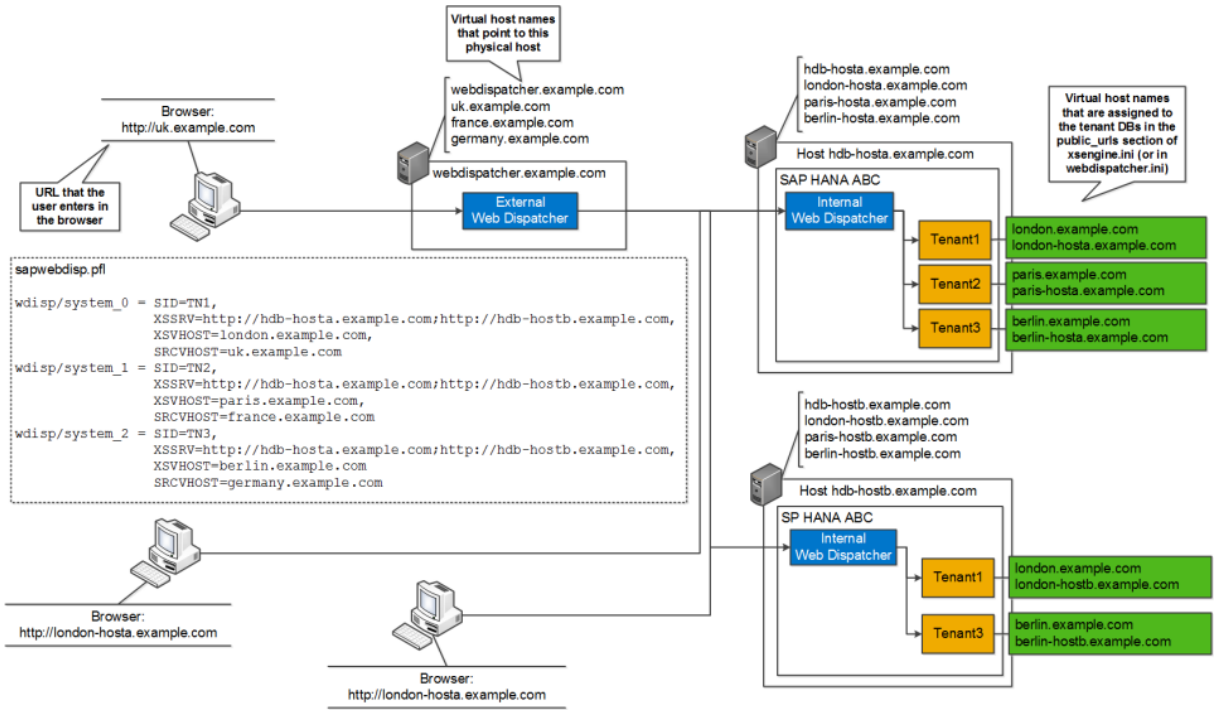
Multiple-Host Systems

In a multiple-host system, the external Web Dispatcher must be configured to connect to all hosts. This means that all hosts with a running XS server (or that may have an XS server in the future) have to be entered as the value for XSSRV as a semi-colon (;) separated list. Even if a tenant database is not running on a host, you should

add the host to the list anyway. This will enable the smooth moving of tenant databases without the need to change the external Web Dispatcher configuration.

> **→ Remember**
>
> In the internal Web Dispatcher configuration, the virtual host name in the `XSVHOST` subparameter must be to assigned to the tenant on **all** hosts.

The following figure depicts the configuration in a multiple-host system:



> **ⓘ Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

Access to Multiple Tenant Databases in Multiple-Host System

What happens after the user enters **`uk.example.com`** into his browser?

The process is identical to the single-host scenario with one exception: The external Web Dispatcher periodically checks on which host(s) a tenant database is actually running. If a tenant database is running on multiple hosts, the external Web Dispatcher performs load balancing between these hosts.

Related Information

Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]

8.4 SAP HANA System Properties

SAP HANA uses a prioritized layered configuration framework to set parameter values which control the detailed operation of the system.

The detailed fine-tuning of how your SAP HANA system operates is managed through an extensive set of configuration parameters. The system can be configured in a hierarchical manner in layers both as a whole and at the level of individual tenant databases. The details of this structure and how to manage it are described and illustrated in this section of the guide.

You can set configuration values using the SQL command line and also using the administration tools SAP HANA cockpit and SAP HANA Studio (use the links below to open the relevant sections of these documents). SAP HANA cockpit provides a number of configuration management features, for example, you can maintain configuration templates which can be copied from a database and applied to another database, take snapshots of a database configuration and make comparisons.

Related Information

Configuring SAP HANA System Properties (INI Files) [page 126]

Links to Other Documents

Database Configuration in SAP HANA Cockpit
Configuring System Properties in SAP HANA Studio

8.4.1 Configuring SAP HANA System Properties (INI Files)

An SAP HANA system has several configuration (`*.ini`) files that contain properties for configuring the system as a whole and individual tenant databases, as well as hosts and services.

About Configuration Files

Every SAP HANA service has its own configuration file (for example, `indexserver.ini`, `compileserver.ini`, and so on). For service-independent properties, there is also a `global.ini`.

Configuration files are separated into sections; sections bundle properties of the same category.

SAP HANA uses a prioritized layered configuration framework. This means that properties can be configured at different levels or layers depending on the configuration file. The following layers are available; values from higher priority layers overwrite values of lower priority layers: HOST > DATABASE > SYSTEM. See the following section on database-specific configuration parameters and the detailed example of how this is applied in practice.

Layer	Description
Default	The default value for the property (read-only).
System	The system-specific value for the property (configurable in the system database). If a system-specific value is not configured for a property, the default value applies.
Database	The database-specific value for the property (configurable in the system or tenant database). For some properties, it is possible to set database-specific values. If a database-specific value is not configured, the system-specific or host-specific value applies.
Host	The host-specific value for the property (only configurable in the system database). For some properties, it is possible to set host-specific values for multiple-host systems. If a host-specific value is not configured for a property that can be set at host level, the system-specific value applies.

Configuration files are stored on the SAP HANA server at the following locations according to layer:

- Default: `/usr/sap/<SID>/HDB<instance>/exe/config` (read only)
- System: `<sapmnt>/<SID>/SYS/global/hdb/custom/config`
- Database: `<sapmnt>/<SID>/SYS/global/hdb/custom/config/DB_<dbname>`
- Host: `/usr/sap/<SID>/HDB<instance>/<hostname>`

> **Note**
>
> By default, `<sapmnt>` is `/hana/shared`.

Changing Parameter Values

You can change configuration parameters using the SAP HANA cockpit, the SAP HANA studio, or using the ALTER SYSTEM ALTER CONFIGURATION statement (see also Console Tool setParameter.py).

> **Note**
>
> In a system replication scenario, you cannot alter parameters of the secondary site using the SAP HANA cockpit or studio. Instead you must alter the parameters directly in the *.ini files on the secondary site. Afterwards reconfigure the database using `hdbnsutil -reconfig`.

All public configuration parameters are defined in table CONFIGURATION_PARAMETER_PROPERTIES. In addition to basic properties such as data type, unit and a default value, the definition includes a flag to indicate if a system restart is required before a changed value becomes effective. You can look up parameter details using the system views listed below, using SAP HANA cockpit or by referring to the *Configuration Parameter Reference* in the SAP Help Portal.

Validation

When you attempt to change a parameter value, the change is validated against the definition. Validation can be applied as follows:

- Value restrictions where a list of permitted values has been defined, either as a range (for example 0-100) or as a list of values ({default,on,off})

- Layer restrictions where a change is only permitted at a specified layer (either SYSTEM or HOST)
- Custom restrictions are additional restrictions where more detailed parsing of the value entered is required; this may be used, for example, to validate a value where a file path must be entered.

If the changed value is not valid for some reason, the new value is stored but a warning message is returned. You can prevent invalid values from being saved by setting the `unsupported_configuration_change` parameter to 'error', in which case the change is then rejected.

Because there is no hard validation of configuration values two alert checkers run in the background to monitor changes made to these settings: Alert 136 checks if any parameters are set to an unsupported value and Alert 137 triggers an information message if a configuration change has been made which requires a service restart in order to make it effective. The source of this information is system view M_CONFIGURATION_PARAMETER_VALUES which shows whether a value violates any restrictions (see System Views below).

In general, we do not recommend that you change the default values of parameters unless stated in the documentation or instructed to do so by SAP Support.

Parameter Tracking

For traceability purposes changes to configuration values can be logged; optionally, a reason for the change can be entered in a comment value.

The tracking feature is enabled by default, but can be disabled by setting configuration parameter `write_log` in the `indexserver.ini` file to **`false`**. When it is enabled the SQL ALTER CONFIGURATION statement automatically updates the view SYS.M_INIFILE_CONTENT_HISTORY, this includes details of the time the change was made, the user name, the current and previous values and any comment text that was entered.

The following SQL example shows how the COMMENT keyword is used to include a reason for the change (refer to the *SAP HANA SQL Reference Guide* for more details):

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'DATABASE', 'C11')
SET ('memorymanager', 'allocationlimit') = '500000'
WITH RECONFIGURE
COMMENT 'Reverting to previous setting';
```

An SQL command CLEAR is also available to truncate the ini file history:

```
ALTER SYSTEM CLEAR INIFILE CONTENT HISTORY
```

This automatically adds a timestamp and comment to the history to indicate when the history was deleted.

Any changes made to configuration values at the level of the file system – by scripts or tools such as `hdbnsutil` are also tracked. This is done at system startup when the system recognizes that offline configuration changes have been made and writes an entry into the log history; this entry does not include details of the values which have been changed.

System Views

The system view M_INIFILES contains information about the layers on which the properties of each configuration file can be configured and the view M_INIFILE_CONTENTS contains information about the values configured for the properties of each file and on which layers it has been defined.

The view M_CONFIGURATION_PARAMETER_VALUES shows more details about the value which has been applied. This includes the RAW_VALUE as it is cached in the service, if any parameter restrictions have been violated and also, if the value has been recently changed, whether a restart is still required to make the new value effective. This view can therefore be used to see the currently effective values.

This view may contain multiple entries for each key because a record is maintained for each key for each individual service (HOST / PORT). To query the table and filter the results for your own host and port you can refer to the OWN value maintained in M_CONNECTIONS. The following query shows this and selects, as an example, all keys in the 'expensive_statement' section for the service executing your statement:

```
SELECT v.* FROM SYS.M_CONFIGURATION_PARAMETER_VALUES v, SYS.M_CONNECTIONS c
WHERE v.SECTION='olap' AND v.HOST=c.HOST AND v.PORT=c.PORT AND c.OWN='TRUE';
```

The view M_INIFILE_CONTENT_HISTORY is used by parameter tracking to show the history of changes as described above.

Further Information

An SAP Community blog *Configuration Framework Improvements in SAP HANA 2* gives further details of this feature.

Related Information

Database-Specific Configuration Parameters

Console Tool setParameter.py [page 133]

Links to other documents:

ALTER SYSTEM ALTER CONFIGURATION Statement (System Management)
M_INIFILES System View
M_INIFILE_CONTENTS System View
M_INIFILE_CONTENT_HISTORY System View
M_CONFIGURATION_PARAMETER_VALUES System View
CONFIGURATION_PARAMETER_PROPERTIES System View
Database Configuration in SAP HANA Cockpit
Configuring System Properties in SAP HANA Studio
SAP HANA Configuration Parameter Reference
SAP Community Blog: Configuration Framework Improvements in SAP HANA 2

8.4.2 Database-Specific Configuration Parameters

In addition to the layers "default", "system", and "host", system configuration files also have a "database" layer to facilitate the configuration of properties for individual databases.

In general, you can configure database-specific properties both in the system database and in tenant databases themselves. Properties configured in the system database can be applied to all databases (if configured in the system layer) or to specific databases (if configured in database layer).

Properties configured in a tenant database apply to that tenant database only. Only properties in the following files can be configured in tenant databases:

- `attributes.ini`
- `docstore.ini`
- `dpserver.ini`
- `esserver.ini`
- `executor.ini`
- `extensions.ini`
- `global.ini`
- `indexserver.ini`
- `scriptserver.ini`
- `xsengine.ini`

The file `multidb.ini` is used as a blocklist to protect critical system settings from being changed in tenant databases. The list is delivered with a default configuration but this can be modified. The topic 'Lock Parameters Against Editing for a Tenant Database' describes how to do this from within SAP HANA cockpit.

File Locations

The properties in the `nameserver.ini` file are not database-specific, they can only be configured at system level. The `nameserver.ini` file is stored at:

```
/hana/shared/$SID/global/hdb/custom/config
```

The properties in the `indexserver.ini` file can be database-specific. Properties that are configured in the system layer and apply to all databases are also stored in the `indexserver.ini` at this location `/hana/shared/$SID/global/hdb/custom/config`, but properties configured in the database layer for an individual database (and which override the system-layer value) are stored in a database-specific version of the `indexserver.ini` at:

```
/hana/shared/$SID/global/hdb/custom/config/DB_<dbname>
```

Layered Configuration for Tenant Databases

Many properties can be configured in the system, host, and database layer. Values configured in the database layer take precedence over system-layer values.

Note that when you are connected to a tenant database, you see that the database-layer value of a property is also displayed as the system-layer value. This occurs because, from the perspective of the tenant database, the database and the system are effectively the same. The true system-layer value (that is, the value configured for all databases in the system database) is displayed in the tenant database as the default-layer value. This is illustrated in the detailed example below.

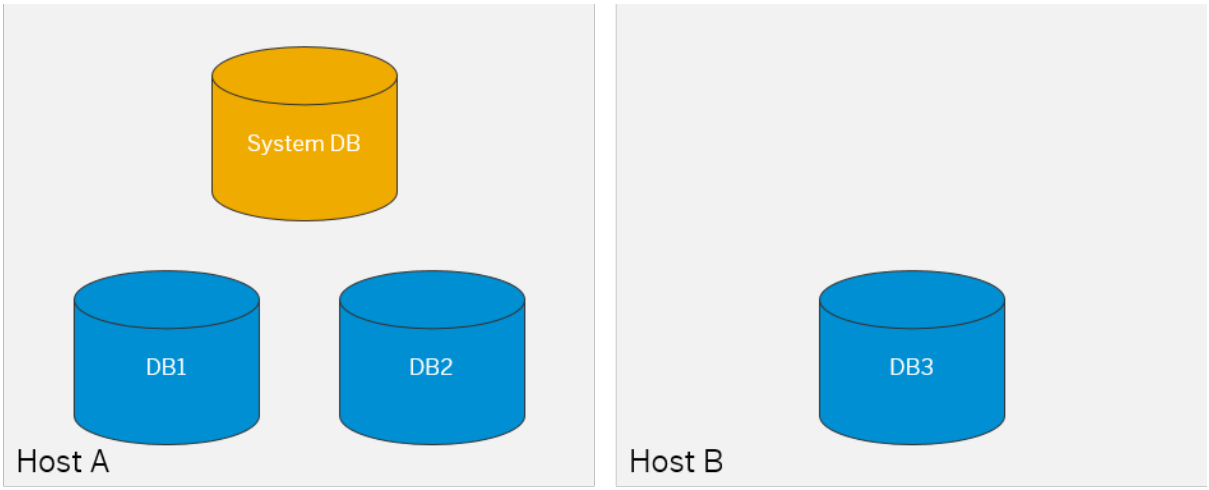
Values configured in the host layer take precedence over database-layer values. Host values can only be configured in the system database.

You can see actual configuration values in SAP HANA cockpit, on the resources *Database Overview* page (select the *Manage System Configuration* link on the *Database Administration* card), or query the following system views:

- M_INIFILE_CONTENTS (SYS_DATABASES)
  This view can be accessed only from the system database. It contains the values configured for all properties on system, host, and database layer for **all** active databases.
- M_INIFILE_CONTENTS (SYS)
  This view is available in every database and contains the values that apply to the database in question. Values that were configured in the system layer in the system database are identified as default-layer values. Values that were configured in the database layer in the tenant database are identified as system- and database-layer values. Values configured at the host layer are shown only for hosts on which the database is running.

Example

A system has 3 tenant databases DB1, DB2, and DB3, distributed across 2 hosts Host A and Host B:



The default value of the property `[execution] max_concurrency` in the `global.ini` file is `0`. The system administrator changes the default configuration of this property in the system database `indexserver.ini file` as follows:

First, the system administrator creates a new system-layer value (**`10`**) in `indexserver.ini`. Since the system-layer value applies to all tenant databases and can't be changed by a tenant database user, users on all tenant databases initially see the value `10` as the default configuration:

Executed in DB1, DB2, and DB3:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10 <-- Effective value
```

Next, the system administrator sets a new value (**`20`**) for DB1, while leaving the configuration for DB2 and DB3 unchanged.

Executed in DB1:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10    |
| indexserver.ini  | SYSTEM       |               |            | execution  | max_concurrency | 20    |
| indexserver.ini  | DATABASE     |               |            | execution  | max_concurrency | 20 <-- Effective value
```

Executed in DB2 and DB3:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10 <-- Effective value
```

> **Note**
>
> In DB1, the database-layer value is duplicated to the system layer because from the perspective of the tenant database, the database and the system are effectively the same. Unlike the M_INIFILE_CONTENTS view, the M_CONFIGURATION_PARAMETER_VALUES view is not from the perspective of the database, but shows the global layer perspective instead.

Finally, the system administrator sets a new value (**`15`**) for host A. Since host values take precedence over database values, this value changes the effective value for DB1 and DB2.

Executed in DB1:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10    |
| indexserver.ini  | HOST         |               | HOST A     | execution  | max_concurrency | 15 <--- Effective value
| indexserver.ini  | SYSTEM       |               |            | execution  | max_concurrency | 20    |
| indexserver.ini  | DATABASE     |               |            | execution  | max_concurrency | 20    |
```

Executed in DB2:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

 FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10    |
| indexserver.ini  | HOST         |               | HOST A     | execution  | max_concurrency | 15 <--- Effective value
| indexserver.ini  | SYSTEM       |               |            | execution  | max_concurrency | 20    |
```

Executed in DB3:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10 <--- Effective value
```

Related Information

M_INIFILES System View
M_INIFILE_CONTENTS System View
M_CONFIGURATION_PARAMETER_VALUES System View
ALTER SYSTEM ALTER CONFIGURATION Statement (System Management)
SAP HANA Configuration Parameter Reference
Lock Parameters Against Editing for a Tenant Database

8.4.3 Console Tool setParameter.py

You can use the console tool `setParameter.py` to make changes to configuration parameter values, it can be run from the command line or integrated with other administration management tools and scripts.

Using The Tool

The tool can be used either when the database is offline or online if for any reason it is not possible to make a configuration parameter change by executing an SQL statement. The tool can be run from the command line or integrated with other cluster management tools which may not have online access to the system. The tool ensures that if changes are made simultaneously to an ini file the changes are synchronized and applied correctly, this eliminates the need for administrators to directly edit ini files.

Using the tool when the system is online you can also enter a comment to provide transparency about why a change was made; comments are saved in table M_INIFILE_CONTENT_HISTORY. Also when the system is online you can make a parameter change and include a request to reconfigure the services at the same time.

When you identify the parameter value to change you also specify at which layer the change should be made (see example which follows):

- SYSTEM
- DATABASE:<DATABASENAME>
- HOST:<HOSTNAME>

Use the host layer to make a change to a parameter on a remote host. In this case the system must be online and the host must be reachable. If no host name is provided localhost is assumed.

Syntax

The command supports both setting and unsetting parameters. No key value is required when unsetting a parameter:

```
python setParameter.py -set(or unset)=<LAYER>/<INIFILE_NAME>/<SECTION>/
<KEY>=<VALUE>
[-comment=comment] [-reconfigure] [-sapcontrol=1]
```

The tool also supports a 'help' argument (-h) which displays a page of help text. The help page is shown automatically if invalid syntax is entered.

Example

You can use the tool as a script to set or unset multiple values at the same time. The following example sets the password length requirement and unsets the password layout parameter. This example uses the `reconfigure` argument (assuming the system is running) so the command will also attempt to reconfigure the service:

```
python setParameter.py -set="DATABASE:MyDatabase/indexserver.ini/password policy/
minimal_password_length=10"
-comment="Required Policy" -unset="DATABASE:MyDatabase/indexserver.ini/password
policy/password_layout"
-comment="layout not required" -reconfigure -sapcontrol=1
```

Return Values

After submitting the command one of the following codes is returned indicating success or failure of the execution:

- 0 Successful execution
- 1 Could not be executed because of incorrect usage (unexpected parameter values)
- 2 Other error

If the `-sapcontrol=1` switch is set then one of the following is printed in the output:

- SAPCONTROL-OK
- SAPCONTROL-ERROR:<errorMsg>

8.4.4 Configure System Usage Type

You can configure the usage type of an SAP HANA system (for example, production, development) during installation with the `system_usage` parameter, or later by changing the system properties. Clients such as the SAP HANA cockpit and SAP HANA studio can use this property to alter behavior.

Prerequisites

You have the system privilege INIFILE ADMIN.

Context

The system usage type also governs to which resource group (production, test, development) the system is automatically assigned in SAP HANA cockpit when you register the system as a cockpit resource. During installation the `system_usage` parameter is available to set this value - refer to the *SAP HANA Server Installation and Update Guide* for details.

Procedure

In the `global.ini` configuration file, change the value of the `usage` property in the `system_information` section.

You can enter any value, but the values listed below can be used by clients to alter behavior. This may trigger a warning to users when they are about to perform critical operations on systems with usage type **`production`** (for example, execute SQL statements, stop or restart the system, perform a data backup, and so on).

- production
- test
- development
- custom (default)

> **Note**
>
> If the system usage type is anything other than one of these values, you will not be able to register the SAP HANA system as a resource in the SAP HANA cockpit.

Results

A system restart is not required.

Related Information

system_usage
Database Groups

8.4.5 Reserve Connections for Administrators

If the maximum number of connections has been reached in an SAP HANA system, it is not possible for anyone to log on, not even an administrator. For this reason, you can reserve a certain number of connections for administrative access only.

Prerequisites

You have the system privilege INIFILE ADMIN and SESSION ADMIN.

Procedure

1. In the `global.ini` configuration file, change the value of the `reserved_connections` property in the `session` section.

   As the value, specify the number of connections you want to reserve. The default number of reserved connections is 10. The minimum number is 1.
2. Restart the system.

Results

When the maximum number of connections minus the number reserved connections is reached, only an administrator with the system privilege SESSION ADMIN can log on to the system, for example, to resolve blocking situations by canceling sessions.

8.5 Managing SAP HANA Licenses

A valid license key is required to use the SAP HANA database. Additional license keys are required for certain applications running on SAP HANA, as well as for certain SAP HANA options and capabilities.

For most on premise SAP software, license keys are required to activate the product. You can request a license from the License Key Application of the SAP ONE Support Launchpad (specific credentials are required):

https://support.sap.com/en/my-support/keys.html

You will receive the license by email as a text file (the license key value is a single line in the file) which you can then upload and apply; this will license the usage of your system up to an allocated memory allowance.

You can apply a license key in SAP HANA cockpit (Upload System DB License). In cockpit you can also see which licenses are available in your database, you can install or remove license keys, and also view memory usage with respect to licensing.

You can also apply a license key from the SQL command line using the statement SET SYSTEM LICENSE where the license text is appended to the command.

Further details of licenses and license types are provided in this section.

Related Information

License Keys for SAP HANA Database [page 137]
License Management (SAP HANA Administration with SAP HANA Cockpit)
SET SYSTEM LICENSE Statement (System Management)
Important Disclaimer for Features in SAP HANA [page 1677]

8.5.1 License Keys for SAP HANA Database

The SAP HANA database supports two kinds of license keys: temporary license keys and permanent license keys

SAP HANA licenses can be installed for the system database (global) or for a single tenant database (local). Global licenses are for the system database and all the tenant databases, but a license installed in a tenant database governs only that tenant database. If you remove a tenant-specific license key, that tenant database reverts to the global license key installed in the system database.

Temporary License Keys

A temporary license key is automatically installed with a new SAP HANA system. A temporary license key is valid for 90 days. During this period, request and install a permanent license key.

Permanent License Keys

Permanent License Keys: Expiration

Before a permanent license key expires, request and apply a new permanent license key. If a permanent license key expires, a (second) temporary license key is automatically installed. This temporary license key is valid for 28 days. During this time, you can request and install a new permanent license key.

You can request a permanent license key on SAP Support Portal (http://support.sap.com) under *Request Keys*. Permanent license keys are valid until the predefined expiration date. Furthermore, they specify the amount of memory licensed to an SAP HANA installation.

Permanent License Keys: Types

There are two types of permanent license key for SAP HANA: unenforced and enforced.

If an unenforced license key is installed, the operation of SAP HANA isn't affected if its memory consumption exceeds the licensed amount of memory. However, if an enforced license is installed, the system is locked down when the current memory consumption of SAP HANA exceeds the licensed amount of memory plus some tolerance. If lockdown occurs, either SAP HANA needs to be restarted, or a new license key that covers the amount of memory in use needs to be installed.

The two types of permanent license key differ from each other in the following line in the license key file:

License Key Type	License Key File Entry
Unenforced	SWPRODUCTNAME=SAP-HANA
Enforced	SWPRODUCTNAME=SAP-HANA-ENF
	SWPRODUCTNAME=SAP-HANA-DEV
	SWPRODUCTNAME=SAP-HANA-DIGITAL

> ⓘ Note
>
> It's technically possible to install an enforced license in an SAP HANA instance with a regular, unenforced permanent license. In this case, the unenforced license key has priority. That is, if a valid unenforced license key is found, excessive memory consumption doesn't result in a system lockdown. However, if one license key expires and becomes invalid, the other license, if valid, becomes the valid license key of the instance. If the latter is an enforced license key, then the memory consumption is checked.

License Keys for Tenant Databases

You can install permanent license keys in individual tenant databases. The license key installed in a tenant database is valid for that database only and takes precedence over the license key installed in the system database. If a tenant-specific license key isn't installed, the system database license key is effective in the tenant database.

> → Tip
>
> The system view SYS.M_LICENSE provides tenant administrators with information on the license key effective in their tenant database, as well as where the license key is installed: in the tenant database itself or in the system database. System administrators can use the view SYS_DATABASES.M_LICENSE to see the same information for all tenant databases.

System Lockdown

The system goes into lockdown mode in the following situations:

- The permanent license key has expired and either:
  - You didn't renew the temporary license key within 28 days, or
  - You did renew the temporary license key but the hardware key has changed
- The installed license key is an enforced license key and the current memory consumption exceeds the licensed amount plus the tolerance.
- You deleted all license keys installed in your database.

In lockdown mode, it isn't possible to query the database. Only a user with the system privilege LICENSE ADMIN can connect to the database and execute license-related queries, such as, obtain previous license data, install a new license key, and delete installed license keys.

In addition, the database can't be backed up in lockdown mode.

Additional SAP HANA Licenses

Additional licenses are required for certain applications running on the SAP HANA database, as well as certain SAP HANA options and capabilities. For more information, see SAP Note 1644792 (License key/installation of SAP HANA).

Related Information

SAP Support Portal
SAP Note 1644792

8.6 Monitoring the SAP HANA Database

It's important that you monitor the operation and key performance indicators of your SAP HANA databases. Monitoring tools and resources are available in both SAP HANA cockpit and SAP HANA studio.

It's important that you monitor the operation of SAP HANA databases on a regular basis. Although SAP HANA actively alerts you of critical situations, keeping an eye on resource usage and performance will help you identify patterns, forecast requirements, and recognize when something is wrong. You can monitor SAP HANA using both the SAP HANA cockpit and the SAP HANA studio. These tools rely on the monitoring and alerting information provided by system views and the statistics service. An overview of system alerts with recommended actions to take if the alert is triggered is available in the *SAP HANA Troubleshooting and Performance Analysis Guide*.

Related Information

Monitoring in SAP HANA Studio [page 140]
Monitoring in SAP HANA Cockpit [page 140]
System and Statistics Views [page 141]
The Statistics Service [page 143]
Alerts and the Statistics Service (Troubleshooting Guide)

8.6.1 Monitoring in SAP HANA Cockpit

You can perform several database monitoring tasks for the system and tenant databases in the SAP HANA cockpit using a range of dedicated apps.

With SAP HANA cockpit you can perform such tasks as:

- Monitor overall database health
- Monitor status and resource usage of individual database services
- Analyze database performance across a range of key performance indicators related to memory, disk, and CPU usage
- Analyze the comparative memory utilization of column tables
- Analyze the memory statistics of the components of database services
- Monitor alerts occurring in the database and analyze patterns of occurrence
- Configure the alerting mechanism by changing alert threshold values, switching alert checkers on/off, and checking for alerts out of schedule
- Monitor the status of system replication (if enabled).

Refer to the Monitoring View section in the following guide for full details of these operations: *SAP HANA Administration with SAP HANA Cockpit*.

Related Information

Monitoring View (SAP HANA Cockpit Guide)
Managing Tenant Databases [page 65]

8.6.2 Monitoring in SAP HANA Studio

The SAP HANA studio provides several tools for database monitoring.

- *System Monitor* ( )
  This editor provides you with an overview of all your SAP HANA systems at a glance, including system availability and current resource usage information. From the *System Monitor* you can drill down into each individual system in the *Administration* editor.

- *Administration* ( )
  This editor provides detailed information about resource usage, current alerts, system performance, system configuration, as well as tools for analyzing and troubleshooting issues in your system.

Refer to the *SAP HANA Studio* guide for details of the monitoring resources which cover such topics as:

- Monitoring System Availability
- Monitoring Overall System Status and Resource Usage
- Monitoring Status and Resource Usage of System Components
- Monitoring Host Status and Auto-Failover Configuration
- Monitoring Alerts
- Monitoring Disk Space
- Using User-Defined SQL Statements for System Monitoring
- Basic Monitoring Checklist for SAP HANA Systems

Related Information

System Monitor (SAP HANA Studio)
Administration Editor (SAP HANA Studio)

8.6.3 System and Statistics Views

The SYS schema of the SAP HANA database contains various information about the current state of the database in its many views. Historical data is collected and stored in the views of the _SYS_STATISTICS schema.

System Views

The SAP HANA database provides many system views that contain important information about the database. Much of the information in these views is available in SAP HANA's administration tools. However, it can be necessary to examine the data directly as part of more detailed monitoring and performance analysis.

System views are located in the SYS schema. However, as public synonyms of all views exist, it is not necessary to specify the schema name when you query these views.

> **Note**
>
> Many system views are available in two versions – one that shows the data gathered since a particular service was last started, and one that shows the data gathered since the time the view was last reset. For example, the view M_VOLUME_IO_TOTAL_STATISTICS shows the total read size and the total write size for each volume since a service was last started. The SQL command `ALTER SYSTEM RESET MONITORING VIEW SYS.M_VOLUME_IO_TOTAL_STATISTICS_RESET` initializes the statistics shown by this view. The view M_VOLUME_IO_STATISTICS_RESET now shows the statistics since the reset time.

You can access the information in a system view by querying the view directly using SQL, or opening it from the catalog using the SAP HANA database explorer or the SAP HANA studio.

> → Tip
>
> Several predefined SQL SELECT statements on system views are available in the SAP HANA studio on the *System Information* tab of the Administration editor or in the statement library of the SAP HANA database explorer . These statements provide you with easy access to important system information. Double-clicking an entry in this list executes the underlying statement.
>
> If you have compiled your own SQL statements for monitoring purposes, you can save these statements for convenient repeated execution.

For more information about all available system views, see the *SAP HANA SQL and System Views Reference*.

Statistics Views

The internal monitoring infrastructure of the SAP HANA database (statistics service) is continuously collecting and evaluating information about status, performance, and resource usage from all components of the SAP HANA database. This information is historicized to tables and views in the schema _SYS_STATISTICS. You can use these tables and views to analyze system behavior over time.

Additional System Views in the System Database

Every database has its own SYS and _SYS_STATISTICS schemas that contain information about that database only. For system-level monitoring, additional views are accessible in the system database: the M_DATABASES (SYS) view and the views in the SYS_DATABASES schema.

Related Information

System Views Reference
Embedded Statistics Service Views (_SYS_STATISTICS schema)
Use the Statement Library to Administer Your Database (SAP HANA Database Explorer)
Use User-Defined SQL Statements for System Monitoring
The Statistics Service [page 143]

8.6.4 The Statistics Service

The statistics service is a central element of SAP HANA's internal monitoring infrastructure. It notifies you when critical situations arise in your systems and provides you with historical monitoring data for analysis.

- Introduction [page 143]
- Technical Implementation [page 143]
- Data Management in the Statistics Service [page 144]
- The Statistics Service in Multitenant Database Containers [page 144]

Introduction

As an SAP HANA database administrator, you need to monitor the status of the system and its services and the consumption of system resources. When critical situations arise, you need to be notified so that you can take appropriate action in a timely manner. For data center operation and resource allocation planning, you need to analyze historical monitoring data. These requirements are met by SAP HANA's internal monitoring infrastructure. A central element of this infrastructure is the statistics service.

The statistics service collects and evaluates information about status, performance, and resource consumption from all components belonging to the system. In addition, it performs regular checks and when configurable threshold values are exceeded, issues alerts. For example, if 90% of available disk space is used, a low priority alert is issued; if 98% is used, a high priority alert is issued.

Monitoring and alert information are stored in database tables in a dedicated schema (_SYS_STATISTICS). From there, the information can be accessed by administration tools, such as SAP HANA cockpit, or SAP HANA studio.

Technical Implementation

The monitoring and alerting features of the SAP HANA database are performed by the statistics service.

The **statistics service** is implemented by a set of tables and SQLScript procedures in the master index server and by the statistics scheduler thread that runs in the master name server. The SQLScript procedures either collect data (data collectors) or evaluate alert conditions (alert checkers). Procedures are invoked by the scheduler thread at regular intervals, which are specified in the configuration of the data collector or alert checker. Data collector procedures read system views and tables, process the data (for example, if the persisted values need to be calculated from the read values) and store the processed data in measurement tables for creating the measurement history.

Alert checker procedures are scheduled independently of the data collector procedures. They read current data from the original system tables and views, not from the measurement history tables. After reading the data, the alert checker procedures evaluate the configured alert conditions. If an alert condition is fulfilled, a corresponding alert is written to the alert tables. From there, it can be accessed by monitoring tools that display the alert. It is also possible to have e-mail notifications sent to administrators if an alert condition is fulfilled. Depending on the severity level of the alert, summary emails are sent with different frequency (hourly, every 6 hours, daily). You can also trigger alert checker procedures directly from monitoring tools (for example, SAP HANA cockpit).

Data Management in the Statistics Service

The following mechanisms exist to manage the volume of data collected and generated by the statistics service:

- Configurable data retention period
  The data collected by the data collectors of the statistics service is deleted after a default number of days. The majority of collectors have a default retention period of 42 days. For a list of those collectors that have a different default retention period, execute the following statement:

  ```
  SELECT o.name, s.retention_days_default FROM
  _SYS_STATISTICS.STATISTICS_SCHEDULE s,  _SYS_STATISTICS.STATISTICS_OBJECTS o
  WHERE s.id = o.id AND o.type =  'Collector'and s.retention_days_default !=
  42 order by 1;
  ```

  You can change the retention period of individual data collectors with the following SQL statement:

  ```
  UPDATE _SYS_STATISTICS.STATISTICS_SCHEDULE set
  RETENTION_DAYS_CURRENT=<retention_period_in_days> where
  ID=<ID_of_data_collector>;
  ```

  → Tip

  To determine the IDs of data collectors execute the statement:

  ```
  SELECT * from _SYS_STATISTICS.STATISTICS_OBJECTS where type = 'Collector';
  ```

  Alert data in the _SYS_STATISTICS.STATISTICS_ALERTS table is also deleted by default after a period of 42 days. You can change this retention period with the statement:

  ```
  UPDATE _SYS_STATISTICS.STATISTICS_SCHEDULE set
  RETENTION_DAYS_CURRENT=<retention_period_in_days> where ID=6002;
  ```

- Maximum number of alerts
  By default, the number of alerts in the system (that is rows in the table _SYS_STATISTICS.STATISTICS_ALERTS_BASE) cannot exceed 1,000,000. If this number is exceeded, the system starts deleting rows in increments of 10 percent until the number of alerts is below the maximum. To change the maximum number of alerts permitted, add a row with the key `internal.alerts.maxrows` and the new maximum value to the table _SYS_STATISTICS"."STATISTICS_PROPERTIES.

  ⁘ Example

  ```
  INSERT INTO _SYS_STATISTICS.STATISTICS_PROPERTIES VALUES
  ('internal.alerts.maxrows', 500000);
  ```

The Statistics Service in Multitenant Database Containers

In multiple-container systems, the statistics service runs as an embedded process in the (master) index server of every tenant database. Every database has its own _SYS_STATISTICS schema.

Monitoring tools such as the SAP HANA cockpit allow administrators in the system database to access certain alerts occurring in individual tenant databases. However, this access is restricted to alerts that identify

situations with a potentially system-wide impact, for example, the physical memory on a host is running out. Alerts that expose data in the tenant database (for example, table names) are **not** visible to the system administrator in the system database.

Related Information

Tenant Databases [page 17]
Using Alert and Collector Profiles [page 145]

8.6.4.1 Using Alert and Collector Profiles

You can apply profiles to alert and collector values.

The internal monitoring infrastructure of the SAP HANA database is continuously collecting and evaluating information about status, performance, and resource usage from all components of the SAP HANA database. You can control the alert and collector behavior by applying profiles.

Available Profiles

Each profile affects the system performance in a different way. Select the profile that suits your system best. All profiles change the value of the parameter `retention_days_default`, which controls for how many days collector data is stored. Additionally, the profiles `HXE` and `SYSTEMDB` reduce the number of enabled alerts and collectors.

The following profiles are available:

Profile	Description
S	This profile applies default values multiplied by 0.5. Resulting values are rounded up, if necessary.
M	No changes are applied to the default alert and collector values. This is the default profile.
L	This profile applies default values multiplied by 1.5. Resulting values are rounded up, if necessary.
HXE	This profile disables several alerts and collectors, and sets the default values for all collectors to 1. This is the default profile for SAP HANA Express systems.
SYSTEMDB	This profile disables several alerts and collectors, and applies default values multiplied by 0.5. Resulting values are rounded up, if necessary. This is the default profile for the system database.

Switching Between Profiles

The following mechanisms exist to manage alert and collector profiles:

- View the currently enabled profile with the following statement:

```
SELECT VALUE FROM _SYS_STATISTICS.STATISTICS_PROPERTIES WHERE KEY =
'internal.sizing.profile'
```

- Enable a profile with the following statement:

```
UPDATE _SYS_STATISTICS.STATISTICS_PROPERTIES SET VALUE = '<S, M, L, HXE,
SYSTEMDB>' WHERE KEY = 'internal.sizing.profile'
```

Customized values for `DEFAULT_VALUE` in the table `_SYS_STATISTICS.STATISTICS_ALERT_THRESHOLDS` and `STATUS`, `INTERVALLENGTH`, `RETENTION_DAYS_DEFAULT` in the table `_SYS_STATISTICS.STATISTICS_SCHEDULE` are stored with each profile.

SAP HANA Express Profile (HXE)

The SAP HANA Express profile disables a number of alerts and sets the default values for all collectors to 1.

The following alerts remain active after enabling the HXE profile:

- Alert_Check_Inactive_Services
- Alert_Check_Restarted_Services
- Alert_Check_Service_Allocation_Limit
- Alert_CrashDump_Files
- Alert_Internal_Disk_Full_Events
- Alert_Internal_Events
- Alert_License_Expiring
- Alert_Lock_Wait_Time_Out
- Alert_Long_Running_Statements
- Alert_Mon_TraceFileSize
- Alert_Password_Expiration
- Alert_RTE_Dump_Files
- Alert_Summary_Email_All
- Alert_Summary_Email_High
- Alert_Summary_Email_Medium_High

The following collectors remain active after enabling the HXE profile:

- Collector_Global_Internal_Events
- Collector_Host_Column_Tables_Part_Size
- Collector_Host_Heap_Allocators
- Collector_Host_Service_Memory
- Collector_Tel_Disk_Usage

- Collector_Tel_Feature_Usage
- Collector_Tel_Host_Information
- Collector_Tel_Inifile_Contents
- Collector_Tel_Licenses
- Collector_Tel_Out_Of_Memory_Events
- Collector_Tel_System_Overview

Related Information

Monitoring Alerts
Monitoring and Analyzing Performance in SAP HANA Studio
SAP Note 2867960

8.6.4.2 Failing Checks

The alerting mechanism of the SAP HANA database relies on the regular execution of checks. If a check fails to execute, it is important to investigate the reason why. Otherwise, you may not be warned about potentially critical situations. Checks often fail due to a shortage of system resources.

If a check fails to execute, an alert is issued indicating that there is an internal statistics service problem. You can also see whether individual checks have stopped running on schedule on the *Alerts* page of the SAP HANA cockpit. As long as a check is not being executed, it cannot alert you about potentially critical situations.

Alerting Mechanism	Response
Statistics service	A check is disabled the first time it fails to execute. It remains disabled for a specific length of time before it is automatically re-enabled. This length of time is calculated based on the values in the following columns of the table STATISTICS_SCHEDULE (_SYS_STATISTICS): • INTERVALLENGTH • SKIP_INTERVAL_ON_DISABLE Once INTERVALLENGTH x SKIP_INTERVAL_ON_DISABLE has elapsed, the check is re-enabled. The default values for all checks are such that failed checks remain disabled for 1 hour. The system determines the status of every check and/or whether the time to re-enablement has elapsed every 60 seconds. You can control the time to re-enablement by changing the value in the column SKIP_INTERVAL_ON_DISABLE. You can also re-enable the check manually.

> **Note**
>
> The behavior described above also applies to the data collectors of the statistics service.

Related Information

Switch Alerts Off/On
SAP Note 1991615

8.7 Managing and Monitoring SAP HANA Performance

Optimizing performance is a key administration activity. This topic gives an overview of the tools and resources available to help improve the performance of your database.

Monitoring past and current information about the performance of the SAP HANA database is important to prevent performance issues and for root-cause analysis of problems. Some of the key tools and resources available to monitor, analyze and optimize performance are introduced here.

Monitoring

Both the SAP HANA cockpit and SAP HANA Studio provide monitoring tools; the Performance Monitor of the SAP HANA cockpit is particularly useful for analysis as it shows side-by-side visual displays of both system performance and the workload currently being applied.

Caching

Caching is used widely in SAP HANA as a strategy to improve performance by re-using queried data rather than re-reading and processing the data every time it is requested. Query result caches are used for caching data and the Plan Cache is available for query optimization by storing execution plans of frequently executed SQL statements.

Hints

Hints are instructions for the SAP HANA database server which influence the way a database request is processed. They are typically used to optimize SAP HANA performance or memory consumption and have no effect on the result set of the request. Predefined hints for various purposes are delivered with SAP HANA, you can also create user-defined hints and apply these to a select statement.

Capture and Replay

SAP HANA capture and replay allows you to capture the workload of a production system and to replay the captured workload on a target system. This can help you evaluate potential impacts on performance or stability after a change in hardware or software configuration.

SQL Plan Stability

SQL Plan Stability is a feature to protect the performance of queries by capturing query execution plans in a source system and reusing them in a target system to regenerate the original query plan. This is typically used after a system upgrade in order to guarantee the performance of a query in the new system. Plan Stability can be administered in SAP HANA cockpit or from the command line as described in the *SAP HANA Troubleshooting and Performance Analysis Guide*.

Managing Bandwidth

To help administrators manage bandwidth a set of configuration parameters is available to manage the HANA File Input/Output layer in order to balance the throughput of different devices and channels. The parameters can be applied to specific usage types such as data volume, log files, backup.

Related Information

Caching [page 149]
Hints [page 158]
Managing Bandwidth (File IO Throttle) [page 164]

Links to other documents:

Monitoring and Analyzing Performance in SAP HANA Studio (SAP HANA Studio)
Monitoring, Analyzing, and Improving Performance(SAP HANA Cockpit)
Capture and Replay (SAP HANA Cockpit)
Plan Stability (SAP HANA Troubleshooting Guide)

8.7.1 Caching

A number of options are available to use caching to improve performance.

Caching is used widely in SAP HANA as a strategy to improve performance by re-using queried data rather than re-reading and processing the data every time it is requested. Query result caches are used for caching data and the Plan Cache (see following topic) is available for query optimization by storing execution plans of frequently executed SQL statements.

The query result cache applies to column store tables and views and offers most potential to improve performance in situations where data is predominantly read; this is because updates to the base tables invalidate this type of cache. Configuration options for this cache include defining a maximum memory budget for the cache and defining a list of allowed tables or views to which caching is applied. This is described in detail in SAP Note 2014148 *Guidelines for Using the Query Result Cache*.

The static result cache (sometimes referred to as *cached views***) and the dynamic result cache** are further applications of caching. The static result cache is created for a specific view and remains valid for the duration of a user-defined retention period. The dynamic result cache is similar but does not have a retention period; it guarantees transactional consistency by maintaining delta records of all changes applied to the underlying table. Details of the static and dynamic result cache can be found in the *SAP HANA Troubleshooting and Performance Analysis Guide*.

Both SAP HANA cockpit and SAP HANA Studio have features for managing and monitoring cache activity.

Cache Size

Overall cache size is controlled by the resource manager. Caches may not be strictly limited in size but will grow on demand dependent on available memory. Objects are automatically unloaded from the cache based on their swappable disposition and least recent usage values. However, you can define upper and lower limits for the global cache size using a set of configuration parameters in the `memoryobjects` section of the global.ini file. You can set values either as an absolute number of bytes or as a percentage of available memory (for

example: `global_cache_upper_limit_abs`, `global_cache_lower_limit_rel`). The percentage values are calculated in relation to the global allocation limit (GAL) or process allocation limit (PAL) whichever is the smaller value (that is: Unit: % of MIN(GAL, PAL)). Refer to the parameter descriptions in the *SAP HANA Configuration Parameter Reference* for details.

Details of all caches including size and hit count are available in the (resettable) monitoring view M_CACHES where each cache can be identified by a unique CACHE_ID value. Further information on caches is also available in SAP Note 2502256 - FAQ: SAP HANA Caches.

Related Information

SQL Plan Cache [page 152]

Links to other documents:

SAP HANA Configuration Parameter Reference
M_CACHES (SAP HANA SQL Reference Guide for SAP HANA Platform)
Result Cache (SAP HANA Troubleshooting and Performance Analysis Guide)
Data Cache (SAP HANA Cockpit)
SQL Plan Cache (SAP HANA Cockpit)
SQL Plan Cache (SAP HANA Studio)
SAP Note 2502256
SAP Note 2014148

8.7.1.1 Cache Management

In addition to monitoring cache usage, features are available to manage the size of caches and also to remove specific entries from a cache.

Two options are available to manage caches:

- You can use the REMOVE CACHE ENTRY statement to remove specific cache entries instead of clearing a cache completely.
- You can manage the maximum size of individual cache instances or groups of cache instances using configuration parameters.

Remove Selected Cache Items

You can improve the effectiveness of a cache by removing specific obsolete entries rather than completely clearing all entries from the cache. The REMOVE CACHE ENTRY statement can be used for any cache listed in the monitoring view M_CACHE_ENTRIES.

The monitoring view M_CACHE_ENTRIES has details about all cache entries including cache ID, memory size, user name, and so on. Because the CACHE_ID value is not a unique identifier, the view also contains an

ENTRY_HASH value, which is required to identify individual entries. You could therefore use a query like the following to firstly analyze the currently cached items:

```
select distinct memory_size, cache_id, entry_id, entry_hash, user_name from
sys.m_cache_entries;
```

You can then use the ENTRY_HASH value and the CACHE_ID value with the REMOVE CACHE ENTRY statement to remove one or multiple items. Only items from the named cache are deleted:

```
ALTER SYSTEM REMOVE CACHE '<cache_id>' ENTRY ('<entry_hash>', '<entry_hash>')
```

This can also be used in distributed systems. Be sure to verify the result on all hosts where the statement is run to test for false positive results, which may be returned to the source host. False positives indicate that a deletion was successful; this may occur when the cache does not exist on a given host.

An error message is generated if the syntax is invalid, but a statement may appear to be successful even though no cache entry is removed. The following statement, for example, does not return an error:

```
ALTER SYSTEM REMOVE CACHE 'HierarchySqlFunctionCache' ENTRY ('')
```

Monitoring Usage

If you want to monitor usage of this REMOVE CACHE ENTRY statement, check the CALL_COUNT value for component CACHE MANAGEMENT in the M_FEATURE_USAGE view. Note, however, that the CALL_COUNT value includes both REMOVE CACHE and CLEAR CACHE executions. The statistics include all occurrences of executing these statements, not just the successful ones.

Limit Cache Memory Usage

The cache framework gives fine-grained control over the memory consumption of individual cache instances. Caches are not strictly limited in size but grow on demand depending on available memory. In general, the amount of memory available for the cache globally is limited by settings of these configuration parameters, which are available in the *memoryobjects* section of the `indexserver.ini` file in the SYSTEM database:

- *global_cache_upper_limit_abs* applies an upper limit entered as a number of bytes.
- *global_cache_lower_limit_abs* applies a lower limit entered as a number of bytes.
- *global_cache_upper_limit_rel* applies an upper limit entered as a percentage of available memory.
- *global_cache_lower_limit_rel* applies an lower limit entered as a percentage of available memory.

The percentage values are calculated in relation to the global allocation limit (GAL) or process allocation limit (PAL), whichever is the smaller value (that is: Unit: % of MIN(GAL, PAL)). By default no limit is applied, but values could be set at this level from the SQL command line if required, for example:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET
('memoryobjects', 'global_cache_upper_limit_abs') = '10240' WITH RECONFIGURE;
```

More specifically, cache instances are organized hierarchically in named groups so that they can be individually identified and the size of each cache instance can be individually managed. As with the global limits you can apply upper and lower memory limits (unit bytes) for a cache group or for an individual instance of the group.

The following example shows a group of MultiModel cache instances. The group and cache instance names are listed in CACHE_ID column of the monitoring view M_CACHES:

- Caches/MultiModel
- Caches/MultiModel/HierarchySqlFunctionCache
- Caches/MultiModel/HierarchyCache
- Caches/MultiModel/HierarchyCacheNoInvalidate

Corresponding configuration parameters are available in the *memoryobjects_properties* section of the `indexserver.ini` file in the SYSTEM database. The parameter names are similar but not identical to the CACHE_ID names. This first example sets an upper limit for the MultiModel group as a whole:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET
('memoryobjects_properties', 'multi_model/upper_limit_abs') = '10240' WITH
RECONFIGURE;
```

The following example sets an upper limit for a single instance in the group:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET
('memoryobjects_properties', 'multi_model/HierarchyCache/upper_limit_abs') =
'10240' WITH RECONFIGURE;
```

If the cache size exceeds the defined limit, entries are removed from the cache on the basis of a least-recently-used algorithm.

Memory usage statistics for each cache can be read from the M_CACHES monitoring view.

The view M_MEMORY_OBJECTS has details including counts for hits and misses for each cache instance. The category of cache instance is available in the RESOURCE_CATEGORY column (the value 'CACHE' indicates that the object has been used as a cache).

Related Information

Link to SAP HANA Database SQL Reference Guide
M_CACHE_ENTRIES System View
M_FEATURE_USAGE System View

8.7.1.2 SQL Plan Cache

The plan cache stores execution plans of frequently executed SQL statements so that they can be reused rather than having to recompile the plan each time the statement is run. Administrators can also use the plan cache for performance analysis to get an overview of what statements are executed in the system.

Plan Cache for Administrators

For administrators, the SQL plan cache is useful for monitoring overall SQL performance as it provides statistics on compiled queries such as the number of executions, runtime details, and lock/wait statistics.

You can use this information to get insights into frequently executed queries and slow queries and identify potential candidates for optimization.

You can view the plan cache in SAP HANA cockpit by clicking on the *Top SQL Statements* card in your resource's *Database Overview*; then, on the *Monitor Statements* screen select the *SQL Plan Cache* page. See cockpit documentation (*Monitor and Analyze Statements with SQL Plan Cache*) for more information.

Eager Eviction

SQL statements stay in the Plan Cache as long as they are required, but entries which are of little value are evicted from the cache. Eviction takes place automatically by default, but if you wish to manage this more carefully (or turn eviction off completely) configuration settings are available to manage the cache on the basis of usage frequency or a more balanced usefulness score which is automatically calculated. Configuration parameters for this are in the `[SQL]` section of the indexserver.ini file.

To manage eviction you must firstly activate the `plan_cache_eager_eviction_mode` parameter by changing its default value (auto) to 'custom'.

By default, cached plans that are not used for more than two days are automatically evicted. You can change this retention period and also the frequency with which the plan cache is scanned for old plans.

Parameters for Retention Period	Default	Usage
plan_cache_eager_eviction_mode	auto	Set this to 'custom' if you wish to change the default eviction settings. Set this to 'off' to disable plan cache eager eviction.
plan_cache_eager_eviction_time_based_retention_time	2160	Sets the minimum retention period in minutes for time-based eviction. Setting this to a negative number disables this feature.
plan_cache_eager_eviction_time_based_scan_period	180	Sets the frequency, in minutes, of how often time-based eviction occurs.

For score-based eviction, the plan cache periodically assigns a score to each plan to evaluate it in terms of a range of criteria such as: Frequency of execution, Time since last execution.

You can configure score-based eviction to evict a certain percentage of low-scoring plans. By default, the bottom 10 percent of plans are evicted each time the scan of the cache is run.

Parameters for Score-based Eviction	Default	Usage
plan_cache_eager_eviction_score_based_rate	3	Sets the percentage of low-score plans to be evicted. Setting this to a negative number disables this feature.
plan_cache_eager_eviction_score_based_scan_period	60	Sets the frequency, in minutes, of how often score-based eviction occurs.

Selective Removal of Plan Cache Entries

In some circumstances it may be necessary to clear the plan cache, this may be necessary if, for example, problems are caused by invalid entries. One option is to completely clear the plan cache using the following CLEAR statement:

```
ALTER SYSTEM CLEAR SQL PLAN CACHE
```

However, this takes time and there will be a considerable performance overhead afterwards as all SQL statements will have to be parsed and recompiled from scratch. A better approach to improving the quality of the plan cache is to delete individual plans selectively. You can do this with the REMOVE statement shown here either by identifying a specific plan number or by using a WHERE clause:

```
ALTER SYSTEM REMOVE SQL PLAN CACHE [ ENTRY <plan_id> | WHERE <condition> ]
```

The following command, for example, removes the plan cache entry ID 2374637 from the SQL plan cache:

```
ALTER SYSTEM REMOVE SQL PLAN CACHE ENTRY 2374637;
```

The WHERE clause operates on the `M_SQL_PLAN_CACHE` monitoring view which contains over 90 columns. This option therefore provides many possibilities for removing specific items from the cache with minimal negative impact on performance. The following examples illustrate this; all plans meeting the condition are removed from the cache:

This example removes entries which are not frequently used:

```
ALTER SYSTEM REMOVE SQL PLAN CACHE WHERE EXECUTION_COUNT < 5
```

This example removes entries which are not valid:

```
ALTER SYSTEM REMOVE SQL PLAN CACHE WHERE IS_VALID = 'FALSE'
```

This example removes entries which are querying specific tables:

```
ALTER SYSTEM REMOVE SQL PLAN CACHE WHERE ACCESSED_TABLE_NAMES LIKE '%AAA%'
```

For both REMOVE and CLEAR statements, the OPTIMIZER ADMIN privilege is required. Refer to the *SAP HANA SQL Reference Guide* for full details of the syntax.

Auto Recompilation

An 'auto recompilation' feature is available which is able to detect the poor performance of a query and then on the basis of this assessment remove the query plan entry from the cache so that recompilation is triggered.

The feature is disabled by default but can be enabled by setting the configuration parameter `plan_cache_auto_recompilation_enabled` in the `sql` section of the <service>.ini file to **`True`**.
When the feature is enabled recompilation statistics are maintained in the monitoring view M_SQL_PLAN_CACHE_AUTO_RECOMPILATION_STATISTICS.

The monitoring view M_SQL_PLAN_CACHE_EXECUTION_ENGINE_STATISTICS shows execution statistics per execution engine. This view may therefore have two entries for a query string: if the query runs with HEX and

non-HEX there is one row for each engine; you can use this information to compare the performance of each plan. If the query only runs with either HEX or non-HEX then there is only a single row.

Monitoring Views

The following monitoring views have details of cached plans:

- M_SQL_PLAN_CACHE_OVERVIEW includes, for example, timestamp values of the last scans for time/score-based eviction.
- M_SQL_PLAN_CACHE includes, for example, timestamp values of the last execution time of a plan and also score details of current plan cache entries.
- M_SQL_PLAN_STATISTICS includes a value for last invalidation reason, such as, LOW_SCORE or EXPIRED.
- M_SQL_PLAN_CACHE_AUTO_RECOMPILATION_STATISTICS holds details about triggered auto recompilations.
- M_SQL_PLAN_CACHE_EXECUTION_ENGINE_STATISTICS The EXECUTION_ENGINE column indicates whether the HEX or non-HEX engine was used to compile the query plan. Where both engines are used the view shows one row for each engine for each PLAN_ID.

Related Information

Links to other documents
ALTER SYSTEM REMOVE SQL PLAN CACHE ENTRY Statement (System Management)
M_SQL_PLAN_CACHE_OVERVIEW System View
M_SQL_PLAN_CACHE System View
M_SQL_PLAN_STATISTICS System View
Monitor and Analyze Statements with SQL Plan Cache

8.7.1.2.1 SQL Plan Cache for Parameterized Queries

Plan cache variants are used to save execution plans of parameterized queries. This feature is referred to as Parameter Sensitive Optimization (PSO).

In the case of parameterized queries it is possible to store multiple SQL plans ('variants') that correspond to the different parameter sets which have been used for the query. Plan variants increase the probability of matching the query with an optimal cached SQL plan for the given parameter set.

The feature can be enabled by activating a configuration parameter, in which case the matching process is applied generally to all incoming queries. Alternatively, you can apply the feature to individual queries by appending the `plan_variant` hint to the statement as shown in the examples in this topic.

The `plan_variant` hint is similar to the `use_hex_plan` hint in that it also prefers the HEX execution engine when possible.

When the feature is active, new variant plans are created and cached when required but there's a limit on the number of saved variant plans. When the limit is reached plans are removed on the basis of usage: the least

recently used plans are deleted from the cache first. The default maximum number of stored plans in the plan variant cache for each sql statement is 10, but you can set a different limit. For checking if the cache has an optimal plan for a given parameter set, parameterized filters are considered. The number of filters to consider is limited: the default maximum number of filters is 3, but you can also set a different limit.

Activation and Configuration

The following configuration parameters are available in the 'sql' section of the indexserver.ini file:

Parameter	Description
plan_variant_enabled	The feature is disabled by default. Set this parameter to 'true' to enable the feature generally so that it is applied to all incoming queries. Note that if you disable this feature after it has been in use, existing cached variant plans in the SQL plan cache are kept and are available when the feature is enabled again.
plan_variant_plan_count	You can modify this parameter to set the maximum number of plans saved.
plan_variant_filter_count	You can modify this parameter to set the maximum number of variant filters saved.

Example of changing a configuration parameter in SQL:

```
alter system alter configuration('indexserver.ini','system') set
('sql','plan_variant_enabled')='true' with reconfigure;
```

Enabling this feature through configuration does not automatically apply variant plans to all parameterized queries. The SQL optimizer first checks whether the query includes any variant filters—these are parameterized filters that can influence the choice of cached SQL execution plan based on the input values. If no such filters are found, a plan variant will not be applied.

To influence this behavior, the plan_variant hint can be used to suggest specific filters as variant filters. This hint helps guide the SQL optimizer to treat certain parameterized filters as important for cached SQL plan selection, increasing the likelihood of matching a suitable plan variant. For an example, see the next section on how to prioritize a particular parameterized filter during plan matching.

Applying Plan Cache Variants

To apply the feature to a single query append the `plan_variant` hint as a parameter to the query. The various options are illustrated here:

Apply the feature to the current query and where possible apply an existing cached plan:

```
select * from T1 where C1 = ? with hint (plan_variant);
```

Additionally, you can pass column arguments (table names and column names) with the `plan_variant` hint so that a certain parameterized filter is preferred when deciding on a cache hit, for example:

```
select * from T1 where C1 = ? with hint (plan_variant(T1.C1));
```

Note that the reference to the provided column is not validated. This means that if an invalid column is supplied as part of the argument, the plan variant will still be applied, but no parameterized filter will be preferred.

To exclude the current query so that a variant plan is not applied, use the hint `no_plan_variant`:

```
select * from T1 where C1 = ? with hint (no_plan_variant);
```

If you use `no_plan_variant` when a column argument is provided then filters with this argument are avoided as variant filter. This may be useful in statements where a number of parameter values are given but some values should not be considered:

```
select * from T1 where a1 = ? and b1 = ? and c1 = ? with hint (plan_variant,
no_plan_variant(c1));
```

Managing Cached Variants

Change the Filter or Plan Count

You can change the maximum number of variant plans saved by using the `plan_variant_plan_count` hint providing the maximum number as a parameter. You may need to increase the number if the feature fails to cache enough plans for your use cases:

```
select * from T1 with hint (plan_variant, plan_variant_plan_count(20));
```

You can change the maximum number of variant filters saved by using the `plan_variant_filter_count` hint providing the maximum number as a parameter. You may need to increase the number if the feature fails to cache enough optimal plans for your use cases:

```
select * from T1 with hint (plan_variant, plan_variant_filter_count(10));
```

Clearing the Cache

Multiple plan variant entries are connected with an SQL Plan Cache entry. You can remove an SQL Plan Cache entry by referring to the `plan_id` value and the connected plan variant entries will be removed:

```
alter system remove sql plan cache entry <plan_id>;
```

You can retrieve the plan ID from the monitoring view M_SQL_PLAN_VARIANT_STATISTICS view as described below. See also the topic 'SQL Plan Cache' for more details on cache management.

Monitoring

Details of the caching and hit count are available in monitoring view M_SQL_PLAN_VARIANT_STATISTICS. This view is resettable so that statistics from a given point in time can be monitored (see ALTER SYSTEM RESET MONITORING VIEW). The view includes over 60 data items including the ID values for plan and variants, the column VARIANT_FILTERS shows details of any filters which have been applied. The statement string and a hash value for the statement are also available and can be used to query the view:

```
select * from SYS.M_SQL_PLAN_VARIANT_STATISTICS_RESET
where STATEMENT_STRING like 'select * from T1 where C1 = ? with hint
(plan_variant)';
```

Information about memory usage can be derived from the monitoring view M_HEAP_MEMORY. You can query the view on the basis of CATEGORY to see how much memory is currently allocated for plan variants:

```
select * from M_HEAP_MEMORY where category like '%PlanVariant%';
```

Explain Plan

Explain plan can also be used with a plan variant ID in combination with a plan ID. You can select the PLAN_ID and PLAN_VARIANT_ID values from the M_SQL_PLAN_VARIANT_STATISTICS monitoring view as illustrated above and then use these values in an EXPLAIN PLAN statement to retrieve details of a cached variant plan:

```
EXPLAIN PLAN FOR SQL PLAN CACHE ENTRY 1 VARIANT 1;
```

Related Information

SQL Plan Cache [page 152]

Links to SQL Reference Guide

HINT Details

EXPLAIN PLAN Statement (Data Manipulation)

M_SQL_PLAN_VARIANT_STATISTICS System View

ALTER SYSTEM RESET MONITORING VIEW Statement (System Management)

8.7.2 Hints

Predefined and user-defined hints can be used to influence the execution of SQL queries.

Hints are instructions for the SAP HANA database server which influence the way a database request is processed. The SQL Optimizer determines the access path of a query but you can override the optimizer by specifying hints in the query to enforce a certain access path. Hints are typically used to optimize SAP HANA performance or memory consumption and have no effect on the result set of the request. Predefined hints for various purposes are delivered with SAP HANA, you can list these from the HINTS system view; refer to the *SAP HANA SQL Reference Guide for SAP HANA Platform* for more details. You can also create user-defined hints and apply these to a select statement. If a query has both a user-defined and a system-defined hint, the user hint is used in preference.

For convenience you can associate queries with specific hints so that the hint is always applied to the query at runtime. This can be done in SQL using the statement ALTER SYSTEM ADD STATEMENT HINT and also in the Statement Hints app of SAP HANA cockpit (see also STATEMENT_HINTS system view).

As an alternative to linking a hint to a select statement you can pin a hint to a specific execution plan in the SQL plan cache. Pinning is applied using the ALTER SYSTEM PIN SQL PLAN CACHE ENTRY instruction and uses the execution plan PLAN_ID value to link to a hint. Refer to the *SAP HANA SQL and System Views Reference Guide* and to SAP Note 2400006 *FAQ: SAP HANA Statement Hints* for details.

Many hints are available for use with cached data or data replicas to provide 'hint-based routing' functionality so that administrators have control over exactly which source is used when a choice of data sources is available, see the following section *Using Hints to Query Data Snapshots* for details and examples of the configurable hint classes which are available.

Related Information

Using Hints to Query Data Snapshots [page 159]

Links to other documents:

HINT Details (SAP HANA SQL Reference Guide for SAP HANA Platform)

Statement Hints (SAP HANA cockpit)

SAP Note 2400006

8.7.2.1 Using Hints to Query Data Snapshots

Several features in SAP HANA use data snapshots to improve performance. You can use configurable hint classes as a standard way of controlling at run time how the data is selected, either from the snapshot or from the database.

Several features in SAP HANA use data snapshots or replicated data to improve performance; this includes:

- Result Cache
- Asynchronous Table Replication
- System Replication (Active/Active Read Enabled)

Snapshots carry the risk of holding stale data and administrators need to be able to specify a maximum time value up to which the replicated data may lag behind the live data. The features listed above all use the RESULT_LAG() hint with a set of standard configurable hint classes as a common way of controlling how the data is selected at run time (also referred to as hint-based routing). Hint classes give the administrator a tool to balance a system in terms of query response time, query load, resource utilization and freshness of data. Moreover, they de-couple SAP HANA features from application development and administrator choices.

Using the Hint RESULT_LAG()

The RESULT_LAG() hint takes two parameter values to determine:

- Which is the preferred data source (snapshot or live data)
- How much time the retrieved data may lag behind the live data.

The syntax for the RESULT_LAG() hint is as shown here:

```
WITH HINT (RESULT_LAG(“<short_class_name>”, [seconds]))
```

Class name: the following pre-defined hint classes are available for use with this hint, these, and options for configuring them, are described in detail below. Only the last two elements of the name are used in SQL statements, for example 'hana_cache', 'hana_short' and so on:

- hint_result_lag_**hana_cache** - for Result Cache, the default lag time value is the cache retention time
- hint_result_lag_**hana_atr** - for Asynchronous Table Replication, the default lag time value can be defined as a configuration parameter
- hint_result_lag_**hana_sr** - for System Replication (Active/Active), the default lag time value can be defined as a configuration parameter
- hint_result_lag_**hana_es_atr** - for the extended storage replica node, the default lag time value can be defined as a configuration parameter
- hint_result_lag_**hana_short**
- hint_result_lag_**hana_long**

Note that customers and developers may also add classes of their own (and corresponding configuration values) but these must not use the hana_* or sap_* naming convention.

Seconds: This parameter is an optional time value in seconds; this is the maximum time lag up to which the data is judged to be acceptable; if the snapshot or replica is older then it will not be used and the query is routed to the data source. If no seconds value is given the default value will be used. In some cases this feature can be disabled by setting a configuration option - see details below.

Configuration of Routing

Each hint class has its own section in the indexserver.ini file ([hint_result_lag_hana_short] for example, sections must also be added for user-defined classes) where one or more parameters are available. All classes have the `enable_features` parameter; this is the 'routing' parameter which is used to define an order of preference for the data source used. The following table shows the default enable_features setting for each class, for example, the *hana_long* class is defined by default with the sequence, 'resultcache,atr,sr', meaning: data from the resultcache is preferred, if not then data from a replica table (asynchronous table replication), otherwise data from a secondary system (system replication). The same maximum lag time value defined is applied to the whole sequence of data sources (an example is given below):

Class	Parameter	Default Setting (Preferred Source Sequence)
hana_cache	enable_features	resultcache
hana_atr	enable_features	atr
hana_sr	enable_features	sr
hana_es_atr	enable_features	es,atr
hana_short	enable_features	atr, sr
hana_long	enable_features	resultcache,atr,sr

More detail of how these classes are used in each of the three application areas is described in the following subsections.

Hint Class for Result Cache

Using the class `hana_cache` the result cache of the target view / table function is exploited if it is enabled and applicable. This is shown in the following example:

```
SELECT * FROM V1 WITH HINT( RESULT_LAG ( 'hana_cache', 30) );
```

If a value for the seconds (time) parameter of the hint is given it overrides any retention time value which has been defined for the view. For example, the 30 second time lag defined in the above example would override the hundred minute retention period defined when the cache was created:

```
ALTER VIEW v1 ADD CACHE RETENTION 100;
```

The result cache and dynamic result cache are described in detail in the *SAP HANA Troubleshooting and Performance Analysis Guide*.

Hint Class for Asynchronous Table Replication

You can use the `hana_atr` class to access replicated tables.

```
SELECT * FROM T1 WITH HINT( RESULT_LAG( 'hana_atr', 10 ));
```

If the current lag time for the data on the replica is less than the stated value for the [seconds] parameter (10 seconds in this example) then the query is executed on the replicated table, otherwise it would be executed on the source.

More Configuration Options for Class hana_atr

Parameter:	**atr_default_lag_time**
Purpose	The default lag time value for this class. This value will be applied if no seconds value is entered with the hint on the SQL command line.
Default	-1 In this case if no seconds value is entered with the hint the query is always routed to the replica.
Unit	seconds
Parameter:	**atr_ignore_lag_time**
Purpose	Set this to true to disable the check for the specified lag time so that the query will always access replica tables regardless of the value entered with the hint and the default lag time setting.
Default	false

For more information refer to the *Asynchronous Table Replication* section of this guide.

Hint Class for System Replication (Active/Active read enabled)

You can use the class `hana_sr` for system replication and select data from the secondary system. This is shown in the following example:

```
SELECT * FROM T1 WITH HINT( RESULT_LAG ('hana_sr', 60) );
```

If this is used in a situation where the result cache might also be available, the following table shows specifically which data source is used in either of the two possible cases (result cache available yes or no):

enable_features	**Result Cache Available?**	**Data Source**
sr,resultcache	Yes	Secondary (cache used)
sr,resultcache	No	Secondary
resultcache,sr	Yes	Primary (cache used)

enable_features	Result Cache Available?	Data Source
resultcache,sr	No	Secondary
sr	Yes	Secondary
sr	No	Secondary
resultcache	Yes	Primary (cache used)
resultcache	No	Primary

This table shows the behavior on the primary system. If the application connects to the secondary system directly the result cache can be also used at secondary. For more information refer to the *Active/Active (Read Enabled)* section of this guide.

More Configuration Options for Class hana_sr

Other parameters are also available for the *hana_sr* hint class:

Parameter:	sr_default_lag_time
Purpose	The default lag time value for this class. This value will be applied if no value is entered with the hint on the SQL command line.
Default	-1 In this case if no seconds value is entered with the hint the query is always routed to the secondary.
Unit	seconds

Parameter:	sr_ignore_lag_time
Purpose	Set this to true to disable the check for the specified lag time.
Default	false

Parameter:	sr_enable_primary_redirection_for all errors
Purpose	The routed statement will be redirected to primary for all errors.
Default	true

Parameter:	sr_enable_primary_redirection
Purpose	This parameter is used for error handling: if an out of memory error occurs on the secondary, or if the lag time is exceeded, then the routed statement will be redirected to the primary and executed.
Default	true

Examples Using HANA_SHORT and HANA_LONG

You can use `hana_short` and `hana_long` classes in combination with other application-specific hint classes. Here, we give two examples; in the first a value for the seconds parameter is given, and in the second the default values for each data source as specified in the configuration file are used. The configuration file for `hana_long` for example, could be as follows:

Sample Code

```
[hint_result_lag_hana_long]
   enable_features = resultcache,atr,sr
   atr_default_lag_time = 10
   sr_default_lag_time = 20
```

The `enable_features` parameter specifies the sequence of data sources as: result cache preferred, secondly table replica, otherwise system replication secondary system.

Example 1 In this example a *seconds* value is given in the query and this value is used in all three cases to evaluate whether the cached/replicated data is acceptable.

```
SELECT * FROM V1 WITH HINT( RESULT_LAG ('hana_long') , 10 );
```

First if the result cache data is less than the seconds parameter value (10 seconds) the query will read data from here. If the result cache is too stale then the age of the replicated table data will be evaluated against the seconds parameter value and if this data is too stale then the system replicated data will be evaluated to determine whether to execute the query on the primary or secondary system. If none of the cached sources is available (seconds value smaller than the current lag time) then the primary system is accessed.

Example 2 In the following example (referring to the same sequence of three data sources) a time value is not given in the query, and the default lag values for each data source are used to evaluate whether the data source is acceptable:

```
SELECT * FROM V1 WITH HINT( RESULT_LAG ('hana_long') );
```

The default values are: for result cache the predefined retention period, for `atr` and `sr` the ini parameter values defined in the [hint_result_lag_hana_long] section apply, that is, `atr_default_lag_time` for asynchronous replication (10 seconds in the sample code above) and `sr_default_lag_time` for system replication (20 seconds in the sample code above).

Related Information

SAP HANA Troubleshooting and Performance Analysis Guide
Active/Active (Read Enabled) [page 852]
Operations for Asynchronous Table Replication [page 395]

8.7.3 Managing Bandwidth (File IO Throttle)

A set of configuration parameters is available to manage the HANA File Input/Output layer in order to balance the throughput of different devices and channels.

Configuration Options

The HANA File IO layer tries to use the full bandwidth of the underlying IO device/channel to guarantee maximum database performance. However, services, service instances and components using the same IO device or channel compete for IO bandwidth and administrators may be able to optimize throughput of a whole service, or certain components within a service, by setting values for the configuration parameters described here.

The configurable parameters are in the [fileio] section of the relevant <service>.ini configuration file:

- max_throughput - limitation applied to total throughput for the service as a whole
- max_read_throughput - limitation applied to read requests
- max_write_throughput - limitation applied to write requests

The parameters define an upper limit for IO throughput in megabytes per second (MB/s). Each IO request observes the limitations defined and the file IO layer balances requests within those limits. For example, with the following configuration, read requests would be throttled to 80MB/s while write requests (and throughput for the service as a whole) would be throttled to 100MB/s:

```
max_throughput=100
max_read_throughput=80
```

By default, no throttling is applied for each parameter (the default value for all parameters is zero), but limits applied by other matching parameters are respected. Throttling values do not guarantee throughput up to the limits specified; these speeds may not be achieved if there are other reads and writes competing for the same IO bandwidth.

To give more precise control, each parameter can be applied multiple times with one of the following usage types which then applies the throttle to the corresponding file type. These usage types refer to the type of file the throttle is applied to, not to the type of IO request. So for example, usage type DATA refers to all operations on the data volume file(s). Note the scope of the different BACKUP parameters:

Usage Type	Applies to
DATA	Data volume
LOG	Log volume An option to exclude this type is supported using the value '-1', see example below.
LOG_MIRROR	
DATA_BACKUP	Data backup volume

Usage Type	Applies to
LOG_BACKUP	Log backup volume
CATALOG_BACKUP	Catalog backup volume
ROOTKEY_BACKUP	
BACKUP	BACKUP does not cover data, log and catalog backups - use DATA_BACKUP, LOG_BACKUP, CATALOG_BACKUP for these specific types - see examples. BACKUP is only for use on backup/recovery components such as diagnosis tool (for example `hdbbackupdiag`) and operations on backup files, it does not apply to external tools like `backint` - see example below.

Example

```
max_write_throughput[DATA] = 100
max_read_throughput[BACKUP] = 100
max_throughput[DATA_BACKUP] = 200
```

Further examples:

`max_throughput = 250`	Applies a limit of 250MB/s throughput for the service as a whole.
`max_write_throughput[BACKUP] = 100`	Available bandwidth for writing to backup medium is restricted to a maximum of 100MB/s. This can be used to throttle writes to local backup files but writes to `backint` endpoints cannot be throttled this way.
`max_write_throughput[LOG] = -1`	The value '-1' can be used for the LOG type to prevent throttling of requests which write to log files since this would have a severe impact on overall DB performance. If this value is set then no throttling is applied to log file writes.

Although quite complex rule sets can be defined, for evaluation and testing purposes (see Monitoring Queues below), you are recommended to set only one or two parameter values. Setting very small values is not recommended. Also, the balancing algorithm of the FileIO layer is optimized to work with throttle values greater than the maximum page size (64MB), it may therefore be difficult to monitor and test throttle values smaller than 64.

Monitoring Queues

Before requests are handed over to the file system for execution they are queued in the SubmitQueue. However, if throttling is active, requests are first added to the ThrottleQueue which applies the throttle parameters and then hands them over to the SubmitQueue. For example, if you have configured a 100MB/s throttle, the ThrottleQueue will hand over 100MB of requests to the SubmitQueue every second.

There are two monitoring views you can use to see the effect of configured FileIO throttles and the queuing activity:

- M_VOLUME_IO_TOTAL_STATISTICS: can be used to monitor physical and effective throughput (see following note on resetting the aggregated values)
- M_VOLUME_IO_SUBMIT_STATISTICS: can be used to monitor FileIO layer queues

Monitoring via M_VOLUME_IO_TOTAL_STATISTICS:

The unit value of the SIZE fields in this view is byte and the unit value of the TIME fields in this view is microsecond, so the throughput 'T' of the underlying IO device/channel can be computed as:

```
T = (TOTAL_(IO/READ/WRITE)_SIZE * 1000 * 1000) / (TOTAL_(IO/READ/WRITE)_TIME *
1024 * 1024)
```

Before applying throttling, check the physical throughput value T and then decide on meaningful values for the throttle parameters.

Similarly, the throughput 't' observed by a user of the FileIO layer can be computed as:

```
t  = (TOTAL_(IO/READ/WRITE)_SIZE * 1000 * 1000) / (TOTAL_PROCESSING_(READ/
WRITE)_TIME * 1024 * 1024)
```

If throttling is activated, T should stay more or less stable, but t should reflect the configured throttles.

> **Note**
>
> All columns of this view aggregate values over time. To get meaningful values for a time interval you can do either of the following:
>
> - Refer to statistics server view HOST_VOLUME_IO_TOTAL_STATISTICS (and the corresponding delta columns)
> - Use the 'reset' version of the monitoring view which has a resettable counter and only contains the values that have accumulated since the last reset of the main view. You can reset the counter for this view by running the following reset command which restarts aggregation at the current time:
>
> ```
> ALTER SYSTEM RESET MONITORING VIEW SYS.M_VOLUME_IO_TOTAL_STATISTICS_RESET;
> ```
>
> Refer to the *SAP HANA SQL Reference Guide* for more details and the 'Analyzing I/O Throughput and Latency' section of the *SAP HANA Troubleshooting and Performance Analysis Guide*.

The following table gives a commentary and explanation of the main columns used for monitoring:

Columns	Values
TOTAL_READ_SIZE TOTAL_WRITE_SIZE	The total size (unit: byte) read from or written to the usage in column TYPE. The total size read and written is: <total IO size> = TOTAL_READ_SIZE + TOTAL_WRITE_SIZE Use this formula for total IO throughput calculations: `T = (TOTAL_(READ/WRITE)_SIZE * 1000 * 1000) / (TOTAL_(READ/WRITE)_TIME * 1024 * 1024)`

Columns	Values
	or T = (<total io size> * 1000 * 1000) / (TOTAL_IO_TIME * 1024 * 1024)
TOTAL_IO_TIME TOTAL_READ_TIME TOTAL_WRITE_TIME	These values show the time spent (unit: microsecond) executing IO requests on the physical device/channel. Note that because of overlapping reads and writes, TOTAL_IO_TIME is not the same as TOTAL_READ_TIME + TOTAL_WRITE_TIME. Throttling has little impact on these times, they simply reflect the physical capabilities of the underlying IO device/channel.
TOTAL_PROCESSING_TIME TOTAL_PROCESSING_READ_TIME TOTAL_PROCESSING_WRITE_TIME	These values show the total time IO requests have spent in the FileIO layer with the throttling limitations applied. These are the times a user can measure when they execute IO requests. Use this formula for total processing throughput calculations: t = (TOTAL_(READ/WRITE)_SIZE * 1000 * 1000) / (TOTAL_PROCESSING_(READ/WRITE)_TIME * 1024 * 1024) or t = (<total io size> * 1000 * 1000) / (TOTAL_PROCESSING_TIME * 1024 * 1024)

Monitoring via M_VOLUME_IO_SUBMIT_STATISTICS

This view shows which IO requests have been added to the hierarchy of queues of the FileIO layer.

View	Shows
DISPATCH_SUBMIT_QUEUE_*_SIZE	These columns show the total size of requests added directly to the SubmitQueue. If no throttling is applied all IO activity is reflected here showing how fast the upper layers push requests to the FileIO layer.
DISPATCH_THROTTLE_QUEUE_*_SIZE	These columns show the total size of requests added to the ThrottleQueue showing how fast the upper layers push requests to the FileIO layer when throttling is active.
THROTTLED_DISPATCH_SUBMIT_QUEUE_*_SIZE	These columns show the total size of requests moved from the ThrottleQueue to the SubmitQueue showing how fast the requests are pushed.

Related Information

HOST_VOLUME_IO_TOTAL_STATISTICS View (Embedded Statistics Service)
M_VOLUME_IO_TOTAL_STATISTICS System View
Analyzing I/O Throughput and Latency

8.7.4 SQL Analyzer Tool for SAP HANA

The SQL analyzer tool for SAP HANA is now available as a separate application in SAP Business Application Studio.

The SQL analyzer tool for SAP HANA helps you visualize and analyze SQL statement execution plans. You can use it by exporting the query plan of a statement and importing it into the tool for analysis. The execution plan for a query can be downloaded in the required PLV format from SAP HANA cockpit. Real-time analysis and visualization of currently running statements are also possible using the SQL Plan Execution Monitor.

This tool supersedes the query analysis tools that were released in the older administration applications SAP HANA studio and SAP HANA cockpit.

Related Information

Links to other documents

Analyzing SQL and Saving Plans (SAP HANA Cockpit)

SQL Analyzer Tool for SAP HANA (Product Page)

SQL Analyzer now available in Business Application Studio (SAP Community blog)

8.8 Memory Usage in the SAP HANA Database

Memory is a fundamental resource of the SAP HANA database. Understanding the types of memory available, how the SAP HANA database requests, uses, and manages this resource is crucial to the understanding of SAP HANA.

SAP HANA provides a variety of memory usage indicators that allow for monitoring, tracking, and alerting. The most important indicators are used memory and peak used memory. Since SAP HANA contains its own memory manager and memory pool, external indicators such as the size of resident memory at host level and the size of virtual and resident memory at process level can be misleading when you are estimating the real memory requirements of an SAP HANA deployment.

You can find detailed information about memory consumption of individual components and operations by looking into memory allocator statistics.

For more information about memory consumption with regards to SAP HANA licenses, see the topic on Memory Sizing and SAP Note 1704499 *System Measurement for License Audit*.

Related Information

Types of Memory: Overview [page 169]

SAP HANA Used Memory [page 171]

Memory Sizing [page 177]

Allocated Memory Pools and Allocation Limits [page 178]
SAP HANA Memory Usage and the Operating System [page 173]
Workload Management [page 415]
Memory Allocator Statistics [page 176]
SAP Note 1704499

8.8.1 Types of Memory: Overview

This topic gives an introduction to the types of memory and the terminology used. Further details are provided in the topics which follow.

The size of a HANA system can be measured by the amount of **physical memory** installed on the host. This can range from 256 gigabytes (GB) on on-premise systems (although cloud systems can be much smaller) upwards to several terabytes (TB) and it is required to run the Linux operating system, SAP HANA, and all other programs.

In addition to conventional DRAM memory, non-volatile memory (NVRAM) is supported in SAP HANA for DAX enabled (Direct Access) file systems, referred to in SAP HANA as **persistent memory**. Persistent memory combines the qualities of both DRAM and persistent Flash storage. It is byte addressable like DRAM, it can be treated by the CPU as RAM and offers a key benefit of significantly faster startup times. For licensing purposes, the same sizing rules apply to PMEM as to a SAP HANA system with only DRAM. The main memory utilization of a SAP HANA database is defined as the main memory currently consumed by all services on all non-standby nodes.

You can verify the licensed amount of main memory by checking the PRODUCT_LIMIT value in monitoring view M_LICENSE. The value of HOST_ACTIVE in the monitoring view M_LANDSCAPE_HOST_CONFIGURATION indicates if a node is active or in a standby role.

Persistent Memory

Although the performance of Persistent Memory (PMEM) is slower than DRAM it can be used to supplement the memory capacity and capabilities of the HANA System and therefore give higher memory capacities per server.

Because of the higher latency of PMEM it is used in SAP HANA specifically for MAIN data storage of the column store where approximately 90% of the data of a table is held. Because the data stored in MAIN persists it does not need to be reloaded during a system restart. Data in DELTA storage, where the fastest possible performance is required, continues to be stored in DRAM.

When sizing your system, the proportion of PMEM required can be judged by assessing the amount of column store main storage data required for the system, however, the usage of PMEM can be applied selectively, for specific tables or for specific columns of a table.

Note that it may be possible to mitigate the impact of the slower performance in comparison to DRAM by the use of features such as hardware prefetchers.

See topic 'Persistent Memory' for implementation details and SAP Notes 2700084 *FAQ: SAP HANA Persistent Memory* and 2786237 *Sizing SAP HANA with Persistent Memory*.

You can monitor persistent memory usage in, for example, system view M_CS_TABLES – values PERSISTENT_MEMORY and PERSISTENT_MEMORY_SIZE_IN _TOTAL, see also the Monitoring Views section of the 'Persistent Memory' topic.

Linux Memory Manager: Virtual Memory and Resident Memory

The Linux operating system reserves memory for each running program - referred to as its **virtual memory** which grows when the process requests more memory from the operating system, and shrinks when the process relinquishes unused memory. When part of the virtually allocated memory needs to be used, it is loaded or mapped to the real, physical memory of the host and becomes **resident memory**.

This mapping process is handled by Linux and if necessary, to avoid out of memory situations, the operating system may swap out some of a process's resident memory according to a least-recently-used algorithm to make room for other code or data. Swapping is normally not necessary in a properly sized SAP HANA appliance.

SAP HANA Memory Manager: Allocation Limits and Used Memory

The amount of DRAM memory reserved for SAP HANA is determined by a predefined `global allocation limit` which provides a pool of available free memory managed by SAP HANA. When more memory is required, such as for table growth or temporary computations, the SAP HANA memory manager obtains it from the pool. The global allocation limit must be consistent with the system's license: by default it is calculated as 90% of the first 64 GB of available physical memory on the host, plus 97% of each further GB.

The term **used memory** refers to the portion of the virtual memory that is in current use for code, stacks, tables and intermediate results. While the physical memory and allocation limit remain stable over time, memory usage constantly fluctuates as the SAP HANA memory manager dynamically allocates memory when required. Allocated memory is referred to as **heap memory** which remains allocated until no longer required when it is freed. You can monitor the current level of used memory by querying the view M_HEAP_MEMORY.

Memory is also allocated as **shared memory** to support cacheing of essential data required by both the name server and the index server(s). The allocation limit for individual services (such as the indexserver) can be set using the configuration parameter `allocationlimit` in the corresponding <service>.ini file. Details of shared memory are available in the monitoring view M_SHARED_MEMORY, such as the values for EFFECTIVE_ALLOCATION_LIMIT and TOTAL_MEMORY_USED_SIZE.

For further details refer to the topic on Allocator Statistics which provide detailed information about memory consumption.

Related Information

Persistent Memory [page 186]
Memory Allocator Statistics [page 176]
M_LICENSE System View

M_LANDSCAPE_HOST_CONFIGURATION System View
M_CS_TABLES System View
M_HEAP_MEMORY System View
SAP Note 2700084
SAP Note 2786237

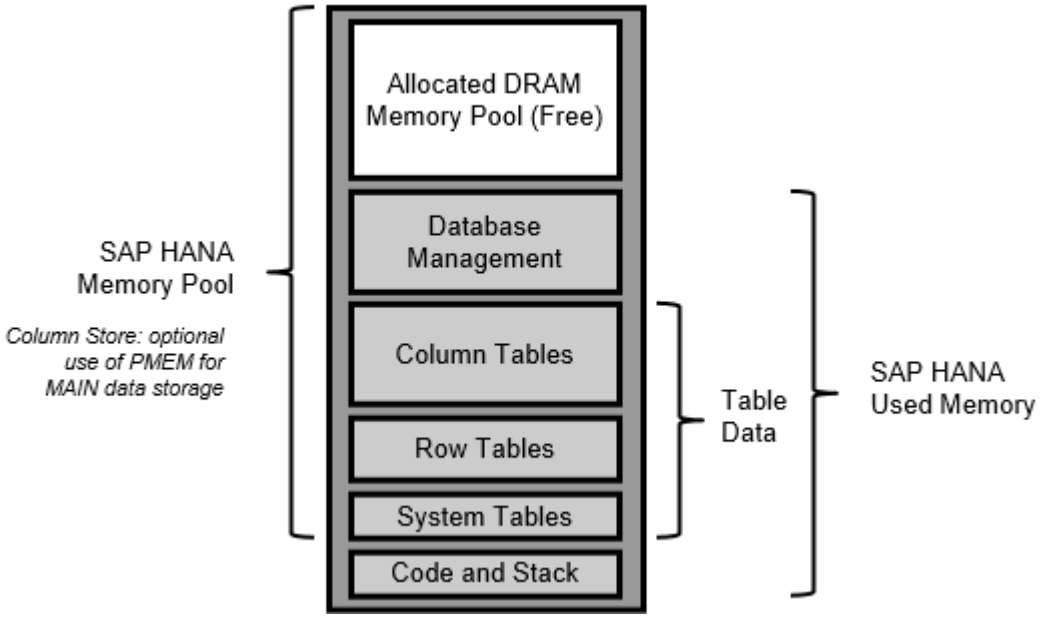
8.8.2 SAP HANA Used Memory

The total amount of memory used by SAP HANA is referred to as used memory. It includes program code and stack, all data and system tables, and the memory required for temporary computations.

SAP HANA consists of a number of processes running in the Linux operating environment. Under Linux, the operating system (OS) is responsible for reserving memory for all processes. When SAP HANA starts up, the OS reserves memory for the program code, the program stack, and static data. It then dynamically reserves additional data memory when requested by the SAP HANA memory manager. Dynamically allocated memory consists of heap memory and shared memory.

The following figure shows used memory, consisting of code, stack, and table data:

SAP HANA Used Memory



Since the code and program stack size are about 6 GB, almost all of used memory is used for table storage, computations, and database management.

Service Used Memory

An SAP HANA system consists of multiple services that all consume memory, in particular the `indexserver` service, the main database service. The index server holds all the data tables and temporary results, and therefore dominates SAP HANA used memory.

Peak Used Memory

Ultimately, it's more important to understand the behavior of used memory over time and under peak loads. For this purpose, SAP HANA has a special used memory indicator called peak used memory. As the value for used memory is a current measurement, peak used memory allows you to track the maximum value for used memory over time.

You can monitor the current level of used memory by querying the monitoring view M_HEAP_MEMORY which shows various details of memory allocator statistics. This has a corresponding resettable view (M_HEAP_MEMORY_RESET) which shows only values since the last reset. You can use this for example, to test the impact of a certain workload on memory usage, reset the view:

```
ALTER SYSTEM RESET MONITORING VIEW "SYS"."M_HEAP_MEMORY_RESET";
```

You can then run the workload and then examine the new peak used memory value. The RESET_TIME value indicates the last time the data was reset.

Memory Usage of Tables

The dominant part of the used memory in the SAP HANA database is the space used by data tables. Separate measurements are available for column-store tables and row-store tables. If persistent memory (PMEM) is in use to supplement the capacity and capabilities of the HANA System, it is effective here for persistent storage of MAIN data.

> ⓘ Note
>
> The SAP HANA database loads column-store tables into memory column by column only upon use, which is sometimes called "lazy loading". This method means that columns that are unused aren't loaded and memory waste is avoided. When the SAP HANA database runs out of allocatable memory, it tries to free up some memory by unloading unimportant data (such as caches) and even table columns that haven't been used recently. Therefore, if it's important to measure precisely the total, or worst-case, amount of memory used for a particular table, it's important to ensure that you fully load the table into memory.

Memory Usage of Expensive Statements

Every query and statement consumes memory, for the evaluation of the statement plan, caching, and, mainly the calculation of intermediate and final results. While many statement executions use only a moderate amount of memory, some queries, for instance a query using unfiltered cross joins, places a heavy load on even large systems.

Expensive statements are individual SQL statements whose execution time exceeded a configured threshold. The expensive statements trace records information about these statements for further analysis. If in addition to activating the expensive statements trace, you enable per-statement memory tracking, the expensive statements trace also shows the peak memory size used to execute expensive statements.

Workload management features are available to further protect an SAP HANA system against excessive memory usage due to uncontrolled queries, for example, by limiting the amount of memory used by single statement executions per host - see 'Setting a Memory Limit for SQL Statements'.

Related Information

Setting a Memory Limit for SQL Statements [page 431]
Load/Unload a Column Table into/from Memory [page 294]
M_HEAP_MEMORY System View (SAP HANA SQL Reference Guide)
Links to SAP HANA Cockpit Guide
The Performance Monitor
Monitor Table Usage
Monitor and Analyze Active Statements
Monitor and Analyze Expensive Statements

8.8.3 SAP HANA Memory Usage and the Operating System

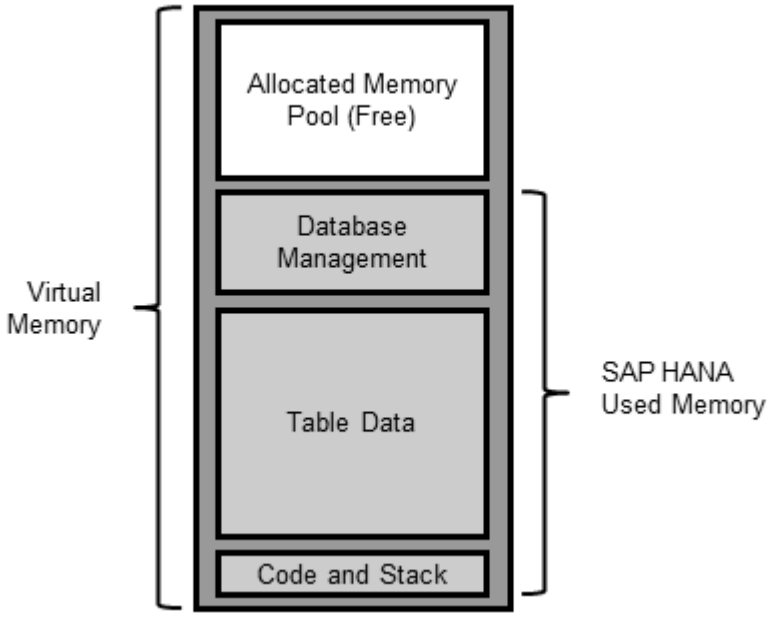
Due to the way in which SAP HANA manages memory, the relationship between Linux memory indicators and SAP HANA's own memory indicators may not correlate as expected.

From the perspective of the Linux operating system, SAP HANA is a collection of separate processes. Linux programs reserve memory for their use from the Linux operating system. The entire reserved memory footprint of a program is referred to as its virtual memory. Each Linux process has its own virtual memory, which grows when the process requests more memory from the operating system, and shrinks when the process relinquishes unused memory.

Virtual, Physical, and Resident Memory

You can think of virtual memory size as the memory amount that the process has requested (or allocated) from the operating system, including reservations for its code, stack, data, and memory pools under program control. SAP HANA's virtual memory is logically shown in the following figure. Although SAP HANA really consists of several separate processes, the figure shows all SAP HANA processes combined:

SAP HANA Virtual Memory

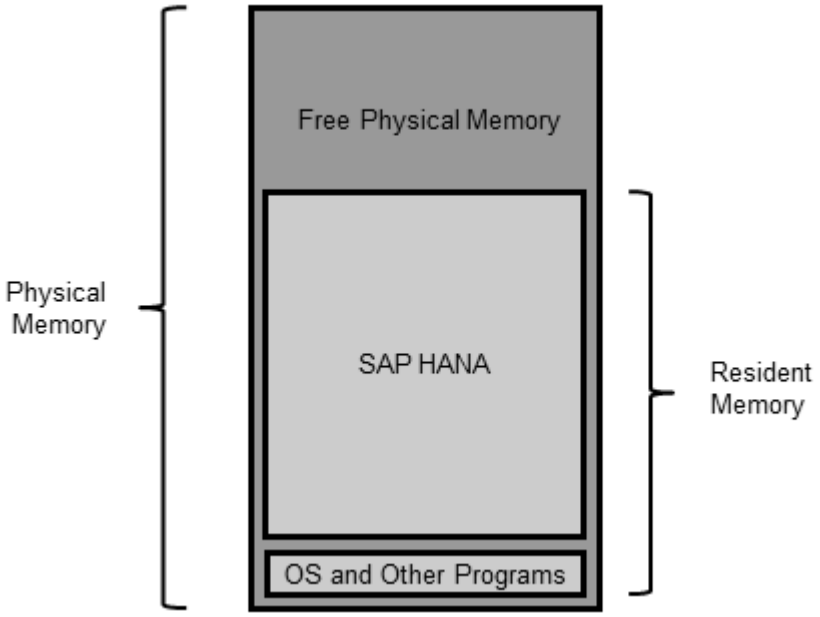


When part of the virtually allocated memory actually needs to be used, it's loaded or mapped to the real, physical memory of the host and becomes resident. Physical memory is the DRAM memory installed on the host. On most SAP HANA hosts, it ranges from 256 gigabytes (GB) to 1 terabyte (TB). It's used to run the Linux operating system, SAP HANA, and all other programs.

Resident memory is the physical memory actually in operational use by a process. Over time, the operating system may swap out some of a process's resident memory according to a least-recently-used algorithm to make room for other code or data. Thus, a process's resident memory size may fluctuate independently of its virtual memory size. In a properly sized SAP HANA appliance, there's enough physical memory so that swapping is disabled and isn't observed.

This memory usage can be illustrated as follows:

SAP HANA Resident Memory



On a typical SAP HANA appliance, the resident memory part of the operating system and all other running programs usually doesn't exceed 2 GB. The rest of the memory is therefore dedicated for the use of SAP HANA.

When memory is required for table growth or for temporary computations, the SAP HANA code obtains it from the existing memory pool. When the pool can't satisfy the request, the SAP HANA memory manager requests and reserves more memory from the operating system. At this point, the virtual memory size of SAP HANA processes grows.

Once a temporary computation completes or a table is dropped, the freed memory is returned to the memory manager, which recycles it to its pool without informing the operating system. Therefore, from SAP HANA's perspective, the amount of used memory shrinks, but the process's virtual and resident memory sizes aren't affected. This creates a situation where the used memory value can shrink to below the size of SAP HANA's resident memory, which is normal.

> **Note**
>
> The memory manager can also choose to return memory back to the operating system, for example when the pool is close to the allocation limit and contains large unused parts.

Related Information

SAP HANA Used Memory [page 171]
Memory Sizing [page 177]
Allocated Memory Pools and Allocation Limits [page 178]

8.8.4 Memory Allocator Statistics

Detailed information about memory consumption can be found by looking into allocator statistics.

Allocator statistics track the memory consumption of individual components and operations in SAP HANA and may help you to analyze issues related to memory consumption. Statistics are saved in the system views M_HEAP_MEMORY (allocated memory by component) and M_CONTEXT_MEMORY (allocated memory that can be associated with a connection, a statement, or a user). Both views have a reset feature so that statistics can be captured for a specific period of time. The embedded statistics service also includes a view which tracks memory allocation per host: HOST_HEAP_ALLOCATORS.

Refer to the *SAP HANA SQL and System Views Reference* for full details of these views.

Allocator statistics are saved automatically for each core processor and in certain scenarios where systems have a large number of logical cores the statistics can consume a significant amount of memory. To save memory, statistics logging can be reduced to save statistics only for each node or only for each statistics object. An example of using the `lscpu` command to retrieve details of the physical and logical CPU architecture is given in the section *Controlling CPU Consumption*.

You can configure this feature by setting values for the following two configuration parameters in the global.ini file:

- The parameter `pool_statistics_striping` can reduce the amount of memory consumed by the component-specific allocator statistics (rows in M_HEAP_MEMORY with category *PoolAllocator*).
- The parameter `composite_statistics_striping` can reduce the amount of memory consumed by statement-specific allocator statistics (rows in M_CONTEXT_MEMORY).

The parameters can be set to one of the following values, the configuration can be changed online, but the change will only affect newly created statistic objects:

Value	Effect
auto (default value)	Let the system decide the statistics strategy. By default HANA will try to utilize as much memory as possible for maximum performance.
core	The system allocates one stripe per logical core.
numa	The system allocates only one stripe per NUMA node.
none	In this case, the system creates a single stripe per statistics object.

Related Information

SAP HANA SQL Reference Guide for SAP HANA Platform

Controlling CPU Consumption [page 421]

8.8.5 Memory Sizing

Memory sizing is the process of estimating in advance the amount of memory that is required to run a certain workload on an SAP HANA database. To understand memory sizing, several questions need to be answered.

- What is the size of the data tables that must be stored in the SAP HANA database?
  You may be able to estimate this based on the size of your existing data, but unless you precisely know the compression ratio of the existing data and the anticipated growth factor, this estimate may not be accurate.
- What is the expected compression ratio that SAP HANA applies to these tables?
  The column store of the SAP HANA database automatically uses a combination of various advanced compression algorithms (dictionary, RLE, sparse, and so on) to compress each table column separately. The achieved compression ratio depends on many factors, such as the nature of the data, its organization and data types, the presence of repeated values, the number of indexes (SAP HANA requires fewer indexes), and so on.
- How much extra working memory is required for temporary computations?
  The amount of extra memory depends on the size of the tables (larger tables create larger intermediate result tables in operations such as joins), but even more on the expected workload in terms of the concurrency and complexity of analytical queries (each concurrent query needs its own workspace).
- Can persistent memory be used?
  In addition to conventional DRAM memory, non-volatile memory (NVRAM) is supported in SAP HANA for DAX enabled (Direct Access) file systems. Because of the higher latency of PMEM it is used in SAP HANA specifically for MAIN data storage of the column store where approximately 90% of the data of a table is held. It can be used selectively, for specific tables or for specific columns of a table.
- Is additional memory required for the NSE buffer cache?
  If you are using the Native Storage Extension to handle warm data, a buffer cache is used to hold recently-used data which has been loaded from disk. By default 10% of the HANA memory is used for the buffer cache, but configuration parameters are available to manage the size depending on the size of your warm data set. Typically, the ratio of the buffer cache to the warm data on disk should not be smaller than 1 to 8 (1 GB of memory for the buffer cache for 8 GB of warm data storage). Refer to 'Buffer Cache Sizing Guidelines' for more information and details of the configuration options available.

The following SAP Notes provide additional tools and information to help you size the required amount of memory:

- SAP Note 1514966 - SAP HANA 1.0: Sizing SAP In-Memory Database
- SAP Note 1637145 - SAP BW on HANA: Sizing SAP In-Memory Database
- SAP Note 2296290 - New Sizing Report for BW on HANA
- SAP Note 2786237 - Sizing SAP HANA with Persistent Memory

However, the most accurate method is to import several representative tables into an SAP HANA system, measure the memory requirements, and extrapolate from the results.

Related Information

Buffer Cache Sizing Guidelines [page 219]
SAP Note 1514966
SAP Note 1637145

SAP Note 2296290
SAP Note 2786237

8.8.6 Allocated Memory Pools and Allocation Limits

SAP HANA, across its different processes, reserves a pool of memory before actual use. This pool of allocated memory is preallocated from the operating system over time, up to a predefined global allocation limit, and is then efficiently used by SAP HANA as needed.

SAP HANA preallocates and manages its own memory pool, used for storing in-memory table data, thread stacks, temporary results, and other system data structures. When more memory is required for table growth or temporary computations, the SAP HANA memory manager obtains it from the pool. When the pool can't satisfy the request, the memory manager increases the pool size by requesting more memory from the operating system, up to a predefined allocation limit.

By default, the allocation limit is calculated as follows: 90% of the first 64 GB of available physical memory on the host, plus 97% of each further GB.

The limit can be changed by modifying the `global_allocation_limit` configuration parameter in the global.ini file. It can be defined either as a fixed value in MB or as a flexible percentage of the available main memory size. If you enter a percentage value, the precise value of the limit is calculated automatically by the system. Moreover, if you then change the size of the container where the system runs the allocation limit automatically adjusts to the correct percentage of the new container size.

There's normally no reason to change the value of this parameter. However, on development systems with more than one SAP HANA system installed on a single host you could limit the size of the memory pool to avoid resource contentions or conflicts.

A change may also be necessary to remain in compliance with the memory allowance of your license if you purchased a license for less than the total amount of physical memory available. This change is illustrated in the following examples:

> Example
>
> - You have a server with 512 GB, but purchased an SAP HANA license for only 384 GB. You therefore set the `global_allocation_limit` to 393216 (384 * 1024 MB).
> - You have a distributed HANA system on four hosts with 512 GB each, but purchased an SAP HANA license for only 768 GB. Set the `global_allocation_limit` to 196608 (192 * 1024 MB on each host).

Service Allocation Limit

In addition to the global allocation limit, each service running on the host has an allocation limit, the service allocation limit. Given that collectively, all services can't consume more memory than the global allocation limit, each service has what is called an effective allocation limit. The effective allocation limit of a service specifies how much physical memory a service can in reality consume given the current memory consumption of other services.

> **Example**
>
> A single-host system has 100-GB physical memory. Both the global allocation limit and the individual service allocation limits are 92.5% (default values). Therefore:
>
> - Collectively, all services of the SAP HANA database can use a maximum of 92.5 GB.
> - Individually, each service can use a maximum of 92.5 GB.
>
> Therefore, if 2 services are running and the current memory pool of service 1 is 50 GB, then the effective allocation limit of service 2 is 42.5 GB. This allocation happens because service 1 is already using 50 GB and together they can't exceed the global allocation limit of 92.5 GB.

What happens when the allocation limit is reached?

Memory is a finite resource. Once the allocation limit has been reached and the pool is exhausted, the memory manager only allocates memory for internal operations after first giving up something else. Buffers and caches are released, and column store tables are unloaded, column by column, based on a least-recently-used order, up to a preset lower limit. When tables are partitioned over several hosts, it's managed on a host-by-host basis; that is, column partitions are unloaded only on hosts with an acute memory shortage.

Avoid table (column or partition) unloading since it leads to performance degradation later when the table is queried and the data has to be reloaded. You can identify pool exhaustion by examining the `M_CS_UNLOADS` system view.

However, it's still possible that the memory manager needs more memory than is available leading to an out-of-memory failure. This may happen, for example, when too many concurrent transactions use up all memory, or when a complex query performs a cross join on large tables and creates a huge intermediate result that exceeds the available memory.

Related Information

Change the Global Memory Allocation Limit

8.8.7 Change the Global Memory Allocation Limit

SAP HANA preallocates a pool of memory from the operating system over time, up to a predefined global allocation limit. You can change the default global allocation limit.

Prerequisites

You have the system privilege INIFILE ADMIN.

Context

The `global_allocation_limit` parameter is used to limit the amount of memory that can be used by SAP HANA. The unit for this parameter is MB. The default value is 0 in which case the global allocation limit is calculated as follows: 90% of the first 64 GB of available physical memory on the host plus 97% of each further GB. Or, in the case of systems with small physical memory (4GB or less) the limit is set to the smaller value of either 3GB or 90% of available physical memory. Alternatively, you can define this limit as a flexible percentage of the available main memory size. If you enter a percentage value the precise value of the limit will be calculated automatically by the system. Moreover, if you then change the size of the container where the system runs the allocation limit will automatically adjust to the correct percentage of the new container size.

Changing this parameter does not require a restart.

Procedure

In the `global.ini` configuration file change the value of the `global_allocation_limit` in the `memorymanager` section.

You can enter a value for the entire system and, for distributed multiple-host systems, for individual hosts. If you enter only a value for the system, it is used for all hosts. For example, if you have 5 hosts and you set the limit to 5 GB, the system can use up to 5 GB on each host (25 GB in total). If you enter a value for a specific host, then for that host, the specific value is used and the system value is only used for all other hosts.

Refer to the *Workload Management* section for details of other options for managing memory including setting a statement limit and admission control.

Related Information

Allocated Memory Pools and Allocation Limits [page 178]
Memory and CPU Usage for Tenant Databases [page 113]
Setting a Memory Limit for SQL Statements [page 431]
Managing Peak Load (Admission Control) [page 434]
SAP Note 1999997

8.9 Persistent Data Storage in the SAP HANA Database

Persistent storage media are required for ongoing save operations for data and redo log files.

To protect data from the risk of memory failure SAP HANA persists in-memory data to storage media and flushes all changed data from memory to the data volumes. This operation takes place on the basis of savepoints which occur by default every 5 minutes.

Customers have a choice of storage media types: in addition to conventional hard disk storage, non-volatile RAM is also supported for data storage, specifically, for the MAIN fragment of column store tables. For more information about the MAIN and DELTA fragments of tables refer to Delta Merge.

The following subsections describe how these types of persistence work and how they must be configured for use. The two storage types operate in a similar way but there are some essential differences and a separate section is dedicated to each type.

Related Information

The Delta Merge Operation [page 301]

8.9.1 Data and Log Volumes

To ensure that the database can always be restored to its most recent committed state, changes to data in the database are periodically copied to disk, logs containing data changes and certain transaction events are also saved regularly to disk. Data and logs of a system are stored in volumes.

SAP HANA persists in-memory data by using savepoints. Each SAP HANA service has its own separate savepoints. During a savepoint operation, the SAP HANA database flushes all changed data from memory to the data volumes. The data belonging to a savepoint represents a consistent state of the data on disk and remains so until the next savepoint operation has completed. Redo log entries are written to the log volumes for all changes to persistent data. In the event of a database restart (for example, after a crash), the data from the last completed savepoint can be read from the data volumes, and the redo log entries written to the log volumes since the last savepoint can be replayed.

The frequency at which savepoints are defined can be configured in the `persistence` section of the `global.ini` file (every 5 minutes by default). Savepoints are also triggered automatically by a number of other operations such as data backup, and database shutdown and restart. You can trigger a savepoint manually by executing the following statement `ALTER SYSTEM SAVEPOINT`.

You must always ensure that there is enough space on the disk to save data and logs. Otherwise, a disk-full event will occur and the database will stop working.

Directory Hierarchy for Data and Log Storage

During the installation process, the following default directories are created as the storage locations for data and log volumes:

- `/usr/sap/<SID>/SYS/global/hdb/data`
- `/usr/sap/<SID>/SYS/global/hdb/log`

> **Note**
>
> These default directories are defined in the parameters `basepath_datavolumes` and `basepath_logvolmes` in the `persistence` section of the `global.ini` file.

These directories contain a series of sub-directories to store data for each of the following:

Host	Named mnt00001, mnt00002, mnt00003... by default.
Tenant database	"DB_" + <database name>
Database service	Named hdb00001, hdb00002, hdb00003... by default.
Replica volume	A separate persistence for each replica volume; the subdirectory is the numerical Replica ID value.

Example of a data subdirectory for a tenant database named "M90" (see also the following illustration):

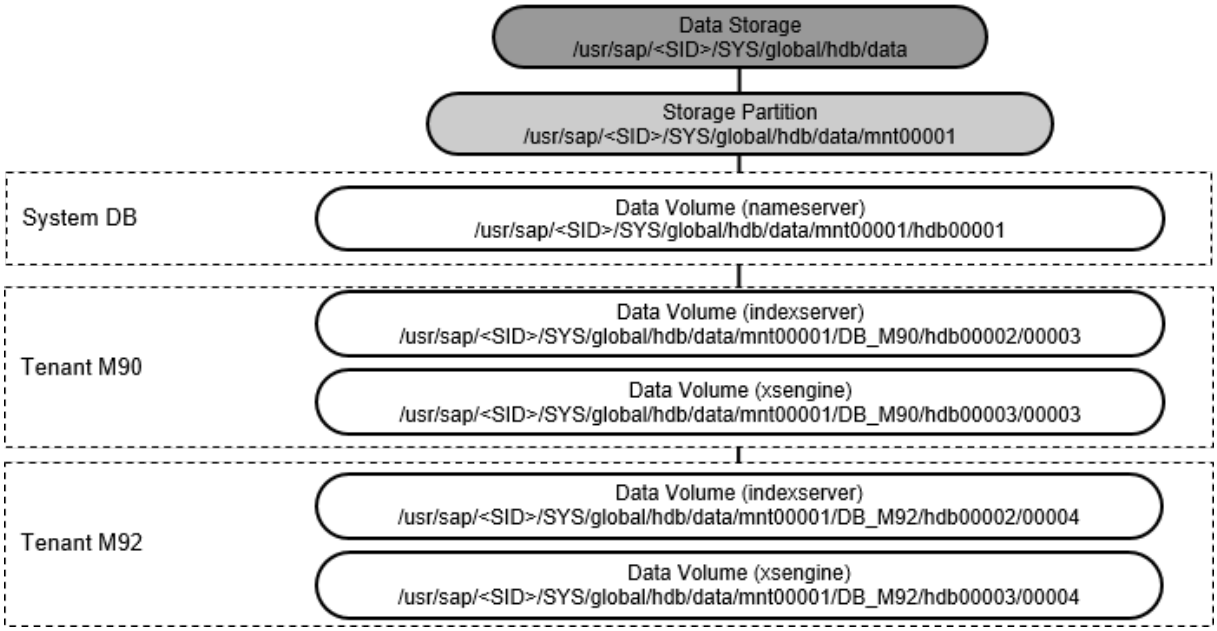
```
/usr/sap/DB1/SYS/global/hdb/data/mnt00001/DB_M90/hdb00002.00001/
```

The services that persist data and therefore have volumes are the following:

Service	Note
nameserver	Only the nameserver service in the system database on the active master host persists data. Slave namesever hosts communicate with the master, but do not persist data.
indexserver	The indexserver service in tenant databases on all hosts except standby hosts persists data. This data volume can be partitioned to improve performance when reading and writing to this server.
xsengine (if running as a separate service)	The xsengine service persists data on any host on which it is running.

The following figure illustrates the default storage hierarchy described above using the example of data storage. The system is a replication system with two tenant databases on a single host.

Directory Hierarchy for Persistent Data Storage



Data and Log Volumes

Data Volumes

By default each data volume contains one file (`datavolume_0000.dat`) in which data is organized into pages, ranging in size from 4KB to 16MB (page size class). Data is written to and loaded from the data volume page-wise. Over time, pages are created, changed, overwritten, and deleted. The size of the data file is automatically increased as more space is required. However, it is not automatically decreased when less space is required. This means that at any given time, the actual payload of a data volume (that is the cumulative size of the pages currently in use) may be less than its total size. This is not necessarily significant – it simply means that the amount of data in the file is currently less than at some point in the past (for example, after a large data load). The `Total Size` and the `Fill Ratio` values in the monitoring view M_DATA_VOLUME_STATISTICS give details about capacity.

If a data volume has a considerable amount of free space, it might be appropriate to shrink the data volume. However, a data file that is excessively large for its typical payload can also indicate a more serious problem with the database. SAP support can help you analyze your situation.

The indexserver data volume can be partitioned so that data can be segmented and distributed over several *stripes* on the disk to improve performance. SQL commands are available to create and drop partitions, a numeric partition ID value is assigned automatically by the system. In this case, the volume ID and an offset value to identify the stripe where the data is saved are included in the volume file name (`datavolume_nnnn_<offset>.dat`). See *Partitioning Data Volumes* for details.

Log Volumes

Each log volume contains the file `logsegment_<partition_ID>_directory.dat` and one or more log segment files (`logsegment_<partition_ID>_<segment_number>.dat`). Currently only one log partition is supported for each service, so the default file names are `logsegment_000_directory.dat` and `logsegment_000_00000000.dat`, `logsegment_000_00000001.dat`, `logsegment_000_00000002.dat` and so on. Log segment files are cyclically overwritten depending on the log mode. The log mode determines how logs are backed up. Log volumes only grow if there are no more segment files available for overwriting. Log segment files that are available for overwriting have been backed up and are not required for a database restart. If necessary you can remove these files to free up space in your file system by executing the SQL statement `ALTER SYSTEM RECLAIM LOG`. Note that new log segment files will need to be created later and this will affect performance.

You can monitor disk usage in SAP HANA cockpit using the Disk Usage and Performance Monitor apps. See *Monitoring Disk Space* and *Monitoring in SAP HANA Cockpit* for details.

> ⚠ Caution
>
> Do not remove either data files or log files using operating system tools as this will corrupt the database.

Multiple Files in Data Volumes for Ext3 File Systems

The Ext3 file system has a file size limitation of 2TB. If the existing files in a data volume located in an Ext3 file system reach the 2 TB limit, SAP HANA automatically creates additional files. This allows the use of Ext3 file systems even with applications (in particular, single-host SAP ERP systems) that have a larger memory requirement per host.

For more information about splitting data backups into files of limited size, see the section on data backups.

Related Information

Data Backups [page 942]
ALTER SYSTEM SAVEPOINT Statement (System Management)
ALTER SYSTEM RECLAIM LOG Statement (System Management)
Partitioning Data Volumes [page 184]

8.9.2 Partitioning Data Volumes

Data volumes on the indexserver can be partitioned so that read and write operations can run in parallel with increased data throughput. Additionally, you can partition data volumes on remote system replication hosts.

Creating Partitions

Each SAP HANA service has one data volume which by default stores its data in one single file. Administrators have the option to partition the data volume for the indexserver so that data files can be segmented and stored in different locations and can then be accessed in parallel threads. In the case of Network File Systems data can also be written in parallel across multiple connections. Partitioning data volumes in this way will therefore speed up all read/write operations on the volume including savepoints, merges, restart, table loading operations, and backups.

> ⓘ Note
>
> The problem of excessively large file sizes in relation to large data volumes is overcome by striping. Some file systems may have limits for the maximum file size and the persistency layer automatically chooses appropriate striping sizes for these file systems. You can define a maximum file size for stripes using the striping configuration parameters in the persistence section of the indexserver.ini file: `datavolume_striping` (TRUE / FALSE), and `datavolume_striping_size_gb` (a value between 100 and 2000GB).

If you create a new partition on the index server it is added simultaneously to all indexservers in the topology. New partitions become active after the next savepoint on each indexserver, this is shown in the partition STATE value which changes from ACTIVATING to ACTIVE. By default all data volumes have a single partition with the numeric ID zero. A numeric partition ID is assigned automatically to new partitions by the HANA persistency layer. If the partition numbering is for any reason inconsistent across all indexservers then any attempt to add new partitions will fail.

By default the basepath for data volume partitions is `/usr/sap/<SID>/SYS/global/hdb/data`, however, you can also add partitions in a specified path of your own choice (see examples below using the ADD PARTITION PATH syntax). The path must be reachable by all nodes or services in the topology. Beneath the specified path the standard folder structure is created automatically with a numbered folder for each

host. A user-defined path is required, for example, if you are using multiple NFS connections so that data can be written in parallel to different paths. This option must be explicitly enabled by setting the `PERSISTENCE_DATAVOLUME_PARTITION_MULTIPATH` parameter in the `customizable_functionalities` section of global.ini to TRUE.

The partition basepath is saved in the indexserver.ini configuration file in the `basepath_datavolume` key of the `partition ID` section.

Example SQL Statements

Details of the SQL statements to manage data volume partitions are given in the *SAP HANA SQL and System Views Reference*, the two main statements to add and drop partitions are shown here. If you execute the command to add a partition on the indexserver it will add a new partition to all indexservers in the topology:

```
ALTER SYSTEM ALTER DATAVOLUME ADD PARTITION
```

You can drop specific data volume partitions by specifying the partition ID number:

```
ALTER SYSTEM ALTER DATAVOLUME DROP PARTITION <id>
```

This command drops the identified partition from all indexservers in the topology. The default partition with ID zero cannot be dropped. If you drop a partition then all data stored in the partition is automatically moved to the remaining partitions and for this reason dropping a partition may take time. This operation also removes the partition entry from the configuration file.

Data Volumes for System Replication

In a system replication scenario you can specify the id of a host used for system replication so that you can replicate data to a specific partition. The following examples show adding and dropping a replication site by using the SYSTEM REPLICATION SITE syntax:

```
ALTER SYSTEM ALTER DATAVOLUME ADD PARTITION [PATH '<path>'] [SYSTEM REPLICATION
SITE <site>]
```

```
ALTER SYSTEM ALTER DATAVOLUME DROP PARTITION <id> [SYSTEM REPLICATION SITE
<site>]
```

The value for <site> can be specified as either the site name as a string value or the numeric SAP HANA system replication site id value.

Note that this SQL statement to create data volume partitions on a secondary site only works if the secondary has already been setup. Therefore the secondary is initially created without additional partitions and during initial data shipping all the data pages are stored within the same partition. When new partitions are created later on with the statement given above, changed data pages will be distributed among newly created partitions, but it will take some time until all partitions are used to the same extent.

To avoid this problem the partitions could be configured on the secondary site before the system is newly registered as secondary and the initial data shipping is executed. There are several options depending on the scenario:

- Newly installed HANA system (not yet configured as secondary). In this case, you can configure data volume partitions using SQL before HSR is configured, then register as secondary afterwards.
- System is configured as secondary, still attached to primary. In this case, you can configure data volume partitions using the SQL statement from the primary site then re-register as secondary afterwards.

- System is configured as secondary, not attached anymore to primary. In this case, bring the system online using takeover or unregister, use the SQL statement to configure data volume partitions then re-register as secondary afterwards.

Note also that in a running system replication setup, you may not be able to drop an active data volume partition as system replication uses data volume snapshot technology. You may see the following error: *Cannot move page inside/out of DataVolume*. In this case it may be necessary to disable system replication functionality, drop the partition, and then setup system replication again.

Monitoring Views

You can see the current data volume configuration from the following two views:

- M_DATA_VOLUME_STATISTICS: This provides statistics for the data volume partitions on the indexserver including the number of partitions and size.
- M_DATA_VOLUME_PARTITION_STATISTICS: This view provides statistics for the individual partitions, identified by PARTITION_ID, and includes the partition STATE value.

In a replication scenario you can monitor the M_DATA_VOLUME_PARTITION_STATISTICS view on the secondary via proxy schema SYS_SR_SITE<siteName> (where <siteName> is the name of the secondary site).

8.9.3 Persistent Memory

Persistent memory (non-volatile RAM, also referred to as Intel® Optane™ Storage Class Memory) is supported in SAP HANA as a persistent storage type.

Persistent memory (or NVRAM) is an emerging class of memory that combines the qualities of both DRAM and Flash storage and bridges the gap between disk storage and main memory. For the purposes of SAP HANA, its most important characteristics are that it's byte addressable like DRAM and can be treated by the CPU as RAM, offering fast read and write performance; its latency characteristics are also very close to DRAM.

The latest information about this feature, such as hardware availability and operating system support, is provided in SAP Note 2618154 *SAP HANA Persistent Memory (NVM)*.

The persistent storage functions as DRAM memory and is used specifically for MAIN data fragments of the column store where approximately 90% of the data of a table is held. In this way, DELTA data can continue to be stored in DRAM (a relatively small quantity of data requiring the fastest possible access) with the MAIN segment of a table in NVRAM. Note that if persistent memory storage is enabled, data is still written to the data volumes but is ignored. After a system restart, the main data fragments saved in persistent storage are still available in memory and don't have to be reloaded. This storage type has no impact on backup and recovery or high availability processes.

As with the data volume persistent storage feature, data is persisted on the basis of savepoints. In the case of non-volatile memory storage, NVM blocks are constructed whenever the in-memory MAIN fragment is created and are kept consistent according to HANA's savepoints.

The key performance benefit is in greatly accelerated start-up times so that data can be quickly reloaded into memory and any system downtime can be minimized. Other benefits can include higher memory capacities per server.

Performance differences of this storage type compared to DRAM due to higher latency are mitigated by features such as hardware prefetchers. It's also possible to enable persistent memory storage for selected tables or for specific columns of a table so that a selective implementation of this storage type is possible (see the Configuration section).

Installation

The persistent memory feature can be set up during system installation using the SAP HANA database lifecycle manager (HDBLCM). The lifecycle manager uses two parameters to enable persistent memory (`use_pmem`) and the `pmempath` parameter to set the basepath (see the Configuration section). For more information on lifecycle manager, refer to the *SAP HANA Server Installation and Update Guide*.

At the level of the Linux operating system, SAP HANA only supports DAX enabled (Direct Access) file systems.

Filesystem layout of persistent memory block storage

The directory specified as the basepath (see Configuration) is the root persistent memory storage directory beneath which data for each savepoint is stored. The structure beneath the basepath is maintained by HANA and no attempt should be made to modify this content. Multiple logical versions of blocks are saved and are organized in subdirectories named on the basis of, for example, service name, Volume ID and current savepoint version. There are two subdirectory systems, one for data and one for deletion requests: the memory storage persistence maintains a history of requests to delete blocks in a *tombstones* directory parallel to the *data* directory. In the case of tombstone blocks, the savepoint version relates to the savepoint when the data block was requested for deletion.

Configuration

The default behavior for using persistent memory is determined automatically: the system tests for the DAX filesystem at the defined persistent memory basepath. If the hardware is installed and the basepath correctly configured pointing to the persistent memory mount, then all tables use persistent memory by default. The SAP HANA installer sets the persistent memory basepath parameter if it's able to identify the DAX filesystem and if the user of the installation tool confirms the usage; the basepath parameter can also be set manually if necessary.

This default behavior can be overridden at four levels in the sequence shown here:

4	Database	Can be enabled or disabled by configuration parameter.
3	Table	Can be enabled or disabled by SQL statement.
2	Partition	Can be enabled or disabled by SQL statement.
1	Column	Can be enabled or disabled by SQL statement.

This level of granularity offers a high degree of flexibility: if persistent memory is applied at a certain level it's automatically inherited at lower levels of the sequence but can also be overridden at lower levels.

At the highest configuration level (database), persistent memory is managed by setting the `table_default` configuration parameter for the database as a whole. You can switch `table_default` to OFF to enable more selective persistent memory storage at lower levels for particular tables and partitions.

At other levels, persistent memory is managed at the SQL command line using the `alter_persistent_memory_spec` clause with the CREATE / ALTER table commands. Here, we give examples to show how it can be applied at each level; refer to the SAP HANA SQL Reference Guide for full details of this clause.

Configuration Parameter: Basepath

As with the other data and log persistent storage volumes and also the fast restart option (tmpfs), the storage location basepath for persistent memory storage must be defined as a configuration parameter in the `persistence` section of the global.ini file. Enter the basepath location in the following parameter:

Parameter	**basepath_persistent_memory_volumes**
Short Description	Location of NVRAM storage.
Full Description	Data for NVRAM-enabled tables or columns is stored at the location defined in this parameter. Multiple locations can be defined here using a semicolon as a separator (no spaces). Multiple paths correspond to NUMA nodes without any ordering restriction in terms of NUMA node mapping, for example, for a four-NUMA node system: `"/mnt/pmem0;/mnt/pmem1;/mnt/pmem4;/mnt/pmem2`
Type	Text
Change	Restart required (offline changeable only)
Default	Blank
Layer Restriction	Can only be set at the HOST or SYSTEM layers.

Configuration Parameter: Enable persistent memory storage

The `table_default` parameter is in the `persistent_memory` section of the indexserver.ini file.

Parameter	**table_default**
Short Description	Enable or disable persistent memory storage for the database generally.
Full Description	This parameter can be set to ON or OFF or DEFAULT (no explicit preference, follow the existing default behavior). The setting applies for the complete database but can be overridden by settings applied at a lower level.
Change	Online
Default	DEFAULT

SQL Alter Persistent Memory Clause

Persistent memory can be enabled by SQL command using the `alter_persistent_memory_spec` clause with CREATE and ALTER table. This command can be used at any of the three levels (table, partition, or column) to enable or disable persistent memory storage using the PERSISTENT MEMORY switch. The following examples illustrate this method using CREATE TABLE.

Example 1: Create table with persistent memory storage enabled for the new table.

```
CREATE COLUMN TABLE PMTABLE (C1 INT, C2 VARCHAR (10)) PERSISTENT MEMORY ON
```

PERSISTENT MEMORY can be set to ON, OFF, or DEFAULT meaning no preference.

Example 2: Create range-partitioned table with persistent memory storage for a selected partition (only valid for range partitions):

```
 CREATE COLUMN TABLE PMTABLE (C1 INT) PARTITION BY RANGE (C1) (
                   PARTITION '0' <= VALUES < '10' PERSISTENT MEMORY ON,
                   PARTITION OTHERS PERSISTENT MEMORY OFF);
```

Example 3: Create table with persistent memory storage applied to selected columns of a table:

```
CREATE COLUMN TABLE PMTABLE (C1 INT PERSISTENT MEMORY ON, C2 VARCHAR (10), C3
INT PERSISTENT MEMORY OFF)
```

For ALTER TABLE, a preference value is also required (in addition to the ON/OFF switch) to determine how the change is applied. This clause is the `alter persistent memory preference` clause that requires one (or more) of the following keywords:

- IMMEDIATE - the change is applied immediately and the specified storage is populated.
- DEFERRED - the specified storage will be populated at the time of the next delta merge or reload.
- CASCADE - this keyword can be used at table level in addition to IMMEDIATE and DEFERRED to apply the change to all lower levels of the hierarchy.

The following example shows this usage with ALTER TABLE and cascades the change immediately to partitions and columns of the named table:

```
ALTER TABLE MYTABLE PERSISTENT MEMORY ON IMMEDIATE CASCADE
```

Unloading Tables from Memory

The standard UNLOAD statement, and also table unloads, executed implicitly by HANA services due to memory running low, don't by default free the memory but simply remove the mapping of relevant data structures from the HANA service process address space and the MAIN data fragment remains; this applies to both persistent memory and the fast restart option (tmpfs). The configuration parameter `table_unload_action` in the [persistent_memory] section of the indexserver.ini file can be used to control this unload behavior. The default value of the parameter is RETAIN but this can be set to DELETE in which case MAIN data fragments are cleared from memory.

Regardless of the setting of this parameter, you can also use UNLOAD with the additional DELETE PERSISTENT MEMORY clause to free memory, for example:

```
UNLOAD FRAOTABLE DELETE PERSISTENT MEMORY;
```

Managing Disk Space Usage

To avoid performance issues that might occur due to lack of disk space you should limit the filesystem space allocation for PMEM devices. You can do this at the level of the operating system using XFS quotas to set a hard limit on the space usage of each PMEM device mount (see Linux manual page on xfs_quota for details) or, if the Linux/XFS features cannot be modified for any reason, using a configuration parameter from within SAP HANA.

The parameter in SAP HANA is hidden (not listed in the Parameter Reference Guide) and disabled by default, but can be used to apply a soft (percentage-based) limit as a proportion of the available disk space. It is available in the `persistent_memory` section of the `<service>.ini` file:

Parameter	**volume_quota_rel**
Description	Apply a percentage limit threshold (or quota) on filesystem space usage for PMEM devices for each NVM device mount.
Change	Online
Default	100 (100% therefore disabled by default).

If the parameter has been set, a check is made on the installed mount point(s) (defined in the basepath value(s)) to see if NVM data currently exceeds the limit. The check refers to the `NVMVolumeManager` to assess the projected used filesystem space of the mount path as a percentage of the total filesystem space.

If the limit is reached (or exceeded) it will no longer be possible to create more blocks and an out of memory error will be generated with the reason 'NVM_MOUNT_SPACE_LIMIT'.

Monitoring Views

For full details of the monitoring views and values named here, refer to the *SAP HANA SQL and System Views Reference*.

The following monitoring views show information about persistent memory in the PERSISTENT_MEMORY column: M_CS_TABLES, M_CS_COLUMNS and M_CS_ALL_COLUMNS. This information shows either TRUE (persistent memory is enabled) or FALSE.

At the level of tables (M_CS_TABLES), usage information is shown as PERSISTENT_MEMORY_SIZE_IN _TOTAL.

At the level of columns (M_CS_COLUMNS and M_CS_ALL_COLUMNS) usage information is shown in a series of columns named: MAIN_PERSISTENT_MEMORY_SIZE_IN_*. These columns show usage in the categories DATA, DICT, INDEX, MISC, and UNUSED. These column views also include STORED_IN_PERSISTENT_MEMORY and LOADED_FROM_PERSISTENT_MEMORY, both TRUE/FALSE values.

Three monitoring views show full statistical details of persistent memory usage:

- M_PERSISTENT_MEMORY_VOLUMES - capacity, usage, and metadata of persistent memory volumes configured per NUMA Node.
- M_PERSISTENT_MEMORY_VOLUME_DATA_FILES - metadata statistics about files created by SAP HANA services for data storage on the persistent memory volumes.

- M_PERSISTENT_MEMORY_VOLUME_STATISTICS - statistics of physical lifecycle events of blocks managed by SAP HANA services on the persistent memory volumes.

The view M_TABLE_PERSISTENT_MEMORY_FILES shows persistent memory file information for all of the tables, partitions, and columns in the database.

Statistics server collectors save historic information in the following statistics views:

- _SYS_STATISTICS.HOST_PERSISTENT_MEMORY_VOLUMES - reports the capacity, usage, and metadata of persistent memory volumes.
- _SYS_STATISTICS.HOST_PERSISTENT_MEMORY_VOLUME_STATISTICS - reports the statistics of physical lifecycle events of blocks managed by SAP HANA services on the persistent memory volumes.

Related Information

SAP Note 2618154
SAP HANA Server Installation and Update Guide
SAP HANA Fast Restart Option

8.9.4 SAP HANA Fast Restart Option

The SAP HANA Fast Restart option uses storage in the file system to preserve and reuse MAIN data fragments to speed up SAP HANA restarts. This is effective in cases where the operating system is not restarted.

The SAP HANA Fast Restart option makes it possible to reuse MAIN data fragments after an SAP HANA service restart without the need to re-load the MAIN data from the persistent storage. It leverages the SAP HANA Persistent Memory implementation with the key performance benefit of greatly accelerated start-up times so that data can be quickly reloaded into memory and system downtimes can be minimized. In contrast to Persistent Memory which works with a DAX enabled file system, the Fast Restart option stores MAIN data fragments in tmpfs file systems with content in DRAM. Tmpfs file systems can grow and shrink dynamically.

The Fast Restart option can be used with all supported SAP HANA hardware platforms and all supported operating system versions. The HANA online performance and sizing KPIs are not impacted.

Comparison of Persistent Memory and File System Storage

The Fast Restart option is an alternative technology for environments where servers with Persistent Memory are not available and cannot be used in conjunction with Persistent Memory. It provides less downtime but not the cost efficiency which can be gained with Persistent Memory. The following table compares the two features and indicates in which cases in-memory data of MAIN data fragments can be reused without reading the data from the persistent storage.

Scenario	Persistent Memory	Fast Restart Option
HANA restart	Yes	Yes
HANA service restart	Yes	Yes

Scenario	Persistent Memory	Fast Restart Option
HANA upgrade/service patch	Yes	Yes
Operating system update/service patch	Yes	No
Server reboot (planned)	Yes	No
Server reboot (HW failure)	Yes	No
Power outage	Yes	No
BIOS/firmware update	Yes	No
HW upgrade w/o Persistent Memory replacement	Yes	No

Preparation

To use this feature you must create a mount on the tmpfs file system for each NUMA node and HANA instance and set the basepath parameter to identify these locations (see *Configuration* below).

Firstly, use the following command to determine the number of CPU sockets available in the HANA server:

```
> cat /sys/devices/system/node/node*/meminfo | grep MemTotal | awk 'BEGIN
{printf "%10s | %20s\n", "NUMA NODE", "MEMORY GB"; while (i++ < 33) printf "-";
printf "\n"} {printf "%10d | %20.3f\n", $2, $4/1048576}'
```

This returns the node details, for example, two nodes numbered zero and one:

```
NUMA NODE |            MEMORY GB
----------------------------------
        0 |              376.638
        1 |              376.514
```

Make a directory and mount for each node equipped with memory making sure that all tmpfs mounts have preferred NUMA policy, for example:

```
> mkdir -p /hana/tmpfs0/<SID>
> mount tmpfs<SID>0 -t tmpfs -o mpol=prefer:0 /hana/tmpfs0/<SID>
> mkdir -p /hana/tmpfs1/<SID>
> mount tmpfs<SID>1 -t tmpfs -o mpol=prefer:1 /hana/tmpfs1/<SID>
```

Allow access to the directories:

```
> chown -R <SID>adm:sapsys /hana/tmpfs*/<SID>
> chmod 777 -R /hana/tmpfs*/<SID>
```

If you want to limit the size of these mounts you may use the SIZE option of the mount command, for example:

```
mount tmpfs<SID>0 -t tmpfs -o mpol=prefer:0,size=250G /hana/tmpfs0/<SID>
```

To ensure that the mount points are available after an operating system reboot you must also add entries into the file system table, for example: `/etc/fstab`:

```
>tmpfs<sid>0 /hana/tmpfs0/<sid> tmpfs
rw,relatime,mpol=prefer:0,uid=<SID>adm_uid,gid=sapsys_gid 0 0
>tmpfs<sid>1 /hana/tmpfs1/<sid> tmpfs
rw,relatime,mpol=prefer:1,uid=<SID>adm_uid,gid=sapsys_gid 0 0
```

Configuration

Basepath As with the other existing data and log persistent storage volumes, for MAIN data fragments the in-memory storage location basepath must be defined as a configuration parameter in the [persistence] section of the global.ini file. Enter the basepath location in the `basepath_persistent_memory_volumes` parameter (this parameter can only be set at the level of HOST). All MAIN data fragments are stored at the location defined here. Multiple locations corresponding to NUMA nodes can be defined using a semi-colon as a separator (no spaces), for example:

```
/hana/tmpfs0/<SID>;/hana/tmpfs1/<SID>
```

See also the *Persistent Memory* topic where the same basepath parameter is used.

Enablement for Partitions, Tables or Columns You can determine storage preferences for Fast Restart in the same way as for Persistent Memory by using the SQL CREATE and ALTER table statements with the `alter_persistent_memory_spec` clause.

Refer to the examples in *Persistent Memory* for details.

Global allocation limit As an alternative to setting a limit on the size of each mount as described above, overall tmpfs memory usage for all NUMA nodes for a given HANA instance and a given server node can be limited by setting the `persistent_memory_global_allocation_limit` parameter in the [memorymanager] section of the global.ini file. By default, no limit is specified, this lets HANA decide a reasonable value (usually 95% of total filesystem space of configured NVM basepaths). You can enter a value in MB to set a limit on the size of the used tmpfs space for each host. Note that when decreasing this value NVM blocks are not automatically deleted, it is still necessary to actively manage how tables are unloaded from memory as described in the following paragraphs on Unloading Tables from Memory.

Unloading Tables from Memory

The standard UNLOAD statement and also table unloads executed implicitly by HANA services due to memory running low, do not by default free the memory but simply remove the mapping of relevant data structures from the HANA service process address space and the MAIN data fragment remains - either in persistent memory or in tmpfs. The configuration parameter `table_unload_action` in the [persistent_memory] section of the indexserver.ini file can be used to control this unload behavior. The default value of the parameter for tmpfs is DELETE in which case MAIN data fragments are cleared from tmpfs, but this can be set to RETAIN if required.

Regardless of the setting of this parameter, you can also use UNLOAD with the additional DELETE PERSISTENT MEMORY clause to completely free memory, for example:

```
UNLOAD MYTABLE01 DELETE PERSISTENT MEMORY;
```

If you wish to limit the size of this main memory storage you can decrease the tmpfs file system size or use the parameter `persistent_memory_global_allocation_limit` to restrict the used tmpfs size.

Monitoring Views

The same monitoring views used by persistent memory are also used by the Fast Restart feature (see *Persistent Memory*). For example, the PERSISTENT_MEMORY column in M_CS_TABLES, M_CS_COLUMNS and M_CS_ALL_COLUMNS shows TRUE if real persistent memory or tmpfs is enabled.

For full details of the monitoring views refer to the *SAP HANA SQL and System Views Reference*.

Related Information

Persistent Memory [page 186]

8.9.5 Persistence Consistency Check

A command-line diagnosis tool is available to analyze physical corruptions of the persistent storage.

The persistence checking tool, `hdbpersdiag`, can be used to carry out page consistency checks on storage snapshot backups of SAP HANA data volumes. You can use the basic 'check all' option of the tool to verify the healthy state of data volumes. If problems occur, the tool may help to identify the cause so that an effective solution can be quickly adopted. Contact SAP Support if necessary; support staff may use the more advanced options of the tool to analyze problems in greater detail.

It is possible that although pages of data were correctly written to disk (which is guaranteed by SAP HANA verification using checksums) data is corrupted on the disk and errors occur as data from the storage snapshot backup is loaded. Errors may be related to an incorrect checksum value or to an incorrect savepoint version, for example:

```
Wrong checksum: Calculated <calc_checksum>, stored <act_checksum>.;
$lpno$=<log_pageno>; $ppno$=<phys_pageno>
```

Using the hdbpersdiag Tool

The tool, hdbpersdiag, is independent of the SAP HANA native backup process and is delivered as part of the standard installation in the following directory:

```
/usr/sap/<sid>/HDB<inst_id>/exe
```

You can run the tool and execute the 'check all' option which checks all pages for the given data volume path. Note that the data volume must not be currently in use by HANA:

```
hdbpersdiag
check all <data_volume_path>
```

From the output, it may be possible to determine the name of the table(s) which are affected. Possible actions to restore the table include recovering the corrupt page from the trace directory, merging with the data still in memory, deleting a corrupted table or deleting corrupted rows. In a high availability scenario where system replication is in place it may be possible to simply perform a takeover to a secondary or tertiary site to avoid the need to restore data from a corrupt data volume.

Full details of using the tool and examples are given in SAP Note 2272121 – *How-To: Analyzing Physical Corruptions with the SAP HANA Persistence Diagnosis Tool*.

Related Information

SAP Note 2272121

8.10 Data Tiering

SAP HANA offers various software solutions to intelligently manage multi-temperature data which brings advantages for performance, economy, and security.

Data tiering is the assignment of data to different storage and processing tiers based on the aging characteristics of the data. The tiered storage options available in SAP HANA to manage multi-temperature data (hot, warm, and cold) are introduced briefly here.

For hot data, non-volatile memory (persistent memory) can be used. This has a lower TCO than DRAM memory and can be added to SAP HANA to expand overall memory capacity and increase the volume of hot data that can be stored in-memory. Data in persistent memory is saved after a system shutdown so database startup times are faster since data does not need to be loaded into memory from disk.

For warm data SAP HANA native storage extension is a general-purpose, built-in warm data store in SAP HANA that lets you manage less-frequently accessed data without fully loading it into memory.

SAP HANA dynamic tiering is an optional add-on to the SAP HANA database. It extends SAP HANA memory by providing the ability to create, query, and load warm data into disk-based, columnar database tables (called extended tables and multistore tables).

An Extension Node is a dedicated HANA in-memory node for warm data processing that allows the SAP HANA database system to scale out at a lower cost and is appropriate for both SAP BW and native SAP HANA applications. The Extension Node is an additional SAP HANA node with relaxed memory and CPU requirements but with the same functionality as a pure SAP HANA node.

Cold data can be stored on the lowest cost storage tiers but still remain accessible to SAP HANA via SQL on request. A number of options are available for native SAP HANA applications (XSA/XSC) and SAP Business

Warehouse using SAP IQ via Near-line Storage (NLS) / Data Tiering Optimization (DTO). See the following links for further details:

- SAP HANA Data Warehousing Foundation - Data Lifecycle Manager Guide for XSC
- SAP HANA Data Warehousing Foundation Data Lifecycle Manager Guide for XSA
- Data Tiering (SAP BW/4 HANA)
- Multi-Temperature Data Management (SAP BW on SAP HANA)

Related Information

Extension Node [page 243]
Persistent Memory [page 186]
SAP HANA Native Storage Extension [page 196]
SAP HANA Dynamic Tiering

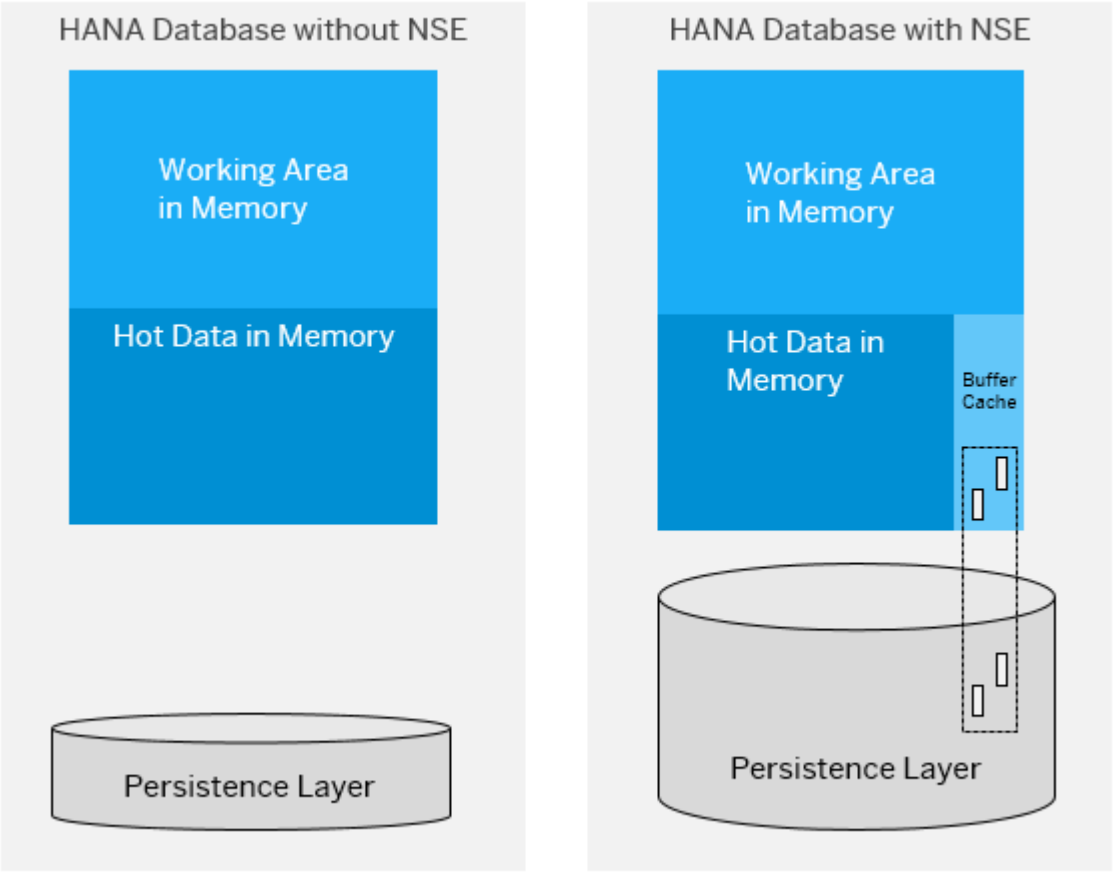
8.10.1 SAP HANA Native Storage Extension

SAP HANA Native Storage Extension (NSE) is a general-purpose, built-in warm data store in SAP HANA that lets you manage less-frequently accessed data without fully loading it into memory. It integrates disk-based or flash-drive based database technology with the SAP HANA in-memory database for an improved price-performance ratio.

SAP HANA offers various software solutions to manage multi-temperature data (hot, warm, and cold), such as DRAMs for hot data and SAP HANA Extension Nodes, SAP HANA dynamic tiering for warm data, and SAP HANA Cold Data Tiering for cold data.

- Hot data is used to store mission-critical data for real-time processing and analytics. It is retained continuously in SAP HANA memory for fast performance, and is located in the highest performance (and highest TCO) storage.
- Warm data is primarily used to store mostly read-only data that need not be accessed frequently. The data need not reside continuously in SAP HANA memory, but is still managed as a unified part of the SAP HANA database — transactionally consistent with hot data, and participating in SAP HANA backup and system replication operations, and is stored in lower cost stores within SAP HANA.
- Cold data is used to store read-only data, with very infrequent access requirements. You manage cold data separately from the SAP HANA database, but you can still access it from SAP HANA using SAP HANA's data federation capabilities.

This image shows the difference between standard HANA in-memory storage and the storage offered with NSE:



The capacity of a standard SAP HANA database is equal to the amount of hot data in memory. However, the capacity of a SAP HANA database with NSE is the amount of hot data in memory plus the amount of warm data on disk.

Since growth in data volume results in increased hardware costs, the ability to decouple data location from a fixed storage location (layer) is one of the key themes of a multi-temperature data storage strategy.

NSE is integrated with other SAP HANA functional layers, such as query optimizer, query execution engine, column store, and persistence layers. Key highlights of NSE are:

- A substantial increase in SAP HANA data capacity, with good performance for high-data volumes.
- The ability to co-exist with the SAP HANA in-memory column store, preserving SAP HANA memory performance.
- An enhancement of existing in-market paging capabilities by supporting compression, dictionary support, and partitioning.
- An intelligent buffer cache that manages memory pages in SAP HANA native storage extension column store tables.
- The ability to monitor and manage buffer cache statistics via system views.
- The ability to support any SAP HANA application.
- A simple system landscape with high scalability that covers a large spectrum of data sizes.

- An advisor that collects object access statistics and provides column store object load unit recommendations.

> → Remember
>
> - The NSE feature in SAP HANA does not require you to modify your applications.
> - Although SAP HANA 2.0 calculates 10% of memory for the buffer cache by default, this memory is only reserved and not allocated. SAP HANA accesses 100% of its memory (including the 10% reserved for the buffer cache) if you are not using NSE.
> - If there are page loadable tables in your current version of SAP HANA and you move to another, later, version, only those tables that were designated as page-loadable in the earlier version use the buffer cache in the later version (up to the limit that was calculated in the original version of SAP HANA you were running).

Related Information

Managing Warm Data With the Native Storage Extension

8.10.1.1 SAP HANA NSE Features

Key capabilities of the SAP HANA native storage extension (NSE) include page loadable objects (that is, data is stored so that individual pages can be loaded) with support for all general data types as well as large object and spatial data.

Additional capabilities of SAP HANA NSE are:

- Paging support: Specify which database tables, partitions, or columns are page loadable with the `CREATE TABLE` and `ALTER TABLE` commands. See Reduce the Memory Footprint Using Page-Loadable Columns [page 201].
- Paging support for inverted indexes: Inverted indexes on column tables (or on constraints that create these indexes) reduce the memory footprint by removing the redundant storage for strings and improve performance by providing faster unique checks. See Changing the Load Units for Indexes Using ALTER INDEX [page 209].
- Paging for dictionaries for columns having any fixed size data type.
- Buffer cache: A portion of the main memory that lets you temporarily store pages and move them between main memory and SAP HANA's persistence layer. The SAP HANA NSE buffer cache replaces the SAP HANA default page replacement model and memory limit mechanism for the memory pages. The buffer cache works with the existing SAP HANA resource and memory manager to implement page management policy. NSE is designed to reduce memory usage for specific columns, partitions, and tables by not requiring those columns to be fully memory-resident. See SAP HANA NSE Buffer Cache [page 218].

> ⓘ Note
>
> As a native feature, SAP HANA NSE supports any SAP HANA application. However, existing applications do not automatically use its capabilities, and thus application developers are responsible for making

database configuration changes to leverage the NSE feature (such as specifying a load unit in configuration parameters) while creating or altering database tables.

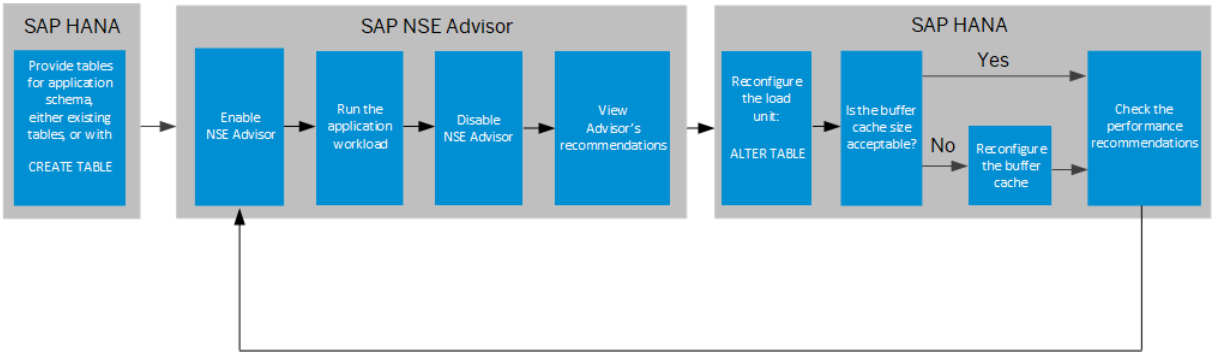
8.10.1.2 SAP HANA NSE Functional Restrictions

The SAP HANA native storage extension is subject to certain functional restrictions. For details, see the 3508592 *SAP HANA Native Storage Extension 2.0 SPS 08 Functional Restrictions*.

8.10.1.3 Getting Started With SAP HANA NSE

Use the command line or the SAP Cockpit to use SAP HANA NSE.

The steps for configuring SAP HANA for NSE are:



Use this syntax to perform SAP HANA NSE tasks:

Task	Syntax	For more information see:

Set the load unit for columns, tables, and partitions.	`CREATE [ <table_type> ] { <table_name> | <replica_name> } ... <load_unit>`
	Where:
	`<load_unit>::= PAGE LOADABLE | COLUMN LOADABLE`
		<load_unit> section in CREATE TABLE Statement (Data Definition)

Alter the load unit for existing tables, partitions, primary keys, unique	`ALTER TABLE <table_name> ... <load_unit>`
	Where:
	`<load_unit>::= PAGE LOADABLE | COLUMN LOADABLE | DEFAULT LOADABLE`
		<table_load_unit_clause> section in ALTER TABLE Statement (Data Definition)

Task	Syntax	For more information see:
constraints, and indexes.		

Table Partitions: Note that to use NSE on table partitions, the PAGE LOADABLE load unit can only be applied to tables using heterogeneous partitioning. The <load_unit> clause in the CREATE / ALTER TABLE statements is only available for heterogeneous partitioning – see, for example, Heterogeneous Alter Partition Clauses <alter_partition_load_unit_clause> in the *SAP HANA SQL Reference Guide*.

> → Remember
>
> The only difference between normal tables and tables in NSE is that in the latter the table is of a paged load unit. To the outside (for example, to query engines), there is no functional difference, the only difference is in how certain operations perform; for instance, point queries on cold (unloaded columns) may be faster on paged data because only a few pages are brought into memory.

Related Information

Understanding the NSE Advisor [page 225]
SAP HANA NSE Buffer Cache [page 218]

8.10.1.4 Using Indexes in SAP HANA NSE

Indexes are stored on individual columns in a column store in SAP HANA Native Storage Extension (NSE), and contain the same data that is stored in the column, but sorted differently.

Create your indexes using either a PRIMARY KEY or using the CREATE INDEX command. Regardless of the method used, you can create an index on a single column or on multiple columns.

Indexes created on a single column add an inverted index structure to the existing column, and if this column is configured with a paged load unit, the index structure is also paged.

Indexes created over multiple columns include an internal column that materializes the index values, and is paged according to:

- The load unit configuration of the table and partition on which the column resides.
- The load unit explicitly specified by the ALTER INDEX commands used to alter the index. See ALTER INDEX Statement (Data Definition).

This means the sizes of column-store indexes are comparable to the size of the data, so if the data size is larger and needs to be paged, you can page the indexes as well.

If there is sufficient memory available, keeping all indexes in memory does not hurt — and in some cases, improves — performance. However, you may not need to keep all indexes in memory to get the desired performance.

Indexes in memory are useful if the column is in memory and the index lookups and range queries are frequently accessed.

You can use inverted hash indexes on column tables. See Changing the Load Units for Indexes Using ALTER INDEX [page 209].

8.10.1.5 Reduce the Memory Footprint Using Page-Loadable Columns

SAP HANA native storage extension (NSE) uses techniques to load only the pages into memory that include data that is relevant to your search. Pages containing data that is not accessed by your query are not loaded from disk.

NSE supports column store tables, which support page-loadable columns and designate a column, partition, or an entire table as either page loadable or column loadable. Use the ALTER TABLE and CREATE TABLE commands to specify the `<load_unit>` preferences for column, table, or partition. Setting `<load_unit>` to PAGE enables the columns to load its structures in a pagewise manner into main memory. Specifying the `<load_unit>` as COLUMN loads the entire database object into the memory (the load unit designation only defines the behavior of the column load; it does not perform the load itself).

Page-loadable columns provide benefits in these situations for tables, partitions, or columns that are less frequently accessed:

- Using a fully qualified primary key – allows for a smaller working memory footprint for the page-loadable columns when this OLTP-style data access (during which all columns of a row are accessed, usually qualified by primary key) is compared to the standard column table behavior. NSE uses a smaller memory footprint for OLTP-style access because only a portion of the index columns are needed to locate the row, and only a portion of each column must be loaded to extract the row, as opposed to the entire set of columns that are required for a typical in-memory access.
  For a page loadable column, the read performance may vary depending on whether the page is in the buffer cache or if it must be read from disk. Internal primary key and index columns are implicitly paged in partitions with a load unit of PAGE (see Using Indexes in SAP HANA NSE [page 200]).
- Without a fully qualified primary key – page-loadable columns need a small memory footprint for processing with this OLAP-style data access (which scans all data of a single column), unlike standard column table behavior. NSE uses a smaller memory footprint for OLAP-style data access because only one portion at a time is required to scan the column instead of requiring the whole column to be in memory. This comes at a cost of slightly slower column scan performance due to reads from persistent storage during the query execution.

Typically, SAP HANA keeps all column-loadable data completely in memory for fast processing. For NSE, SAP HANA chooses an in-memory or paged-loadable representation based on the size of the column, access pattern, and so on, using the buffer cache to reduce the memory footprint for warm data.

The query performance on warm data may be somewhat reduced compared to the performance on hot data. You can convert data from column-loadable to page-loadable and vice versa.

> **Note**
>
> Paging is supported for the main store portion of the NSE-enabled columns. The delta store portion remains fully memory resident and is unaffected by the page loadable feature. Index structures on main store page-loadable columns are pageable.

Page-loadable columns provide a paged persistence for data and manage the pages in a buffer-cache, similar to traditional database systems. Use the ALTER TABLE and CREATE TABLE commands with the `<load_unit>`

value (COLUMN, PAGE, or DEFAULT) to specify which columns, tables, and table partitions are page loadable and which are column loadable.

> **Note**
>
> To move a portion of an unpartitioned table to SAP HANA NSE, you must first partition the table.

Related Information

Changing the Load Unit Using ALTER TABLE [page 205]
Understanding Load Unit Behavior in SAP HANA NSE Column Store Tables [page 202]

8.10.1.5.1 Configure SAP HANA to Use Page-Loadable Column Storage

You can configure columns, partitions, tables, and indexes by configuring the `<load_unit>` parameter in DDL statements.

The DDL statements you can use to configure load units for column store storage are:

- CREATE TABLE
- ALTER TABLE
- ALTER INDEX

8.10.1.5.1.1 Understanding Load Unit Behavior in SAP HANA NSE Column Store Tables

The loading behavior is determined by the load unit (one of PAGE, COLUMN, and DEFAULT) specified for the column, partition, and table in SAP HANA native storage extension (NSE) column-store tables.

A column can be an in-memory column (which has a load unit of COLUMN) or a paged-attribute column (which has a load unit of PAGE), depending on its specified load unit. When you run a DDL (for example, ALTER TABLE), SAP HANA determines the effective load unit for a column based on load unit precedence as described below.

To determine whether a column is fully in-memory or is paged, SAP HANA checks the load unit that is being set at the column level, described in the table below:

Load unit is set to:	The effective load unit is:
If the column-level load unit is configured as either PAGE LOADABLE or COLUMN LOADABLE	The configured load unit.

Load unit is set to:	The effective load unit is:
If the column-level load unit is configured as DEFAULT LOADABLE	SAP HANA checks, and uses, the partition-level load unit.
If the partition-level load unit is configured as PAGE LOADABLE or COLUMN LOADABLE	The configured load unit.
If the partition-level load unit is configured as DEFAULT LOADABLE	SAP HANA checks, and uses, the table-level load unit.
If the table load unit is configured as PAGE LOADABLE or COLUMN LOADABLE	The configured load unit.
If the table load unit is configured as DEFAULT LOADABLE	The COLUMN LOADABLE load unit.

The effective load unit also depends on factors such as whether the column is pageable or not. For example, text columns cannot be paged.

> ⓘ Note
>
> DEFAULT indicates that the user does not have any explicit load unit preference for a database object. A database object with the tag as DEFAULT LOADBALE inherits the load unit preference of its parent object. For example, a column inherits its partition's preference and a partition inherits its table's load unit preference. The system default load unit is COLUMN LOADABLE.

Keep in mind the following points while using DEFAULT load unit in the `CREATE TABLE` and `ALTER TABLE` commands:

- For `CREATE TABLE`: If you do not provide any explicit load unit preference, the system uses load unit as DEFAULT.
- For `ALTER TABLE` – If you want to reset the existing load unit preference, you must specify DEFAULT LOADABLE. If you do not specify any load unit tag in the `ALTER TABLE` command (for instance, if you are changing the default value of a column but do not want to change the load unit), the existing load unit is retained.

8.10.1.5.1.2 Setting the Load Unit Using CREATE TABLE

Use the CREATE TABLE statement with the `load_unit` parameter to set the load unit for tables, any heterogenous RANGE partition, and columns. Note that you can't use NSE load unit syntax for time selection tables.

See CREATE TABLE.

> ⓘ Note
>
> Persistent memory (NVRAM) is supported in SAP HANA as a persistent storage type, and is enabled using DDL statements (such as CREATE and ALTER TABLE) using the `PERSISTENT MEMORY` switch at the table, partition, or column level. Although you can specify both paging (by using `PAGE LOADABLE` as the load unit) and persistent memory (by using the `PERSISTENT MEMORY ON` switch) on the same column,

partition, or table, the load unit is given higher precedence over persistent memory. This means that if a column has both PAGE LOADABLE and PERSISTENT MEMORY ON, data is not persisted into NVRAM because page-loadable columns are not supported on persistent memory.

Creating NSE-Enabled Tables at the Table Level

This example sets the load unit to page loadable for the entire table:

```
CREATE COLUMN TABLE T1 (C1 INT) PAGE LOADABLE
```

This example sets the load unit to page loadable for the `C2` column:

```
CREATE COLUMN TABLE T (C1 INT, C2 VARCHAR (10) PAGE LOADABLE)
```

This example creates a table with the load units set for column and tables:

```
CREATE TABLE T2 (COL1 INT COLUMN LOADABLE, COL2 INT) PAGE LOADABLE;
```

Viewing the load unit on table T2 shows that it includes both page- and column-loadable columns:

```
SELECT TABLE_NAME, COLUMN_NAME,LOAD_UNIT FROM M_CS_COLUMNS WHERE TABLE_NAME='T2';
```

```
TABLE_NAME  COLUMN_NAME  LOAD_UNIT
        T2         COL1     COLUMN
        T2         COL2       PAGE
```

Creating NSE-Enabled Tables with Partitions

See NSE-Enabled Partitioned Tables [page 210] for information about changing the load unit for partitioned tables.

Creating NSE-Enabled HDBtables

Use this syntax to create NSE-enabled HDBtables:

```
COLUMN TABLE <TABLE_NAME> (
  <COLUMN_NAME> [, <COLUMN_NAME>] PAGE  LOADABLE,
)
```

For example, this creates the T3 HDBtable, which includes the page loadable column, COL1:

```
COLUMN TABLE T3 (
  COL1 INT PAGE  LOADABLE,
  COL2 INT
)
```

This creates the T4 HDBtable, which includes a page-loadable partition:

> **Note**
>
> The double parenthesis surrounding the partition declaration are syntactically required.

```
COLUMN T4 (COL1 INT, COL2 INT)
PARTITION BY RANGE(COL1) ((PARTITION 0   <= VALUES < 10 PAGE LOADABLE, PARTITION
OTHERS))
```

This creates a page-loadable HDBtable, T1:

```
COLUMN T1 (COL1 INT, COL2 CHAR) PAGE LOADABLE
```

8.10.1.5.1.3 Changing the Load Unit Using ALTER TABLE

Specify the `load_unit` parameter to change the load unit at a table, partition, or column level. The NSE syntax for ALTER TABLE offers improved performance and is non-blocking.

The syntax is:

```
ALTER TABLE <TABLE_NAME> …  <LOAD_UNIT_WITH_CASCADE>;
ALTER TABLE <TABLE_NAME> <PARTITION_ID> | <RANGE> <LOAD_UNIT_WITHOUT_CASCADE>
```

Where:

```
<LOAD_UNIT>::= { COLUMN | PAGE | DEFAULT } LOADABLE
```

Changing the Load Unit for a Table

This example alters the load unit for a table:

```
ALTER TABLE T PAGE LOADABLE CASCADE
```

Use of `CASCADE` removes any existing partition or column-level load unit preferences.

Changing the Load Unit for a Column

This example alters a load unit of column to be DEFAULT LOADABLE, which means there is no explicit preference for the paging attribute that is used when the results are loaded from the table:

```
ALTER TABLE T1 ALTER (C1 INT DEFAULT LOADABLE)
```

Changing the Load Unit for a Partition

Example 1: This example alters a partition's load unit specification (for range partition only):

```
ALTER TABLE T4 ALTER PARTITION 5 PAGE LOADABLE
```

Viewing the load unit on table T4 shows that columns in partition 5 are now page-loadable:

```
SELECT TABLE_NAME, PART_ID, COLUMN_NAME,LOAD_UNIT FROM M_CS_COLUMNS WHERE
TABLE_NAME='T4';
```

```
TABLE_NAME   PART_ID  COLUMN_NAME   LOAD_UNIT
        T4         1           C1      COLUMN
        T4         1           C2      COLUMN
        T4         2           C1        PAGE
        T4         2           C2        PAGE
        T4         3           C1      COLUMN
        T4         3           C2      COLUMN
        T4         4           C1      COLUMN
        T4         4           C2      COLUMN
        T4         5           C1        PAGE
        T4         5           C2        PAGE
```

Example 2: This example alters a partition's load unit by specifying the `RANGE` clause:

```
ALTER TABLE TAB ALTER PARTITION RANGE(A)
    ((PARTITION 0<= VALUES <10)) PAGE LOADABLE
```

Using ALTER TABLE to Change Load Units with a Reduced Blocking Window

Changing the loadable property of a column requires that the column data is re-written in the new format. However, ALTER TABLE supports parallel DMLs that run until the next savepoint is triggered. If you specify multiple columns in the same DDL, all columns are converted in parallel according to the availability of the CPUs, greatly improving the performance of this task.

The syntax is:

```
ALTER TABLE <TABLE_NAME> ALTER (<COLUMN_NAME> ALTER [COLUMN | PAGE] LOADABLE)
```

To run ALTER TABLE with a reduced window for blocking:

- The DDL AUTOCOMMIT property must be enabled:

  ```
  SET TRANSACTION AUTOCOMMIT DDL ON
  ```

- You cannot specify other ALTER clauses with the LOAD UNIT conversion clause.

> **Note**
>
> DDLs that are not online typically do not allow parallel DMLs to run.

For example:

- This alters columns C1 and C2 in parallel to be page- and column-loadable, respectively:

```
ALTER TABLE T1 ALTER (C1 ALTER PAGE LOADABLE, C2 ALTER COLUMN LOADABLE)
```

Using ALTER TABLE Command for Compressed and Uncompressed Columns

- If you execute the following ALTER TABLE command on a compressed column, the C1 column remains compressed with the new load unit specified:

```
ALTER TABLE T ALTER (C1 ALTER <LOAD_UNIT>)
```

- If you execute the following ALTER TABLE command on a compressed column, the C1 column is uncompressed with the new load unit specified:

```
ALTER TABLE T ALTER (C1 INT <LOAD_UNIT>)
```

  Where:

```
<LOAD_UNIT> = {PAGE|COLUMN|DEFAULT} LOADABLE
```

> ### Note
>
> The uncompressed columns will remain uncompressed after you execute the above commands.

Monitoring for NSE DDL

The following NSE DDL show their status in the monitoring view M_JOB_PROGRESS.

Alter Column Load Unit

```
ALTER TABLE <TABLE_NAME> ALTER (<COLUMN_NAME> ALTER <LOAD_UNIT> LOADABLE)
```

This DDL has a master operation and multiple sub-operations. The number of sub-operations are (number of partitions * number of columns).

The status for master operations are as follows:

OBJECT_NAME	JOB_NAME	CURRENT_PROGRESS	MAX_PROGRESS	PROGRESS_DETAIL
<TABLE_NAME>	Alter Column Load Unit	1	5	Initial
<TABLE_NAME>	Alter Column Load Unit	2	5	Exclusive phase 1 request

OBJECT_NAME	JOB_NAME	CURRENT_PROGRESS	MAX_PROGRESS	PROGRESS_DETAIL
<TABLE_NAME>	Alter Column Load Unit	3	5	Altering Columns
<TABLE_NAME>	Alter Column Load Unit	4	5	Exclusive phase 2 request
<TABLE_NAME>	Alter Column Load Unit	5	5	**Note** No progress detail for the last phase because job finishes and its entry is removed from M_JOB_PROGRESS.

The status for sub-operations are as follows:

OBJECT_NAME	JOB_NAME	CURRENT_PROGRESS	MAX_PROGRESS	PROGRESS_DETAIL
<TABLE_NAME>~<PARTITION_ID>~<COLUMN_NAME>	Alter Column Load Unit	1	4	Preparing column conversion for Column <COLUMN_NAME> (part <PARTITION_ID>)
<TABLE_NAME>~<PARTITION_ID>~<COLUMN_NAME>	Alter Column Load Unit	2	4	Data conversion for Column <COLUMN_NAME> (part <PARTITION_ID>)
<TABLE_NAME>~<PARTITION_ID>~<COLUMN_NAME>	Alter Column Load Unit	3	4	Metadata conversion for Column <COLUMN_NAME> (part <PARTITION_ID>)
<TABLE_NAME>~<PARTITION_ID>~<COLUMN_NAME>	Alter Column Load Unit	4	4	**Note** No progress detail for the last phase because job finishes and its entry is removed from M_JOB_PROGRESS.

Alter Table Load Unit

M_JOB_PROGRESS displays the same status as Alter Column Load Unit except JOB_NAME, which displays “Alter Table Load Unit”.

```
ALTER TABLE <TABLE_NAME> <LOAD_UNIT> LOADABLE
```

Alter Partition Load Unit

M_JOB_PROGRESS displays the same status as Alter Column Load Unit except JOB_NAME, which displays “Alter Partition Load Unit”.

```
ALTER TABLE <TABLE_NAME> <PARTITION_SPEC> <LOAD_UNIT> LOADABLE
ALTER TABLE <TABLE_NAME> <PART_ID> <LOAD_UNIT> LOADABLE
```

8.10.1.5.1.4 Changing the Load Units for Indexes Using ALTER INDEX

Using paged value indexes on column tables reduces the memory footprint. This example creates a unique inverted value index named idx on columns `col1` and `col2` on table `COLUMN_TABLE`.

```
CREATE UNIQUE INVERTED VALUE INDEX idx ON COLUMN_TABLE(col1, col2);
```

This example converts an index named `idx`, to page loadable.

```
ALTER INDEX idx PAGE LOADABLE;
```

> ⓘ Note
>
> Paging is supported for inverted value indexes, which are defined on multiple columns.

Monitoring for DDL Alter Index Load Unit

The monitoring view M_JOB_PROGRESS displays the status of DDL Alter Index Load Unit. The status is the same as DDL Alter Column Load Unit except JOB_NAME, which displays “Alter Index Load Unit”. For more information see Monitoring for NSE DDL in Changing the Load Unit Using ALTER TABLE [page 205]

The syntax is:

```
ALTER INDEX <INDEX_NAME> <LOAD_UNIT> LOADABLE
```

8.10.1.5.1.5 Viewing Load Unit Information for Column Store Tables in SAP HANA NSE

Several system and monitoring views provide information on load unit preferences (PAGE loadable, COLUMN loadable, or DEFAULT loadable) set for the SAP HANA NSE column-store table.

- System views
  - TABLES
  - TABLE_PARTITIONS
  - TABLE_COLUMNS
  - INDEXES – shows index load unit for supported index types.

  > ### Note
  >
  > `DEFAULT` indicates that the user does not have any explicit load unit preference for a database object. A database object with the tag `DEFAULT LOADBALE` inherits the load unit preference of its parent object. For example, a column inherits its partition's load unit preference and a partition inherits its table's. The system default load unit is `COLUMN LOADABLE`.

- Monitoring views: The `LOAD_UNIT` column is added to display the derived load unit at a given granularity level, following the precedence logic.
  - M_CS_TABLES
  - M_TABLE_PARTITIONS
  - M_CS_COLUMNS
  - M_CS_ALL_COLUMNS

8.10.1.5.1.6 NSE-Enabled Partitioned Tables

For SAP HANA NSE tables, partitioning can be heterogeneous or non-heterogenous. However, partition load unit can be set only for heterogeneous partitioning. This scenario describes a sequence of steps creating a page-loadable, partitioned table and then altering its load unit.

NSE supports only the RANGE and RANGE-RANGE HETEROGENEOUS partitioning schemes. If you have a RANGE or RANGE-RANGE partition, you can define the load unit at any level of the partitions.

> ### Note
>
> For multistore tables in heterogeneous partitioning, you cannot specify an ENABLE DELTA clause in the `CREATE TABLE` statement. First create a non-delta-enabled heterogenous multistore table and then use the `ALTER TABLE` command to enable delta for the table.

Creating and Altering Single-Level NSE-Enabled Partitions

These examples show how to convert a single-level partitioned table from non-heterogeneous to heterogeneous. The example creates a single-level non-heterogenous partitioned table:

```
CREATE COLUMN TABLE T (C1 INT) PARTITION BY RANGE (C1)
(PARTITION 0 <= VALUES < 1000, PARTITION 1000 <= VALUES < 2000, PARTITION
OTHERS );
```

This statement alters table T with heterogenous partitioning:

```
ALTER TABLE T PARTITION BY RANGE (C1)
((PARTITION 0 <= VALUES < 1000 , PARTITION 1000 <= VALUES < 2000 , PARTITION
OTHERS));
```

This example uses the NSE syntax to create an NSE-enabled, single-level heterogenous partitioned table. Note that the double parenthesis surrounding the partition declaration are syntactically required by CREATE TABLE.

```
CREATE COLUMN TABLE T (C1 INT) PARTITION BY RANGE (C1)
((PARTITION 0 <= VALUES < 1000 PAGE LOADABLE, PARTITION 1000 <= VALUES < 2000
COLUMN LOADABLE, PARTITION OTHERS COLUMN LOADABLE));
```

View the load units to verify the change:

```
SELECT P.schema_name,P.table_name,P.partition,P.range,P.subpartition,P.subrange,
TP.part_id,TP.load_unit,TP.loaded,TP.record_count FROM M_CS_PARTITIONS P INNER
JOIN M_TABLE_PARTITIONS TP
ON P.part_id = TP.part_id AND P.schema_name = TP.schema_name AND P.table_name =
TP.table_name
WHERE P.table_name = 'T';
```

```
SCHEMA_NAME  TABLE_NAME  PARTITION      RANGE  SUBPARTITION  SUBRANGE  PART_ID
LOAD_UNIT  LOADED  RECORD_COUNT
     SYSTEM           T          1     0-1000             1
1      PAGE     FULL             0
     SYSTEM           T          2  1000-2000             1
2    COLUMN     FULL             0
     SYSTEM           T          3                        1
3    COLUMN     FULL             0
```

This changes the load unit for partition 2 to be page loadable:

```
ALTER TABLE T ALTER PARTITION RANGE (C1) ((PARTITION 1000 <= VALUES < 2000))
PAGE LOADABLE;
```

Creating and Altering Second-Level NSE-Enabled Partitions

These examples show altering second level partitions.

Create table T2, which includes this first and second-level heterogeneous partitioning:

```
CREATE COLUMN TABLE T ( C1 INT, C2 INT)
      PARTITION BY RANGE (C1)
((PARTITION 0 <= VALUES < 1000, PARTITION 1000 <= VALUES < 2000 PAGE LOADABLE,
PARTITION OTHERS)
```

```
      SUBPARTITION BY RANGE (C2)
      (PARTITION 0 <= VALUES < 100,
      PARTITION 100 <= VALUES < 200, PARTITION 200 <= VALUES < 300 , PARTITION
OTHERS));
```

View the load unit on table T to verify the changes

```
SELECT P.schema_name,P.table_name,P.partition,P.range,P.subpartition,P.subrange,
TP.part_id,TP.load_unit,TP.loaded,TP.record_count FROM M_CS_PARTITIONS P INNER
JOIN M_TABLE_PARTITIONS TP
ON P.part_id = TP.part_id AND P.schema_name = TP.schema_name AND P.table_name =
TP.table_name
WHERE P.table_name = 'T';
```

```
SCHEMA_NAME  TABLE_NAME  PARTITION      RANGE  SUBPARTITION  SUBRANGE  PART_ID
LOAD_UNIT  LOADED  RECORD_COUNT
     SYSTEM           T          1     0-1000             1     0-100
1     COLUMN    FULL             0
     SYSTEM           T          1     0-1000             2   100-200
2     COLUMN    FULL             0
     SYSTEM           T          1     0-1000             3   200-300
3     COLUMN    FULL             0
     SYSTEM           T          1     0-1000             4
4     COLUMN    FULL             0
     SYSTEM           T          2  1000-2000             1     0-100
5        PAGE    FULL             0
     SYSTEM           T          2  1000-2000             2   100-200
6        PAGE    FULL             0
     SYSTEM           T          2  1000-2000             3   200-300
7        PAGE    FULL             0
     SYSTEM           T          2  1000-2000             4
8        PAGE    FULL             0
     SYSTEM           T          3                        1     0-100
9     COLUMN    FULL             0
     SYSTEM           T          3                        2   100-200
10     COLUMN    FULL              0
     SYSTEM           T          3                        3   200-300
11     COLUMN    FULL              0
     SYSTEM           T          3                        4
12     COLUMN    FULL              0
```

This example changes the load unit for a second-level partition corresponding to a first-level partition. To change the second-level partition for other first-level partitions, you must use the command multiple times on this specific first-level partition:

```
ALTER TABLE T ALTER PARTITION
RANGE (C1) ((PARTITION 0 <= VALUES < 1000)
SUBPARTITION BY RANGE(C2) (PARTITION 200 <= VALUES < 300)) PAGE LOADABLE;
```

Changing the load unit for the first-level partition on T:

```
ALTER TABLE T ALTER PARTITION RANGE (C1) ((PARTITION 1000 <= VALUES < 2000))
COLUMN LOADABLE;
```

View the load unit on table T. The second-level partitions, 4 – 12 now show as column-loadable:

```
SELECT P.schema_name,P.table_name,P.partition,P.range,P.subpartition,P.subrange,
TP.part_id,TP.load_unit,TP.loaded,TP.record_count FROM M_CS_PARTITIONS P INNER
JOIN M_TABLE_PARTITIONS TP
ON P.part_id = TP.part_id AND P.schema_name = TP.schema_name AND P.table_name =
TP.table_name
WHERE P.table_name = 'T';
```

```
SCHEMA_NAME  TABLE_NAME  PARTITION      RANGE  SUBPARTITION  SUBRANGE  PART_ID
LOAD_UNIT  LOADED  RECORD_COUNT
     SYSTEM           T          1     0-1000             1     0-100
1     COLUMN    FULL              0
     SYSTEM           T          1     0-1000             2   100-200
2     COLUMN    FULL              0
     SYSTEM           T          1     0-1000             3   200-300
3       PAGE    FULL              0
     SYSTEM           T          1     0-1000             4
4     COLUMN    FULL              0
     SYSTEM           T          2  1000-2000             1     0-100
5     COLUMN    FULL              0
     SYSTEM           T          2  1000-2000             2   100-200
6     COLUMN    FULL              0
     SYSTEM           T          2  1000-2000             3   200-300
7     COLUMN    FULL              0
     SYSTEM           T          2  1000-2000             4
8     COLUMN    FULL              0
     SYSTEM           T          3                        1     0-100
9     COLUMN    FULL              0
     SYSTEM           T          3                        2   100-200
10     COLUMN    FULL              0
     SYSTEM           T          3                        3   200-300
11     COLUMN    FULL              0
     SYSTEM           T          3                        4
12     COLUMN    FULL              0
```

8.10.1.5.1.7 SAP HANA NSE Support for the Statistics Server

The column tables of SAP HANA statistics server don't use SAP HANA NSE by default. Thus, if you're using NSE in your SAP HANA 2.0 environment, SAP recommends you to activate NSE for the column tables of the SAP HANA statistics server. Use the information here for the required guidance.

Check the NSE Setting for Statistics Server

To check the current NSE setting for SAP HANA statistics server, execute the following SQL query:

```
SELECT DISTINCT LOAD_UNIT FROM SYS.TABLES WHERE SCHEMA_NAME = '_SYS_STATISTICS'
AND IS_COLUMN_TABLE = 'TRUE'
AND IS_USER_DEFINED_TYPE = 'FALSE';
```

On analysing the query result and the current NSE setting, you need to activate or deactivate NSE for column tables of the SAP HANA statistics server by using either the PAGE LOADABLE or DEFAULT LOADABLE clause. See the following sections for details on doing the needful.

Activate NSE for Column Tables of SAP HANA Statistics Server by Using the 'PAGE LOADABLE' Clause

Configure all statistics server column tables in NSE to use **PAGE LOADABLE** by calling the procedure `SHARED_ALTER_PAGE_LOADABLE` in the `_SYS_STATISTICS` schema.

Call this procedure as follows to activate NSE (PAGE LOADABLE) for all statistics server columns:

```
CALL _SYS_STATISTICS.SHARED_ALTER_PAGE_LOADABLE;
```

Deactivate NSE for Column Tables of SAP HANA Statistics Server by Using the 'DEFAULT LOADABLE' Clause

Configure statistics server column tables in NSE to use **DEFAULT LOADABLE** by calling the procedure `SHARED_ALTER_DEFAULT_LOADABLE` in the `_SYS_STATISTICS` schema.

Call this procedure as follows to deactivate NSE (DEFAULT LOADABLE) for all statistics server columns:

```
CALL _SYS_STATISTICS.SHARED_ALTER_DEFAULT_LOADABLE;
```

Related Information

Understanding Load Unit Behavior in SAP HANA NSE Column Store Tables [page 202]

8.10.1.5.2 Monitoring View Extensions for Column Store Paged Data Size

A number of monitoring views provide information about the in-memory and on-disk size of the page-loadable data in relation to the in-memory and on-disk size of non-paged (column-loadable) data, helping you understand the effectiveness of page-loadable storage.

Related Information

Monitor Memory Paging for Non-Partitioned Tables

8.10.1.5.2.1 Configuring Paged Large Object Data Types

To designate an LOB column as page loadable, define it with the PAGE LOADABLE load unit in the `CREATE TABLE` and `ALTER TABLE` commands.

LOB data columns can be configured as either page loadable or column loadable by setting the LOAD UNIT value in the commands CREATE TABLE and ALTER TABLE. However, memory usage for LOB data types is controlled by the Hybrid LOB configuration parameters independent of the LOAD UNIT configuration. While the

LOAD UNIT can be configured to either COLUMN LOADABLE or PAGE LOADABLE for a LOB data column, it does not determine the memory usage for LOB data types.

The `M_CS_ALL_COLUMNS` system view provides the information regarding on-disk and in-memory size of paged data, and also includes the information on LOB supported columns, designated as PAGE LOADABLE. Use the following system views to review the reduction in memory footprint as a result of paged LOBs:

- Main memory – with paged LOB, the main memory size for data and dictionary is zero.
  - MAIN_MEMORY_SIZE_IN_DATA
  - MAIN_MEMORY_SIZE_IN_DICT
- Main page loadable memory – depending on the amount of LOB column being loaded into the memory, the system displays values for the main page loadable memory for data and dictionary.
  - MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DATA
  - MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DICT

As an example, run the following command to view the main and page loadable memories:

```
SELECT
MAIN_MEMORY_SIZE_IN_DATA,
MAIN_MEMORY_SIZE_IN_DICT,
MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DATA,
MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DICT
FROM M_CS_ALL_COLUMNS
WHERE TABLE_NAME = <table_name>
AND COLUMN_NAME = <column_name>
```

For example, for a table named TEST_MST_TABLE in the SAPQM7 schema:

```
SELECT
MAIN_MEMORY_SIZE_IN_DATA,
MAIN_MEMORY_SIZE_IN_DICT,
MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DATA,
MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DICT
FROM M_CS_ALL_COLUMNS
where schema_name='SAPQM7' and table_name='TEST_MST_TABLE'
```

Related Information

Hybrid LOBs (Large Objects) [page 297]

8.10.1.5.2.2 Statistics for Page-Loadable Storage

A number of monitoring views report information at the partition, column, and column sub-object level, along with the existing non-paged memory size statistics.

These views include page-loadable memory size statistics and non-paged memory statistics, so the total memory size reported is the sum of the page loadable and non-paged memory sizes, and the storage distribution is clearly understandable. This example from M_CS_ALL_COLUMNS shows that

the value of MEMORY_SIZE_IN_TOTAL results from the MEMORY_SIZE_IN_MAIN plus the value of MEMORY_SIZE_IN_DELTA:

```
        HOST    PORT  SCHEMA_NAME    TABLE_NAME  COLUMN_NAME
MEMORY_SIZE_IN_TOTAL   MEMORY_SIZE_IN_MAIN   MEMORY_SIZE_IN_DELTA
1  lsvxc0115  30,703      TPCH_PL      LINEITEM   L_ORDERKEY
37,029,428             37,025,152                 4,276
2  lsvxc0115  30,703      TPCH_PL      LINEITEM    L_PARTKEY
34,354,952             34,350,676                 4,276
3  lsvxc0115  30,703      TPCH_PL      LINEITEM    L_SUPPKEY
11,091,296             11,087,020                 4,276
```

The number reported for paged memory footprint is the size of the resident memory in the resource within the buffer cache. This is not the size of the memory in use – which is typically very small or zero, consisting only of pages currently being read or written by a client – and not the total data size on disk.

The monitoring views that include information about the size of in-memory and on-disk page-loadable data are listed below.

Monitoring View	Description

M_CS_ALL_COLUMNS	M_CS_ALL_COLUMNS includes these columns to provide statistics for page-loadable and non-paged memory footprints:
	• MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN
	• MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DATA
	• MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DICT
	• MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_INDEX
	• MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_MISC
	Other columns in M_CS_ALL_COLUMNS also include information about the page-loadable columns; use them to determine the effectiveness of your page loadable data:
	• The value of MEMORY_SIZE_IN_TOTAL = the sum of MEMORY_SIZE_IN_MAIN + MEMORY_SIZE_IN_DELTA
	• The value of MEMORY_SIZE_IN_MAIN includes the value of MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN
	• The value of MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN is compiled by adding:
	`MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DATA`
	`+ MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DICT`
	`+ MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_INDEX`
	`+ MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_MISC`
	`----------`
	`MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN = <sum>`
	• MAIN_PAGE_LOADABLE_MEMORY_SIZE_IN_DICT may have a value of 0 for data types that do not support paged dictionaries. In this situation, M_CS_ALL_COLUMNS reports the in-memory dictionary size in the MAIN_MEMORY_SIZE_IN_DICT column.
	• PAGED_MEMORY_SIZE_IN_INDEX may have a value of 0 for columns without indexes defined or loaded, and for columns that have inverted indexes (light-weight data structures that map column dictionary value IDs to the corresponding row IDs) that are not yet converted to paged indexes — this can occur in older tables that have not yet gone through a delta merge.

Monitoring View	Description
M_CS_COLUMNS	Includes the MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN column to provide page-loadable information about the total resident paged memory size of columns.
M_CS_TABLES	Contains memory statistics for tables. These are aggregates of the column memory statistics in M_CS_ALL_COLUMNS, which includes the MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN column to provide the total resident paged-memory size of a table. Other columns in M_CS_TABLES also include information about the page-loadable columns; use them to determine the effectiveness of your page loadable data: • The value for MEMORY_SIZE_IN_PAGE_LOADABLE is the same as its value in M_CS_ALL_COLUMNS. • The value for MEMORY_SIZE_IN_TOTAL = the sum of MEMORY_SIZE_IN_MAIN + MEMORY_SIZE_IN_DELTA. • The value for MEMORY_SIZE_IN_MAIN includes the value of MEMORY_SIZE_IN_PAGE_LOADABLE_MAIN.
M_TABLE_PARTITIONS	A number of columns in M_TABLE_PARTITIONS include information about the page-loadable columns; use them to determine the effectiveness of your page loadable data: • The value for MEMORY_SIZE_IN_PAGE_LOADABLE is the same as its value in M_CS_ALL_COLUMNS. • The value for MEMORY_SIZE_IN_TOTAL = the sum of MEMORY_SIZE_IN_MAIN + MEMORY_SIZE_IN_DELTA. • DISK_SIZE_IN_PAGE_LOADABLE provides the total disk size, in bytes, of page-loadable storage.

8.10.1.5.2.3 Persistent Size of Page Loadable Data

A number of monitoring views include column and column dictionary, data, and index sub-structure information about the disk size of page-loadable data (this is the size on disk, not the disk size itself).

Monitoring View	Description
M_TABLE_PERSISTENCE_STATISTICS	M_TABLE_PERSISTENCE_STATISTICS includes the DISK_SIZE of column, which describes the total on-disk size of the column. The value for DISK_SIZE also includes the size of page loadable data for the table.
M_CS_COLUMNS_PERSISTENCE	M_CS_COLUMNS_PERSISTENCE includes the MAIN_PHYSICAL_SIZE_IN_PAGE_LOADABLE column, which reports the total on-disk size stored in page-loadable format for this table. The value for MAIN_PHYSICAL_SIZE includes the value for MAIN_PHYSICAL_SIZE_PAGE_LOADABLE.

Monitoring View	Description
M_TABLE_PARTITIONS	M_TABLE_PARTITIONS includes the MEMORY_SIZE_IN_PAGE_LOADABLE column, which reports the size of the total resident-paged memory of the partition, and the DISK_SIZE_IN_PAGE_LOADABLE column, which reports the total disk size of the page-loadable storage for this partition. M_TABLE_PARTITIONS includes these columns to provide statistics for page-loadable and non-paged memory footprints: • The aggregate value of MEMORY_SIZE_IN_PAGE_LOADABLE for the partition. • The value of DISK_SIZE is the same as its value in M_TABLE_PERSISTENCE_STATISTICS. • The value of DISK_SIZE includes the value of DISK_SIZE_IN_PAGE_LOADABLE.

Related Information

SAP Note 3501427

8.10.1.6 SAP HANA NSE Buffer Cache

The SAP HANA Native Storage Extension (NSE) buffer cache replaces the SAP HANA default page replacement and memory limit mechanism for the memory pages.

> **Note**
>
> The buffer cache is enabled by default. You need not perform any tasks to enable the buffer cache.

The buffer cache works with the existing SAP HANA memory manager to implement a page management policy. The buffer cache provides page memory from the dynamically managed memory pools. SAP HANA supports multiple page sizes, ranging from 4 KB to 16 MB. The buffer cache divides its memory into multiple buffer pools, with each pool for one supported page size.

> **Note**
>
> The default size of the buffer cache is 10% of the HANA memory.

8.10.1.6.1 Buffer Cache Sizing Guidelines

Both hot and warm data in an SAP HANA system exist on disk in the data volume. SAP HANA loads hot data column-wise into memory, where it resides until the system has to handle memory pressure. SAP HANA loads warm data stored in NSE page-wise into the buffer cache, and recently-used data resides in the buffer cache.

Consequently, depending on the usage pattern and buffer cache size, SAP HANA must read data from disk, which impacts the performance of queries according to the disk bandwidth.

You should allocate enough buffer cache to avoid accessing the disk for most queries. An application's performance varies based on the amount of data that is available in the buffer cache and how much data the application must read from disk. Although the size of the buffer cache with respect to warm data size varies from application to application, as a starting point the ratio of the buffer cache to the warm data on disk should not to be smaller than 1 to 8. For example, if you have 8 GB of warm storage, you should allocate at least 1 GB of buffer cache. If your frequently-accessed NSE data set is small, you can experiment with a larger ratio of warm storage to buffer cache.

8.10.1.6.2 Adjusting the Buffer Cache Size

Adjust the size of the buffer cache to ensure that it is large enough to manage your warm data set.

You can manually configure the buffer cache size, or you can let the buffer cache calculate the appropriate values automatically based on the actual NSE data persistence size of the system.

Manual Configuration of the Buffer Cache Size

You can use the `ALTER SYSTEM ALTER CONFIGURATION` command to set the maximum size of the buffer cache in SAP HANA NSE-enabled tables and set the unload threshold value. This command is executed on the tenant database and only users with `INIFILE ADMIN` permissions can change the configuration.

The following parameters can help you increase or decrease the size of the buffer cache as needed:

- `max_size`: Indicates the maximum size allowed for the buffer cache, measured in megabytes.
- `max_size_rel`: Specifies the maximum size of the buffer cache as a percentage of the global allocation limit (GAL) or the service allocation limit, if it is set.
- `unload_threshold`: It is set as a percentage of the buffer cache's maximum size. The system periodically checks whether the `unload_threshold` is reached and releases objects accordingly to reduce memory usage.

When both `max_size` and `max_size_rel` are set, the system will use the smaller value. When these values are applied to a scale-out system at the DATABASE layer, `max_size_rel` will be relative to the allocation for each host. For example, if the `max_size_rel` is equal to 10 percent of the allocation limit for the given host, but it is capped at a `max_size` of 20 gigabytes, the relative setting (`max_size_rel`) is used for smaller hosts, whereas the absolute cap (`max_size`) is used for larger hosts. For more information on the parameters and their default values, see the Configuration Parameter Reference Guide.

The following statement configures the buffer cache capacity for SAP HANA tables. The `<cache_name>` is `buffer_cache_cs`, and `<cache_size>` is the size of the buffer cache, in megabytes.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system')
SET ('<cache_name>','max_size') = '<cache_size>' [with reconfigure]
```

SAP HANA does not allocate `max_size` on an immediate basis, only as needed. If both `max_size` and `max_size_rel` are set to non-zero values, the most restrictive value is chosen for the buffer cache.

After reducing the buffer cache, if the current data size is lower than the new and reduced upper limit, the buffer cache remains ENABLED. However, if the current memory usage exceeds the new upper limit, the buffer cache starts to shrink, and the free or used buffers are released to the SAP HANA Memory Manager until the current usage is less than or equal to the new upper limit. The buffer cache state then changes to `shrinking`.

You can also configure the buffer cache size in HANA cockpit. For more information, refer to *View and Configure the Native Storage Extension Buffer Cache Monitor*.

Examples

These examples illustrate how you can configure the buffer cache to 1500 megabytes for the SAP HANA NSE column store table using the `max_size`, `max_size_rel`, and `unload_threshold` parameters.

- Use the `max_size` parameter to set the buffer cache capacity to 1500 megabytes:

  ```
  ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system')
  SET ('buffer_cache_cs','max_size') = '1500'
  ```

- Use the `max_size_rel` parameter to set the buffer cache capacity to 15 percent of the SAP HANA memory, which in this case is 10000 megabytes. This means the buffer cache capacity gets set to 1500 megabytes:

  ```
  ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system')
  SET ('buffer_cache_cs','max_size_rel') = '15'
  [with reconfigure]
  ```

- Use the `unload_threshold` parameter to set the buffer cache to 80 per cent of the `max_size` configured at the time.

  ```
  ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system')
  SET ('buffer_cache_cs','unload_threshold') = '80'
  [with reconfigure]
  ```

The following table lists several scenarios on how to calculate buffer cache capacity based on `max_size` and `max_size_rel`:

Calculating the Effective Buffer Cache Size of the Native Storage Extension

`max_size`, in megabytes	`max_size_rel`, percentage of SAP HANA memory	Effective Buffer Cache Size	Description
Not configured	Not configured	The `page_loadable_columns_limit` value is used as the maximum cache capacity, if it is configured in `global.ini`. If the `page_loadable_columns_limit` value is not specified, then the system uses 10 percent of the SAP HANA server memory as the maximum cache capacity.	Buffer cache maximum values, with the SAP HANA memory limit set to 10000 MB: Column store buffer cache – 1000 MB
Not configured	A non-zero value	Maximum size of the buffer cache equals the `max_size_rel` value.	The non-zero value of `max_size` or `max_size_rel` determines the effective buffer cache size.
A non-zero value	Not configured	Maximum size of the buffer cache equals `max_size` value.	
A non-zero value	A non-zero value		If both the parameters are set to non-zero values, the lowest value is chosen for the buffer cache.

> **Note**
>
> Configuring `max_size` and `max_size_rel` to 0 (zero) determines effective buffer cache size to be the default value which is 10% of the global allocation limit (GAL) or the service allocation limit if set. In multitenant databases `max_size_rel` is relative to the allocation limit set for each tenant.

Automatic Calculation of `unload_threshold`

The buffer cache can automatically calculate the `unload_threshold` value, in regular intervals, based on the actual NSE data persistence size in the system. To enable this feature, define the number of seconds for the `automatic_unload_threshold_setting_interval` parameter and set the `automatic_unload_threshold_ratio`. For more information, refer to *Configuration Parameters SPS 08*.

- `automatic_unload_threshold_setting_interval`: Interval, in seconds, for the automatic calculation of the unload_threshold value. If the value is 0, it means that it is disabled.
- `automatic_unload_threshold_ratio`: By default, this calculation applies a 1/8 ratio (buffer cache size/ NSE disk size) of the actual NSE persistence size for the `unload_threshold`, but the ratio can also

be modified, for example, to 1/10 or 1/6. The buffer cache will release allocated memory that is not being used until the `unload_threshold` value is reached. This way, the allocated buffer cache size will reflect the actual NSE data persistence size in the system.

> **Note**
>
> If the calculated `unload_threshold` is higher than the configured buffer cache `max_size`, it will lead to an alert (HDBNSEBufferCacheUnloadThreshold) with the following message: "The calculated unload threshold value would be higher than 100%. As a result, it has been automatically limited to 100%". If the calculated or configured `unload_threshold` is lower than 1% of the maximum buffer cache size, no unloads will occur.

Related Information

ALTER SYSTEM ALTER CONFIGURATION Statement (System Management)
View and Configure the Native Storage Extension Buffer Cache Monitor
Configuration Parameters SPS 08

8.10.1.6.3 Monitoring the Buffer Cache

Two SAP HANA views — M_BUFFER_CACHE_STATISTICS and M_BUFFER_CACHE_POOL_STATISTICS — provide information on the buffer cache for SAP HANA NSE column store tables.

> **Note**
>
> To view past buffer cache information use HOST_BUFFER_CACHE_STATISTICS and HOST_BUFFER_CACHE_POOL_STATISTICS.

- M_BUFFER_CACHE_STATISTICS – provides information on cache configuration, cache status, and current memory usage. For example, cache name, the name of the cache replacement policy, the maximum and used (allocated) cache capacities, current state of the cache — enabled, disabling, disabled, shrinking — and the number of times a buffer has been released for reuse by the cache. See M_BUFFER_CACHE_STATISTICS.
- M_BUFFER_CACHE_POOL_STATISTICS – provides statistics on each pool in a buffer cache, such as pool size, pool's growth percentage (the memory size calculated as the percentage of the `max_size` parameter, to which the buffer pool will grow if the Free List is empty), the number of buffers currently allocated for this pool, the number of buffers in the LRU chain for the pool, and the number of buffers in the hot buffer list for the pool. See M_BUFFER_CACHE_POOL_STATISTICS.

  > **Note**
  >
  > The buffer cache maintains a buffer pool for each supported page size.

Related Information

HOST_BUFFER_CACHE_STATISTICS
HOST_BUFFER_CACHE_POOL_STATISTICS

8.10.1.6.4 Handling Out-of-Buffer Errors

When your buffer cache is not large enough to handle your workload, SAP HANA generates an alert message.

During an out-of-buffer situation, regular user tasks may be rolled back, and critical tasks (for example, recovery) use the emergency buffer pool.

The following is an example of an out-of-buffers event:

```
9/5/18 3:09 PM At 2018-09-15 14:58:14 on lssjc0064:30003;
Buffer Cache 'CS(buffer_cache_CS)' with max_size 1 MB ran out of buffers
```

You can see out-of-buffer events in the indexserver trace file or in the mornitoring views, using the SQL Console.

View Out-of-Buffer Events in the Trace Files

The event, called "OutOfBuffersEvent," leaves a trace in the indexserver trace file. For example:

```
EventManagerImpl.cpp(00689) : New event reported:
'OutOfBuffersEvent[id= 1, "Buffer Cache 'CS(buffer_cache_cs)'
with max_size 1  MB ran out of buffers", state= NEW, acknowledged= false]'
```

The OutOfBuffersEvent event is triggered when a task requesting for a page to be loaded or allocated finds that the buffer cache has run out of buffers, causing SAP HANA to throw this error:

```
Error ID : 3020054
Error Text : CS cannot provide requested memory as its limit of <N> MByte is
exceeded.
```

This example shows the error message added to the indexserver trace file:

```
[55451]{-1}[723/295150499] 2018-08-29 00:41:12.077284 e attributes
PagedAttributeDef.h(03645) : mergeOldIntoNew : merge exception for paged
attribute
SAPBS4:ACDOCA.MAT_PSPNR: exception 1: no.3020054
(NDS/BufferCache/impl/BufferCache.cpp:314)
 CS cannot provide requested memory as its limit of 10240 MByte is exceeded
```

Resolve out-of-buffer events by increasing the maximum size for the corresponding buffer cache.

To do this, use ALTER SYSTEM ALTER CONFIGURATION, like in this example, which configures the maximum size of the buffer cache to 1000000 MB:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system')
   SET ('buffer_cache_cs','max_size') = '1000000
   [with reconfigure]
```

> **Note**
>
> After the last out-of-buffer error is reported, if the server notices that a number of subsequent buffer requests have been successfully fulfilled by the buffer cache without raising another error, the server then auto-resolves the out-of-buffers event.

View Out-of-Buffer Events in the SQL Console

You can monitor current out-of-buffer events in M_EVENTS by querying the following command:

```
SELECT * FROM M_EVENTS where ID = 133;
```

Old out-of-buffer events, however, are archived in a history as out-of-buffer alerts. You can monitor out-of-buffer alerts in STATISTICS_ALERTS by querying the following command:

```
SELECT * FROM _SYS_STATISTICS.STATISTICS_ALERTS WHERE ALERT_ID = 133;
```

To monitor out-of-buffer events or see the archived history of out-of-buffer cache alerts using HANA Cockpit, refer to *Monitor Out-of-Buffer Events*.

Related Information

Adjusting the Buffer Cache Size [page 219]
Monitor Out of Buffer Events
M_EVENTS System View
STATISTICS_ALERTS View (Embedded Statistics Service)

8.10.1.7 SAP HANA NSE and Paged Large Object Data Types

The large object data types (LOBs) such as binary large object (BLOB), character large object (CLOB), and national character large object (NCLOB) are used to store large amounts of data.

The data vector and the portion of LOBs stored in the LOB dictionary are paged. LOB data columns can be configured as either page loadable or column loadable by setting the LOAD UNIT value in the commands CREATE TABLE and ALTER TABLE. However, memory usage for LOB data types is controlled by the Hybrid LOB configuration parameters independent of the LOAD UNIT configuration. While the LOAD UNIT can be configured to either COLUMN LOADABLE or PAGE LOADABLE for a LOB data column, it does not determine the memory usage for LOB data types.

Related Information

Hybrid LOBs (Large Objects) [page 297]

8.10.1.8 Data Compression in SAP HANA Native Storage Extension

SAP HANA NSE supports all types of columnar compression techniques, including dictionary and advanced compression.

Related Information

Data Compression in the Column Store [page 313]

8.10.1.9 Understanding the NSE Advisor

Use the Native Storage Extension Advisor (NSE Advisor) to get suggestions about load units for tables, partitions, or columns according to how frequently they are accessed. The NSE Advisor provides a flexible interface to act on individual recommended objects to convert from column loadable to page loadable, or vice versa.

> **Note**
>
> You must have the `CATALOG READ` or `DATA ADMIN` role in order to get recommendations from the NSE Advisor. You must have the `TABLE ADMIN` or the `PARTITION ADMIN` system privilege on the corresponding tables to apply the recommendations.

The NSE Advisor determines the temperature of the data based on how frequently it is accessed at the query level for each column, and uses rule-based heuristics to identify hot and warm objects (that is, columns, partitions, and tables) as candidates to be page- or column-loadable. The NSE Advisor allows you to provide adjustable thresholds for warm (rarely accessed, stored on disk) and hot (frequently accessed, stored in-memory) objects.

Generally, the NSE Advisor recommends that you change the load unit of a table, partition, or a column to:

- Page-loadable to reduce memory footprint without much performance impact.
- Column-loadable to improve performance by keeping hot objects in memory.

The NSE Advisor runs a rule-based algorithm on statistical information that provides recommendations based on the counters it collects during the scan.

Running the NSE Advisor results in a list of tables, partitions, and columns with suggestions of whether they should be page- or column-loadable. To apply the recommendations see the section `<load_unit>` in ALTER TABLE Statement (Data Definition)

You can run the NSE Advisor in the following ways:

- Using the SQL interface on SAP HANA Platform. See Using the NSE Advisor with SQL Commands (INI Configurations) for Warm Data Recommendations [page 232], Using the NSE Advisor with SQL Commands (Built-in Procedure) for Warm Data Recommendations [page 227] and the SAP HANA SQL Reference Guide.

- Using the SAP HANA Cockpit. See Using the NSE Advisor for Warm Data Recommendations in the *SAP HANA Administration with SAP HANA Cockpit Guide*.

8.10.1.9.1 How the NSE Advisor Uses the Access Statistics Cache for Recommendations

SAP HANA uses heuristics to determine which objects (tables, partitions, or columns) are suitable for conversion to page loadable - to save memory space - or column loadable - to improve performance by keeping hot objects in memory.

Achieving the best cost and performance ratio requires that you know which objects can be made page loadable, reducing the in-memory foot print without significantly affecting performance.

Each workload has its own data access pattern:

- Objects that are small and accessed frequently should be kept in memory to improve performance.
- Objects that are large and rarely accessed are better kept on disk to reduce memory usage. These objects are brought in memory only when needed, and only for the pages that are accessed.

SAP HANA performs this task in two phases: the first phase builds the data access pattern given a specific workload, and the second phase applies a heuristic rule on the data access statistics to generate the recommendations for which objects should be converted to which load unit.

To build the data access pattern for a workload, each physical access to a column fragment is counted and updated at query level in an internal cache called the access statistics cache. Because only the main fragment is paged for a page loadable column, the access count is recorded only for access to the main fragment. The access count for the original main fragment is retained when the delta fragment is merged to the main fragment. When a column is unloaded from memory during the workload, its access count is also retained in the access statistics cache and continues to be updated after the column is loaded again in memory.

DDLs that drop an object clean up the corresponding statistics entries related to the dropped objects. Building the data access statistics has a small impact to the ongoing workload performance, so it is only enabled through a configuration option. Contents of the statistics cache can be viewed through a monitor view. The cache can be cleared when not needed or when a new cache needs to be built for a different workload.

SAP HANA uses heuristics to generate load unit recommendations from the access statistics cache, including:

- Scan density – the object access count and object memory size ratio (the object can be a table, a partition, or a column).
- Hot object threshold – the minimum scan density for an object to be considered a hot object. This is derived from a parameter for percentage of objects to be considered hot.
- Cold object threshold – the maximum scan density for an object to be considered a cold object. This is derived from a parameter for percentage of objects to be considered cold.
- Object size threshold – the minimum object size to be considered for recommendation. This is derived from a parameter for minimum object size.

A list of objects is compiled with recommendations to convert their load unit based on their scan density values. The object can be either a table, or a partition, or a column.

SAP HANA recommends that the object be page loadable if the object's:

```
Scan_Density < Cold Object Threshold and Object Memory Size > Object Size
Threshold
```

SAP HANA recommends that the object be column loadable if the object's:

```
Scan_Density > Hot Object Threshold and Object Memory Size > Object Size
Threshold
```

SAP HANA may recommend that a table be page loadable due to an overall low scan density while a column within the table is recommended to be column loadable due to a high scan density for that particular column. In that case, the column level recommendation takes precedence over the table level recommendation. Similarly, partition level recommendation takes precedence over the table level recommendation.

Use the NSE Advisor to view the recommendations.

8.10.1.9.2 Using the NSE Advisor with SQL Commands (Built-in Procedure) for Warm Data Recommendations

Get column, partition, and table load-unit recommendations through SQL commands (built-in procedure) for SAP HANA Native Storage Extension (NSE). The SAP HANA NSE Advisor tool (NSE Advisor) determines the temperature of data, and uses rule-based (data temperature threshold, access pattern/frequency, data density) algorithms to derive these recommendations, in particular to identify hot and warm objects as candidates to be either page-loadable or column-loadable.

Prerequisites

> **Note**
>
> Do not enable/disable the NSE Advisor with INI configurations and use the built-in procedure simultaneously as this leads to incorrect recommendations. If the built-in procedure is used, we recommend disabling the NSE Advisor after completion with the statement:
>
> ```
> CALL SYS.CS_NSE_ADVISOR(ACTION=>'DISABLE', REQUEST_ID=>?);
> ```

- You need either the `CATALOG READ` or `DATA ADMIN` system privilege to execute the SQL statements used in this task.
- The recommendations monitor view applicable for this task is `M_CS_NSE_ADVISOR`.
- The recommendations details monitor view applicable for this task is `M_CS_NSE_ADVISOR_DETAILS`.
- Use the built-in procedure CS_NSE_ADVISOR.
  This procedure's syntax is as follows:
  IN action VARCHAR(32),
  'ENABLE' | 'DISABLE' | 'RESET'
  IN hot_object_threshold TINYINT DEFAULT NULL,
  Minimum Scan Density Threshold

IN cold_object_threshold TINYINT DEFAULT NULL,
Maximum Scan Density Threshold
IN min_object_size BIGINT DEFAULT NULL,
Filter out objects below this size from NSE Advisor Consideration
IN duration BIGINT DEFAULT -1,
Duration in seconds for automatic disable and clean up. If duration is specified as -1, the NSE Advisor runs until disabled.
OUT request_id INT
Returns the request ID for reference if the NSE Advisor run in the new M_CS_NSE_ADVISOR_STATISTICS table.

> **Note**
>
> Ensure that the built-in procedure is not started multiple times in parallel. Execute the following command to RESET the built-in procedure for the NSE Advisor if the procedure returns an error message or you added/removed one or more HANA indexserver in a scale-out enviroment:
>
> ```
> CALL SYS.CS_NSE_ADVISOR(ACTION=>'RESET', REQUEST_ID=>?)
> ```

- The table below gives information on specific SQL statements you can use in this task:

SQL Statement	Privilege Required	Applicable SAP HANA Version and Revision
`ALTER SYSTEM ALTER CONFIGURATION`	`INIFILE ADMIN`	*No specific version/revision recommendation*

Context

Get load unit recommendations for NSE from the NSE Advisor with the built-in procedure for the following use cases:

- Starting the bult-in procedure for a defined duration (first procedure).
- Starting the bult-in procedure without duration (second procedure).

Procedure

1. Disable any NSE Advisor using INI configuration and disable any exiting built-in NSE Advisor:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
   ('cs_access_statistics','collection_enabled') = 'false' WITH RECONFIGURE
   CALL SYS.CS_NSE_ADVISOR(ACTION=>'DISABLE', REQUEST_ID=>?)
   ```

2. Adjust the configuration for collecting access statistics *(Optional)*

Enlarge the configuration for `'cs_access_statistics','min_row_count'` to limit the cache entry count in the Access Stats Cache as follows if you met performance issues when opening the NSE advisor data collection:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
('cs_access_statistics','min_row_count') = '10000' WITH RECONFIGURE;
```

> ### Note
>
> `min_row_count` controls the minimum row count for a table, or a partition of a partitioned table, to be considered for warm recommendations. Small tables, or partitions, with a lower row count than `min_row_count` will not insert cache entries into the Access Stats Cache and thus, will not receive recommendations from the NSE Advisor. The default value for `min_row_count` is 10,000. Also, if you met performance issues when starting the NSE Advisor data collection, change this configuration and follow the instructions in Step 6 to do a clean-up, then reuse NSE Advisor from Step 1.

3. Start the built-in procedure for the NSE Advisor:

```
CALL SYS.CS_NSE_ADVISOR(ACTION=>'ENABLE', HOT_OBJECT_THRESHOLD=>'30',
COLD_OBJECT_THRESHOLD=>'30', MIN_OBJECT_SIZE=>'1048576', DURATION=>'18000',
REQUEST_ID=>?)
```

   This SQL command uses a background thread to clean up `'cs_access_statistics'`, open NSE Advisor data collection for 18000 seconds, generate NSE advices, close NSE Advisor data collection, and clean up `'cs_access_statistics'` automatically.

   Where:

   - `hot_object_threshold`: Controls the percentage of objects which have been queried in the workload according to `scan_density`(`access_count/size`) of the object, to be considered as hot object in recommendation calculation.
   - `cold_object_threshold`: Controls the percentage of objects which have been queried in the workload according to `scan_density`(`access_count/size`) of the object, to be considered as cold object in recommendation calculation.
   - `min_object_size`: Controls the minimum object size for an object to be considered for the recommendations.
   - `duration`: Controls how many seconds NSE Advisor opens data collection. The default value is -1, which is used in the next procedure.
   - `request_id`: Identity for statistics information in M_CS_NSE_ADVISOR_STATISTICS.

4. Query M_JOB_PROGRESS/M_CS_NSE_ADVISOR_STATISTICS to see progress before `duration` is reached. One or multiple entries appear (the number is equal to indexserver count in the scale-out environment) for build-in NSE Advisor with `request_id` received in step 3.

   > ### Note
   >
   > Ensure the value reaches 200 in the column CURRENT_PROGRESS. This means that all steps, including the auto apply steps, have been completed.

5. Depending on your defined recommendation rules, you can now review the recommendations and modify corresponding tables, columns, and partitions. If you selected the advice mode `AUTO_APPLY`, the recommendations will be automatically applied, but if you selected `ANY`, you will have to change the

load unit manually. Get the recommendations or detailed information from the M_CS_NSE_ADVISOR and M_CS_NSE_ADVISOR_DETAILS views once the process is completed by running the following command:

```
SELECT * FROM M_CS_NSE_ADVISOR
SELECT * FROM M_CS_NSE_ADVISOR_DETAILS
```

> ### Note
>
> The best practice is to use `hot_object_threshold` and `cold_object_threshold` with a value of 30 for both, and to choose part of the recommendations to apply according to 'CONFIDENCE' column of M_CS_NSE_ADVISOR. This avoids tuning the same workload repeatedly.
>
> ```
> SELECT * FROM M_CS_NSE_ADVISOR ORDER BY CONFIDENCE DESC
> ```

6. Do a cleanup. Recommendations generated do not change before using the 'DISABLE' or 'RESET' action to clean it up:

   ```
   CALL SYS.CS_NSE_ADVISOR(ACTION=>'DISABLE', REQUEST_ID=>?)
   ```

7. If you want to view the history of recommendations, you can refer to `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY`. For more information, refer to *Viewing the History of Recommendations*.

Second Procedure

Procedure

1. Disable any NSE Advisor using INI configuration and disable any running built-in NSE Advisor.

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
   ('cs_access_statistics','collection_enabled') = 'false' WITH RECONFIGURE
   CALL SYS.CS_NSE_ADVISOR(ACTION=>'DISABLE', REQUEST_ID=>?)
   ```

2. Adjust the configuration for collecting access statistics *(Optional)*

   Adjust the configuration for `'cs_access_statistics','min_row_count'` to limit the cache entry count in the Access Stats Cache as follows if you met performance issue when open NSE advisor data collection or if you see a timeout error while querying M_CS_NSE_ADVISOR:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
   ('cs_access_statistics','min_row_count') = '10000' WITH RECONFIGURE;
   ```

   > ### Note
   >
   > `min_row_count` controls the minimum row count for a table, or a partition of a partitioned table, to be considered for warm recommendations. This means small tables, or partitions with a lower row count than `min_row_count` will not insert cache entries into the Access Stats Cache and thus, will not be advised by the NSE Advisor. The default value for `min_row_count` is 10,000. Also, if you met performance issue when open NSE advisor data collection, change this configuration and follow the instructions in Step 6 to do a clean-up, then reuse NSE Advisor from Step 1.

3. Start the built-in procedure for the NSE Advisor:

```
CALL SYS.CS_NSE_ADVISOR(ACTION=>'ENABLE', REQUEST_ID=>?)
```

This SQL command uses a background thread to clean up `'cs_access_statistics'` and open NSE Advisor data collection.

> **Note**
>
> - This SQL command runs with default configuration settings for the whole duration (-1).
> - This skips any other parameters used in the built-in procedure but use INI configuration for `hot_object_threshold`, `cold_object_threshold` and `min_object_size`.

4. Get the recommendations or detailed information from the M_CS_NSE_ADVISOR and M_CS_NSE_ADVISOR_DETAILS views at any time. Run the following command:

```
SELECT * FROM M_CS_NSE_ADVISOR
SELECT * FROM M_CS_NSE_ADVISOR_DETAILS
```

5. Review the recommendations and modify corresponding tables, columns, and partitions to page-loadable or column-loadable *(Optional)*. If the output is empty, you might need to fine-tune threshold values as explained below, and then again check the recommendations on the `M_CS_NSE_ADVISOR` view. The hot and cold object threshold values (10) and the minimum object size (999) in the following commands are used as examples:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system') SET
('cs_nse_advisor','hot_object_threshold_rel') = '10' WITH RECONFIGURE;
```

The default threshold value is 10% for hot objects.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system') SET
('cs_nse_advisor','cold_object_threshold_rel') = '10' WITH RECONFIGURE;
```

The default threshold value is 10% for cold objects.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET
('cs_nse_advisor','min_object_size') = '999' WITH RECONFIGURE;
```

The default minimum object size is 1048576 bytes.

Where:

- `hot_object_threshold_rel`: Controls the percentage of objects, according to row scan count and size, to consider when calculating hot object threshold.
- `cold_object_threshold_rel`: Controls the percentage of objects, according to row scan count and size, to consider when calculating cold object threshold.
- `min_object_size`: Controls the minimum object size for an object to be considered for the recommendations.

6. Do a cleanup. Recommendations generated do not change before using the 'DISABLE' or 'RESET' action to clean it up:

```
CALL SYS.CS_NSE_ADVISOR(ACTION=>'DISABLE', REQUEST_ID=>?)
```

7. If you want to view the history of recommendations, you can refer to `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY`. For more information, refer to *Viewing the History of Recommendations.*

Related Information

Use the NSE Advisor for Warm Data Recommendations
M_CS_NSE_ADVISOR System View
M_CS_NSE_ADVISOR_STATISTICS System View
M_CS_NSE_ADVISOR_DETAILS System View
Viewing the History of Recommendations [page 240]

8.10.1.9.3 Using the NSE Advisor with SQL Commands (INI Configurations) for Warm Data Recommendations

Get column, partition and table load-unit recommendations through SQL commands for SAP HANA Native Storage Extension (NSE). The SAP HANA NSE Advisor tool (NSE Advisor) determines the temperature of data, and uses rule-based (data temperature threshold, access pattern/frequency, data density) algorithms to derive these recommendations, in particular to identify hot and warm objects as candidates to be either page-loadable or column-loadable.

Prerequisites

You need either the `CATALOG READ` or `DATA ADMIN` system privilege to execute the SQL statements used in this task.

> **Note**
>
> Do not use the NSE Advisor with INI configurations and built-in procedure simultaneously as this leads to incorrect recommendations.

Additionally, refer to the table below for information on executing specific SQL statements in this task:

SQL Statement	Privilege Required	Applicable SAP HANA Version and Revision
`ALTER SYSTEM CLEAR CACHE`	`DEVELOPMENT`	SAP HANA 2.0 SPS 05, Rev 2.00.052 and lower
	`RESOURCE ADMIN`	SAP HANA 2.0 SPS 05, Rev 2.00.053 and higher
`ALTER SYSTEM ALTER CONFIGURATION`	`INIFILE ADMIN`	*No specific version/revision recommendation*

The recommendations monitor view applicable for this task is `M_CS_NSE_ADVISOR`. The recommendation details monitor view applicable for this task is `M_CS_NSE_ADVISOR_DETAILS`.

Context

To get load unit recommendations for NSE from the NSE Advisor, do the following:

Procedure

1. Clean up all the cache entries in the Access Stats Cache. This step ensures that `cs_access_statistics` cache does not contain any junk information in its last sample.

   ```
   ALTER SYSTEM CLEAR CACHE ('cs_access_statistics');
   ```

2. *(Optional) (Applies to SAP HANA 2.0 SPS 05, Revisions greater than or equal to* **2.00.053***)*

   Enlarge the configuration for `'cs_access_statistics','min_row_count'` to limit the cache entry count in the Access Stats Cache as follows if you see a timeout error while querying `M_CS_NSE_ADVISOR` in Step 4:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
   ('cs_access_statistics','min_row_count') = '10000' WITH RECONFIGURE;
   ```

   > ### Note
   >
   > `min_row_count` controls the minimum row count for a table, or a partition of a partitioned table, to be considered for warm recommendations. This means small tables, or partitions, with a lower row count than `min_row_count` will not insert cache entries into the Access Stats Cache and thus, will not be advised by the NSE Advisor. The default value for `min_row_count` is 10,000. Also, if you see a timeout error while querying `M_CS_NSE_ADVISOR` in Step 4, follow the instructions in Step 6 to do a clean-up, then reuse NSE Advisor from Step 1.

3. Enable CS Access Stats Collection and run the workload:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
   ('cs_access_statistics','collection_enabled') = 'true' WITH RECONFIGURE;
   ```

4. Once the workload has been running for a couple of hours, you can then get the recommendations or detailed information from the `M_CS_NSE_ADVISOR` and `M_CS_NSE_ADVISOR_DETAILS` views. Run the following command:

   ```
   SELECT * FROM M_CS_NSE_ADVISOR
   SELECT * FROM M_CS_NSE_ADVISOR_DETAILS
   ```

5. Depending on your defined recommendation rules, you can now review the recommendations and modify corresponding tables, columns, and partitions. If the output is empty, you might need to fine-tune threshold values as explained below, and then again check the recommendations and details on the

`M_CS_NSE_ADVISOR` and `M_CS_NSE_ADVISOR_DETAILS` views. The hot and cold object threshold values (10) and the minimum object size (999) in the following commands are used as examples:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system') SET
('cs_nse_advisor','hot_object_threshold_rel') = '10' WITH RECONFIGURE;
```

The default threshold value is 10% for hot objects.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system') SET
('cs_nse_advisor','cold_object_threshold_rel') = '10' WITH RECONFIGURE;
```

The default threshold value is 10% for cold objects.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET
('cs_nse_advisor','min_object_size') = '999' WITH RECONFIGURE;
```

The default minimum object size is 1048576 bytes.

Where:

- `hot_object_threshold_rel`: Controls the percentage of objects which have been queried in the workload, according to SCAN_DENSITY(ACCESS_COUNT/SIZE) of the object, to be considered as hot object in recommendation calculation.
- `cold_object_threshold_rel`: Controls the percentage of objects which have been queried in the workload, according to SCAN_DENSITY(ACCESS_COUNT/SIZE) of the object, to be considered as cold object in recommendation calculation.
- `min_object_size`: Controls the minimum object size for an object to be considered for the recommendations.

6. Disable CS Access Stats Collection. By default, an enabled CS Access Stats Collection might consume some extra memory. It is recommended that you disable CS Access Stats Collection and clear CS Access Stats cache to save memory. Run the following command:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','system') SET
('cs_access_statistics','collection_enabled') = 'false' [with reconfigure]
ALTER SYSTEM CLEAR CACHE ('cs_access_statistics');
```

7. If you want to view the history of recommendations, you can refer to `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY`. For more information, refer to *Viewing the History of Recommendations*.

Related Information

Use the NSE Advisor for Warm Data Recommendations
ALTER SYSTEM CLEAR CACHE Statement (System Management)
ALTER SYSTEM ALTER CONFIGURATION Statement (System Management)
Viewing the History of Recommendations [page 240]

8.10.1.9.4 Recommendation Rules

By default, the NSE advisor makes recommendations to all tables in the schema. You can narrow down the scope of the NSE advisor by defining rules in the table `_SYS_ADVISRO.NSE`.

Permissions

By default, the `_SYS_ADVISOR.NSE` schema and table are readable and writable for the SYSTEM user. The SYSTEM user must grant the appropriate privilege to allow other users to change rules:

```
GRANT INSERT/DELETE/UPDATE ON SCHEMA _SYS_ADVISOR TO <User>
```

Defining Rules

You can exclude items at any of the four levels, SCHEMA, TABLE, COLUMN or PARTITION. At the same time, you can define exceptions to these rules.

The recommendation rules table `_SYS_ADVISOR.NSE` has the following columns:

Column Name	Data Type	Description
SCHEMA_NAME	NVARCHAR(256)	Displays the schema name.
TABLE_NAME	NVARCHAR(256)	Displays the table name.
COLUMN_NAME	NVARCHAR(256)	Displays the column name.
PART_ID	INTEGER	Displays the partition ID number.
ADVICE_MODE	NVARCHAR(256)	It can be either: NONE: make no recommendations for items in this rule ANY: make recommendations for any (table, column, partition) matching value AUTO_APPLY: make recommendations and apply them automatically with the built-in procedure, with a duration longer than 0. For other configurations, it will be set to ANY.
WILDCARD	NCHAR	Can be used for schema and table names. Define a single character that will be used as a wildcard.

For more information on the `AUTO_APPLY` advice mode, refer to *Automated Setting of Load Unit*.

Recommendation rules are defined by adding entries into the `_SYS_ADVISOR.NSE` table. The following rule, for example, uses `ADVICE_MODE 'NONE'` to exclude all objects in schema ABC, except for table T1:

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('ABC', NULL, NULL, NULL, 'NONE', NULL);
INSERT INTO _SYS_ADVISOR.NSE VALUES ('ABC', 'T1', NULL, NULL, 'ANY', NULL);
```

To delete the previous rule, enter the following statement:

```
DELETE FROM _SYS_ADVISOR.NSE WHERE SCHEMA_NAME = 'ABC' AND TABLE_NAME IS NULL
AND COLUMN_NAME IS NULL AND PART_ID IS NULL;
```

> **Note**
>
> You cannot filter by partition and column in the same rule.

The sequence of the defined rules is important to follow when excluding individual items or whole schemas. There are four basic options, and the filtering process is applied in the following order, filtering first for partition or column names, then tables, and then schemas:

	Schema	Table	Column	Partition
1	Schema name	Table Name	NULL	Partition name
2	Schema name	Table Name	Column name	NULL
3	Schema name	Table Name	NULL	NULL
4	Schema name	NULL	NULL	NULL

The filtering process looks for objects in the database that match the objects defined in the rules. Matching items, unless they are included by a specific ANY rule, are excluded from the advisory process. Then, recommendations are generated for all other items.

Wildcards can be used in recommendation rules for schemas and tables. The usage of wildcards is illustrated in the following examples.

Examples

You do not want to get recommendations for schema 'S1':

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', NULL, NULL, NULL, 'NONE', NULL)
```

You do not want to get recommendations for objects in the schema 'S1' except for table 'T1':

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', NULL, NULL, NULL, 'NONE', NULL)
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', 'T1', NULL, NULL, 'ANY', NULL)
```

This example excludes partition 1 of table T2 in schema S2 from getting recommendations:

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S2', 'T2', NULL, 1, 'NONE', NULL)
```

Include recommendations for all items whose schema name starts with 'S' and the table name starts with 'NSE':

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S%', 'NSE%', NULL, NULL, 'ANY', '%')
```

Consistency, Conflict, and Syntax Checks

You should make sure that the set of defined rules is consistent, otherwise, it might lead to errors. There are three kinds of checks for recommendation rules: consistency checks, conflict checks and syntax checks.

- Consistency checks verify whether the specific schema, table, partition, or column in the filter rules exist in the SAP HANA system. If recommendation rules have consistency issues, an error message will appear: `There are inconsistent rules in _SYS_ADVISOR.NSE: {$rulenames$}, please fix it and try again.`
- Conflict checks verify whether both column and partition rules for a table are specified. When using wildcards, since the wildcard character is user-defined, it is possible that multiple wildcard characters are used at the same time. If rules conflict for any reason, they are ignored by the process and will lead to an error: `There are conflict rules in _SYS_ADVISOR.NSE for tables: {$tables$}, please fix it and try again.`
- Syntax checks apply to rules with wildcards and verify whether the syntax of the rule is right or not. If recommendation rules with wildcards have a syntax issue, an error message will appear:`Wrong syntax for wildcard rule: {$rulename$}.{$errmsg$}, please fix it and try again.`

If an object is excluded by the rules, the reason is given in the `SKIP_REASON` column of the view `M_CS_NSE_ADVISOR_DETAILS`.

Related Information

Viewing the History of Recommendations [page 240]
Automated Setting of Load Unit [page 237]
M_CS_NSE_ADVISOR_DETAILS System View

8.10.1.9.5 Automated Setting of Load Unit

Use the `AUTO_APPLY` advice mode to allow recommendations and let the NSE advisor apply them automatically.

In this mode, the matching object recommendations are allowed. The NSE advisor will use a background job to automatically apply the recommendations, namely, change the load unit, when using the built-in procedure `CS_NSE_ADVISOR` with a duration longer than 0.

> **Note**
>
> The `AUTO_APPLY` advice mode is only available with the built-in procedure `CS_NSE_ADVISOR` with a duration longer than 0. For other cases (including INI configurations or the built-in procedure

CS_NSE_ADVISOR with a duration equal to -1), AUTO_APPLY will be degraded to the advice mode ANY and you will have to manually change the load unit.

Examples

There are several combinations of schema, table, partition, or column auto apply rules.

If you want to get recommendations for schema 'S1' and automatically apply them, add the following rule:

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', NULL, NULL, NULL, 'AUTO_APPLY', NULL);
```

If you do not want to get recommendations for schema 'S1' but want to get recommendations for 'T1' and automatically apply them, add the following rule:

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', NULL, NULL, NULL, 'NONE', NULL)
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', 'T1', NULL, NULL, 'AUTO_APPLY', NULL);
```

You can also use wildcards. For example, if you want to get recommendations for tables starting with 'T' in schema 'S1' and automatically apply them, add the following rule:

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1', 'T%', NULL, NULL, 'AUTO_APPLY', '%');
```

If you do not want to get recommendations for tables in schema starting with 'S', but want to get recommendations for tables in a schema starting with 'S1' and automatically apply them, add the following rule:

```
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S%', NULL, NULL, NULL, 'NONE', '%')
INSERT INTO _SYS_ADVISOR.NSE VALUES ('S1%', NULL, NULL, NULL, 'AUTO_APPLY', '%');
```

Monitoring the Status of an AUTO_APPLY job

You can monitor the status of the automatically applied recommendations in M_CS_NSE_ADVISOR_STATISTICS. The CURRENT_PROGRESS column displays the estimated progress of the current run:

- From 0 to 100, it indicates that it is waiting for the duration and calculation steps.
- 99 indicates that the calculation of the advice is in progress.
- From 100 to 200, it indicates that the auto apply is in progress.
- 200 indicates all jobs have finished.

If any internal errors occur, this estimated progress doesn't change. It can be used to determine in which progress the error occurred.

You can refer to the columns AUTO_APPLY_TIME and AUTO_APPLY_STATUS in the views M_CS_NSE_ADVISOR and M_CS_NSE_ADVISOR_DETAILS, or in the table _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY to track more details.

You can track the current AUTO_APPLY job with `M_CONNECTIONS` and `M_PREPARED_STATEMENTS`:

```
SELECT * FROM M_CONNECTIONS WHERE CREATE_BY = “NSE Advisor”
SELECT * from M_PREPARED_STATEMENTS WHERE CONNECTION_ID = <connection ID from
previous SQL>
```

Use the DISABLE or RESET actions of the built-in procedure `CS_NSE_ADVISOR` to cancel all the pending `AUTO_APPLY` operations. This will not cancel any existing running jobs.

To cancel the existing running jobs, you can query the following command:

```
ALTER SYSTEM CANCEL [WORK IN] SESSION <connection_ID>
```

Limitations of the AUTO_APPLY mode

The NSE advisor is unable to automatically apply the recommendations generated for hash partitions of RANGE-HASH and RANGE-RANGE-HASH partition tables. If the `AUTO_APPLY` mode is set for those partitions, the system does not apply the recommendations and they will appear as FAILED. This leads to the following message in the trace files:

```
ERROR [SQL-7] feature not supported: Altering load units for hash partitions is
not allowed.
```

For more information on how to manually apply the recommendations and change the load unit, refer to *NSE-Enabled Partitioned Tables*.

Best Practices

Exclude performance-critical tables from automaticaly applying recommendations

Since the NSE advisor relies on data collection to do recommendations, if your workload changes, or the data collection phase does not cover your typical workload, you could get inaccurate advice. Thus, it is recommended to exclude performance critical tables from the `AUTO_APPLY` feature, that is, set the advice mode to `ANY` to ensure that the NSE advisor does not automatically apply the recommendations.

Set a duration time

You can freely schedule the data collection phase, but changing the load unit may temporarily reduce table performance. To minimize impact, schedule the data collection to start during a time that reflects a typical usage workload, and end it during a quiet period, such as the maintenance window for the SAP HANA database servers.

Avoid performance issues

If you face a performance issue, revert the load unit of the affected table. You can monitor the recommendations applied by the NSE advisor in the `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY` table. Then, create a specific rule (with `ANY` or `NONE` as the advice mode) to prevent the NSE advisor from automatically applying recommendations to the affected table.

Related Information

Recommendation Rules [page 235]
Viewing the History of Recommendations [page 240]
NSE-Enabled Partitioned Tables [page 210]
M_CS_NSE_ADVISOR_DETAILS System View
M_CS_NSE_ADVISOR_STATISTICS System View
M_CONNECTIONS System View
M_PREPARED_STATEMENTS System View

8.10.1.9.6 Viewing the History of Recommendations

View the history of NSE advisor recommendations in the_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY table.

The _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY displays the following:

- When using the built-in procedure CS_NSE_ADVISOR with a duration equal to -1 or INI configurations, a background job will be executed to insert the recommendations into the _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY table when querying M_CS_NSE_ADVISOR.
- When using the built-in procedure (CS_NSE_ADVISOR) with a duration longer than 0, the generated recommendations will be inserted into the _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY table as part of a background job. In this case, querying M_CS_NSE_ADVISOR will only output in-memory recommendations, thus will not lead to another insertion into _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY.
- Querying M_CS_NSE_ADVISOR_DETAILS will not insert any recommendations into _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY.

The table is defined as a COLUMN table, marked as PAGE_LOADABLE, and automatically updated by SAP HANA.

_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY has the following columns:

Column name	Data Type	Description
REQUEST_ID	INTEGER	Displays the request ID of the NSE advisor. It's the same as the returned request ID of the built-in procedure CS_NSE_ADVISOR. Value is 0 for recommendations generated with INI configurations.
GENERATE_TIME	TIMESTAMP	Displays the time when a record was inserted into _SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY. Values are NOT NULL, and the default is 0.
HOST	VARCHAR(64)	Displays the host name that generated the recommendation.
PORT	INTEGER	Displays the internal port that generated the recommendation.

Column name	Data Type	Description
COLD_OBJECT_THRESHOLD	INTEGER	Displays the threshold of the cold object, as a percentage, used to generate recommendations.
HOT_OBJECT_THRESHOLD	INTEGER	Displays the threshold of the hot object, as a percentage, used to generate recommendations.
MIN_OBJECT_SIZE	BIGNIT	Displays the minimum size of the object, in bytes, used to generate recommendations.
DURATION	INTEGER	Displays the duration time, in seconds, of the data collection for this run of NSE advisor. -1 for not specified.
SCHEMA_NAME	NVARCHAR(256)	Displays the schema name of the object.
TABLE_NAME	NVARCHAR(256)	Displays the table name of the object.
TABLE_OID	BIGNIT	Displays the object ID of the table.
PART_ID	INTEGER	Displays the partition ID. When the recommendation level is not PARTITION, this field is not applicable and the value 0 is used.
COLUMN_NAME	NVARCHAR(256)	Displays the column name. When the recommendation level is not COLUMN, this field is not applicable and the value NULL is used.
COLUMN_ID	INTEGER	Displays the numeric column ID.
GRANULARITY	VARCHAR(10)	Displays the object level at which the recommendation for this table is given. Possible values are: TABLE, PARTITION, or COLUMN.
CONFIDENCE	INTEGER	Displays a score (range [0-100]) that indicates the confidence of the recommendation. The value can then be used to objectively compare recommendations of a single NSE Advisor run.
LOAD_UNIT	VARCHAR(7)	Displays the recommended load unit for this object. Possible values: COLUMN, PAGE.
MEMORY_SIZE_IN_MAIN	BIGNIT	Displays the memory consumption size, in bytes, per granularity of the object in main memory. If not LOADED, the value 0 is used.
MAIN_PHYSICAL_SIZE	BIGNIT	Displays the storage size, in bytes, per granularity of the object.
AUTO_APPLY_TIME	TIMESTAMP	Displays the timestamp when the AUTO_APPLY_STATUS was last set.

Column name	Data Type	Description
AUTO_APPLY_STATUS	VARCHAR(10)	Displays the status of auto apply. Values are: NONE, PENDING, SUCCEED, FAILED, INVALID or CANCELLED.

> **Note**
>
> By default, only `SYSTEM` users can see and modify the `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY` schema and table. The `SYSTEM` user, however, can allow other users to read history recommendations in `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY` by granting them privileges with SQL `'GRANT':`
>
> ```
> GRANT SELECT ON SCHEMA _SYS_ADVISOR TO <User>
> ```

Retention Time

SAP HANA automatically retains the records of `_SYS_ADVISOR.CS_NSE_ADVISOR_HISTORY`. SAP HANA uses a background garbage collection job to drop records older than a defined number of days by prompting the built-in procedure `CS_NSE_ADVSIOR`.

> **Note**
>
> If there is no call to the built-in procedure `CS_NSE_ADVSIOR`, no garbage collection job will be created.

You can use the following SQL command to change the retention time:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system') SET
('cs_nse_advisor',' history_retention_time') = '<user defined days>' WITH
RECONFIGURE;
```

Related Information

Recommendation Rules [page 235]

M_CS_NSE_ADVISOR_DETAILS System View
M_CS_NSE_ADVISOR System View

8.10.2 Extension Node

For scaled-out systems (SAP HANA native systems or in a SAP Business Warehouse) where a multi-temperature storage strategy is required, you can use the extension node feature to deploy a different type of host in the server landscape which is used exclusively for warm data.

Overview

The hardware sizing guidelines for data storage depend upon the type of data being stored. Hot data must be stored in SAP HANA memory for fast access, and the normal ratio of RAM to data storage in this case is 2:1, for example, a system with 8TB of RAM can store up to 4TB of hot data.

However, if the frequency of usage of data slows down, the data temperature can be classified as warm. Warm data cannot be removed from the primary database as it may still need to be changed, but the performance of select statements is less crucial than for hot data.

Consequently, the sizing requirements for nodes which exclusively contain warm data can be relaxed. For example, it is possible to double the amount of data on such a node, or even quadruple it as long as the sizing of the data volume allows this. Extension nodes still use the column store and need dynamic memory for query processing, so overloading the node in this way is only possible, if the amount of data which is used at any point in time, and the amount of required memory for query processing is within the level of available DRAM on this node. When considering to what extent an extension node can be overloaded, query and model design are extremely important to minimize the amount of warm data that has to be loaded into memory and avoid wasting valuable DRAM unnecessarily.

You can create an extension node in SAP HANA native systems or in SAP Business Warehouse. SAP BW offers powerful data modeling features which make it possible to separate warm data from hot data. An extension node can be implemented by purchasing new hardware or by reconfiguring an existing node in the SAP HANA landscape. After configuring the extension node and modeling the data correctly to categorize the data according to temperature you can redistribute data using landscape management or data distribution optimizer so that objects that are classified for warm data will be moved to the extension node.

Configuration

The extension node is configured as a slave node and has a host sub role (worker group value *worker_dt*) which identifies the special function of the node. Table placement settings and locations are stored in metadata table `SYS.TABLE_PLACEMENT_LOCATIONS`.

The configuration procedure is described in the following two SAP Notes:

- 2741690 - *How-To: Configuring SAP HANA Extension Nodes* (specific to HANA 2.0 SPS 04)
- 2415279 - *How-To: Configuring SAP HANA for the SAP HANA Extension Node* (relevant for HANA native scenarios)

These notes include example commands that are required to set up table placement in a SAP BW on HANA scenario, a SAP BW/4HANA scenario and also for HANA Native scenarios. After completing the configuration procedure you can verify the location of the extension node in the view `M_LANDSCAPE_HOST_CONFIGURATION` and the locations for table placement can be seen in `SYS:TABLE_PLACEMENT_LOCATIONS`.

Note that the worker group name given to the extension node is also used by the following processes where it is important that the warm data is handled correctly:

- Backup and recovery to ensure that the right data is restored to this server
- SAP HANA system replication
- Standby nodes.

The use of sub roles and placing data in this way is not restricted to extension nodes, it can also be used more generically, for example, to pin schemas or applications to specific nodes of the landscape.

Tools Available for Redistribution

Data objects which already exist in SAP Business Warehouse can be reclassified and moved to the extension node using either the landscape redistribution tool in SAP HANA or the data distribution optimizer (DDO).

In SAP Business Warehouse data objects for warm data are mainly write-optimized data store objects (DSO), advanced DSOs (aDSO) or persistent staging area (PSA) tables. In the case of advanced DSOs which contain hot and warm data and are split by partitions, use the data distribution optimizer tool to redistribute the data.

Related Information

Monitoring Host Status and Auto-Failover Configuration
Redistributing Tables in a Scaleout SAP HANA System [page 405]
SAP Note 2741690
SAP Note 2343647
SAP Note 2453736
SAP Note 2415279

8.11 Managing Tables

The SAP HANA database stores data in memory in tables, organized in columns, and partitions, distributed among multiple servers.

In addition to the row and column data storage types, the Document Store for JSON artifacts is fully integrated into the SAP HANA database architecture. The document collections of the Document Store are also classified as a type of table and they can be created, updated and read by SQL. The Document Store allows native operation on JSON, for example, filtering, aggregation, and joining JSON documents with HANA column or row store tables.

The default table type is column-type tables. If the table type in a CREATE TABLE statement is not explicitly stated then the table will automatically be a column table. It is possible to override this behavior by setting the configuration parameter `default_table_type` in the `sql` section of the indexserver.ini file. The default table type for temporary tables is row-type.

Federation (Smart Data Access)

Users may have requirements to access data in remote systems, this can be done either by table replication or by using smart data access to define virtual tables. Virtual tables are placeholders pointing to remote data which can be addressed like local tables and thereby avoid the need to store redundant data copies. For details, refer to the Data Access and SAP HANA Smart Data Access sections of this document.

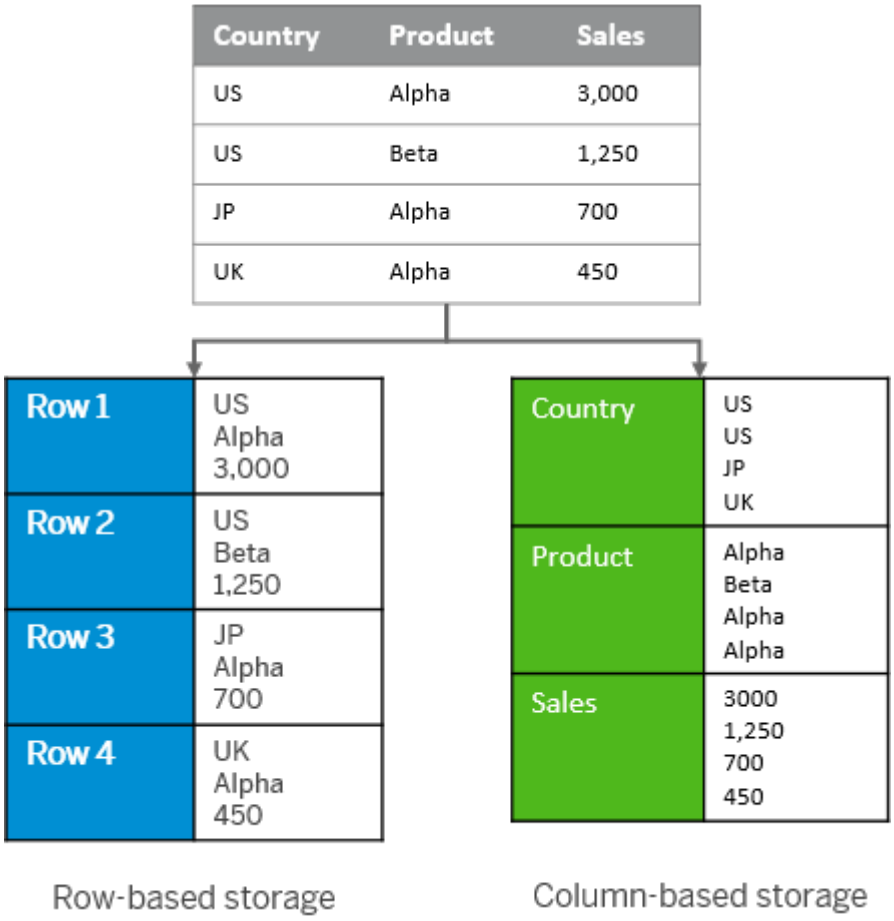
Related Information

Columnar and Row-Based Data Storage [page 245]
The JSON Document Store [page 258]
Basic Table Management in SAP HANA Cockpit [page 259]
Basic Table Management in SAP HANA Studio [page 260]
Table and Catalog Consistency Checks [page 282]
Memory Management in the Column Store [page 292]
The Delta Merge Operation [page 301]
Data Compression in the Column Store [page 313]
Extension Node [page 243]
Table Partitioning [page 317]
Table Replication [page 380]
Table Placement [page 370]
Redistributing Tables in a Scaleout SAP HANA System [page 405]
Data Access [page 1518]

8.11.1 Columnar and Row-Based Data Storage

The SAP HANA database supports two types of table: those that store data either column-wise (column tables) or row-wise (row tables). SAP HANA is optimized for column storage and this is the default table type.

Conceptually, a database table is a two dimensional data structure with cells organized in rows and columns. Computer memory however is organized as a linear sequence. For storing a table in linear memory, two options can be chosen as shown below. A row store stores a sequence of records that contains the fields of one row in the table. In a column store, the entries of a column are stored in contiguous memory locations.



Principle of Row- and Column-Based Storage for a Table

In the SAP HANA database, tables that are organized in columns are optimized for high-performing read operations while still providing good performance for write operations. Efficient data compression is applied to save memory and speed up searches and calculations. Furthermore, some features of the SAP HANA database, such as partitioning, are available only for column tables. Column-based storage is typically suitable for big tables with bulk updates. However, update and insert performance is better on row tables. Row-based storage is typically suitable for small tables with frequent single updates.

The following table outlines the criteria that you can use to decide whether to store your data tables as column tables or row tables:

Storage Type	When to Use
Column store	• Calculations are typically executed on individual or a small number of columns. • The table is searched based on the values of a few columns. • The table has a large number of columns. • The table has a large number of rows and columnar operations are required (aggregate, scan, and so on) • High compression rates can be achieved because the majority of the columns contain only a few distinct values (compared to the number of rows).

Storage Type	When to Use
Row store	• The application needs to process only one single record at one time (many selects and /or updates of single records). • The application typically needs to access the complete record. • The columns contain mainly distinct values so compression rate would be low. • Neither aggregations nor fast searching are required. • The table has a small number of rows (for example, configuration tables).

> **Note**
>
> - The SAP HANA database allows row tables to be joined with column tables. However, it is more efficient to join tables of the same storage type.
> - It is possible to change an existing table from one storage type to the other (`ALTER TABLE ALTER TYPE`).

Advantages of Column-Based Storage

Column tables have several advantages:

- Higher data compression rates
  Columnar data storage allows for highly efficient compression. Especially if the column is sorted, there will be ranges of the same values in contiguous memory, so compression methods such as run length encoding or cluster encoding can be used more effectively.
- Higher performance for column operations
  With columnar data organization, operations on single columns, such as searching or aggregations, can be implemented as loops over an array stored in contiguous memory locations. Such an operation has high spatial locality and efficiently utilizes the CPU caches.
  In addition, highly efficient data compression not only saves memory but also increases speed.
- Elimination of additional indexes
  In many cases, columnar data storage eliminates the need for additional index structures since storing data in columns already works like having a built-in index for each column: The column-scanning speed of the in-memory column store and the compression mechanisms (especially dictionary compression) already allow read operations with very high performance. In many cases, it will not be required to have additional index structures. Eliminating indexes reduces memory size, can improve write performance, and reduces development efforts. However, this does not mean that indexes are not used at all in SAP HANA. Primary key fields always have an index and it is possible to create additional indexes, if required. In addition, full text indexes are used to support full-text search.
- Elimination of materialized aggregates
  Thanks to its column-scanning speed, the column store makes it possible to calculate aggregates on large amounts of data on the fly with high performance. This eliminates the need for materialized aggregates in many cases. Eliminating materialized aggregates has several advantages. It simplifies data model and aggregation logic, which makes development and maintenance more efficient; it allows for a higher level of concurrency because write operations do not require exclusive locks for updating aggregated values; and it ensures that the aggregated values are always up-to-date (materialized aggregates are sometimes updated only at scheduled times).

- Parallelization
  Column-based storage also simplifies parallel execution using multiple processor cores. In a column store data is already vertically partitioned. That means operations on different columns can easily be processed in parallel.

Related Information

ALTER TABLE Statement (Data Definition)

8.11.2 Managing Indexes

For multi-column indexes in the column store you can set a default index type.

Index Types

When you create an index on a table the index type is an optional value. If you do not specify a type then a type is provided automatically depending on the type of table and the data type of the column. For a single column index on a column store table, for example, an INVERTED VALUE index is created if no type is specified. There are three types of index:

INVERTED VALUE

The index-type INVERTED VALUE can be used for indexes on column store tables. For INVERTED VALUE indexes an internal column is created, which is utilized by queries that include some or all of the indexed columns in the where-clause. With INVERTED VALUE the queries on indexed columns provide maximum performance. On the other hand the internal column adds memory and IO overhead, which may be significant in case of large tables.

INVERTED INDIVIDUAL

An INVERTED INDIVIDUAL index is lightweight index type with reduced memory footprint. The name INVERTED INDIVIDUAL reflects that internally only inverted indexes on individual columns are created (that is, no internal column is created). This type of index can only be used for column store tables, and is only available for composite unique constraints, which may be defined as secondary unique index or primary key. Query performance for INVERTED INDIVIDUAL indexes depends on data distribution and is not as predictable as for INVERTED VALUE.

INVERTED HASH

> ⓘ Note
>
> Please note that non-unique INVERTED HASH indexes are still supported in SPS 06 but are now deprecated. In the future only UNIQUE INVERTED HASH indexes will be supported.

The INVERTED HASH index-type provides a compromise between memory-footprint and query performance. For INVERTED HASH indexes an internal concat-column with hash values is created. Compared to the INVERTED VALUE index-type the underlying hash-based data representation typically results in a significantly smaller memory footprint. HASH should not be used as a composite type in cases where range queries or similar queries on the composite keys are a significant part of the workload. In these cases, use INVERTED VALUE instead.

For more information refer to the CREATE INDEX section of the *SAP HANA SQL Reference Guide for SAP HANA Platform* and the following SAP Notes:

- SAP Note 2600076 FAQ: SAP HANA Inverted Individual Indexes
- SAP Note 2109355 How-To: Configuring SAP HANA Inverted Hash Indexes

Configuration

For multi-column indexes in the column store you can determine the default index type by using the configuration parameters described here. There is one parameter for unique constraint indexes and one parameter for primary key constraint indexes, they do not apply to single column indexes and non-unique indexes. The parameters set the default index type globally, it cannot be set for a specific schema.

Details of these parameters, which are in the [sql] section of the indexserver.ini file, are as follows:

Parameter	**cs_composite_primary_key_constraint_index_type**
Description	Use this parameter to set a default index type for primary constraint multi column indexes in the column store. The type defined here will be used if the DDL to create the index does not include an explicit index type.
Values	INVERTED VALUE|INVERTED INDIVIDUAL|INVERTED HASH
Default Cloud	INVERTED INDIVIDUAL
<Default On premise>	<INVERTED VALUE>
Restart required	No. Changes are effective immediately.

Parameter	**cs_composite_unique_constraint_index_type**
Description	Use this parameter to set a default index type for unique constraint multi column indexes in the column store. The type defined here will be used if the DDL to create the index does not include an explicit index type.
Values	INVERTED VALUE|INVERTED INDIVIDUAL|INVERTED HASH
Default Cloud	INVERTED INDIVIDUAL
<Default On premise>	<INVERTED VALUE>
Restart required	No. Changes are effective immediately.

Example

The following CREATE TABLE statement creates a multi-column primary key index for the table without specifying the index type:

```
CREATE COLUMN TABLE MYTAB (A INT, B INT, PRIMARY KEY(A, B));
```

The type will be determined by the primary_key_constraint parameter.

Related Information

SAP Note 2600076
SAP Note 2109355
CREATE INDEX Statement (Data Definition)

8.11.3 Locks

SAP HANA uses two lock types: database locks to ensure transaction consistency, and application locks to manage access to application resources.

Database locks, which are used to ensure transaction consistency, are mostly invisible to users and handled by the transaction manager. Application locks identify a lock by name and can be explicitly acquired and released by application developers to manage access to application resources; this makes it possible for applications to define custom locking strategies independent of standard database locks.

This topic gives general information about locks so administrators can analyze locks and transactions in detail when necessary; information about both kinds of lock is available in monitoring views.

Database Locks

Database locks control concurrent access to the database, prevent conflicts between concurrent transactions and ensure data consistency. SAP HANA uses locks on table level (object locks - can also be applied to partitions) and on row level (record locks). If two requests are submitted simultaneously the first transaction that successfully acquires the lock is executed. The lock is automatically released when the transaction is committed or when a given timeout value is exceeded. The default timeout of 30 minutes is defined in the configuration parameter lock_wait_timeout. If the transaction does not complete within the timeout period then the database returns an error and the transaction is rolled back (see Blocked Transactions below).

Deadlocks may occur where transactions compete for access to a record, but these are detected by the transaction manager which aborts one of the transactions.

Locks can be shared or exclusive:

- Shared locks can be granted if no transaction holds an exclusive lock for the same object.

- Exclusive locks require that no other transaction holds a lock for the same object (shared or exclusive).

In general, SAP HANA uses multi-version concurrency control (MVCC) to ensure consistent read operations. MVCC uses versioning when updating existing records so that a transaction may update a record while concurrent transactions still have read access to previous versions.

To prevent concurrent write operations on the same data record, the SAP HANA database uses exclusive write locks at row level. These are acquired implicitly by update and delete operations. For each write access except insert operations, a row-level write lock is obtained. Concurrent transactions which need to write the same record have to wait until the lock is released.

Insert operations do not require a row-level write lock. This is because SAP HANA uses Multi-Version Concurrency Control (MVCC). When a new row is inserted, it creates a new version of the record, which is only visible to the inserting transaction until it commits. Since the insert operation is creating new, uncommitted data, there is no conflict with existing records or concurrent transactions.

Table level locks are required to prevent data updates (DML requests) from interfering with changes to the table definition (DDL). The operations on the table content require a shared table lock, while changes to the table definition require an exclusive table lock.

Explicit Locking by Users

The process of locking tables and rows is normally not visible to users, however, SQL statements are available for users to manage locks explicitly at row and table (or partition) level:

- An application can explicitly request a row level lock with the *SELECT <for_update_clause>* statement which locks the selected records until the end of this transaction.
- You can explicitly lock a table or partition using the LOCK TABLE statement which includes a *<partition_id>* clause to identify a partition to lock.

Blocked Transactions

If a requested lock cannot be granted, the transaction blocks until the timeout is reached, the transaction is then rolled back and an SQL error is generated (Error 131: transaction rolled back by lock wait timeout). A blocked transaction may also generate an alert (Alert ID 49: 'Long-running blocking transaction' - the severity level is determined by configurable time thresholds).

Blocked transactions can be monitored in the view M_BLOCKED_TRANSACTIONS. A blocked transaction is also visible as a long-running thread and in SAP HANA cockpit you can use the Thread monitoring features to see details of these threads and transactions.

For more information refer to the topic 'Blocked Transactions' in the *SAP HANA Troubleshooting and Performance Analysis Guide*. An FAQ Knowledge Base Article provides further information: *1999998 FAQ: SAP HANA Lock Analysis*.

Monitoring Views

Details of database locks are available in the following monitoring views:

- M_OBJECT_LOCKS - includes the partition ID if the object is a partition.
- M_RECORD_LOCKS

Application Locks

An application lock is a synchronization mechanism used within an application to control access to a shared resource. Application locks have an owner and are identified by a lock name (an arbitrary user-defined value). No additional metadata is required for the lock. The lock can be used for example to prevent users from applying changes simultaneously:

- A user submits changes and the application applies a lock. If the application lock is successfully acquired the application applies the changes.
- Other users cannot submit changes until a lock can be acquired: if the requested lock is not acquired then a message is returned to the user who must retry later.

Two procedures are available to acquire and release locks:

- CALL ACQUIRE_APPLICATION_LOCK('<lock_name>', '<lock_mode>', '<lock_scope>', '<timeout>')
- CALL RELEASE_APPLICATION_LOCK('<lock_name>', '<lock_scope>')

These procedures support parameters to specify `mode` (EXCLUSIVE or SHARED), `scope` (TRANSACTION, SESSION) and a `timeout` value in milliseconds. The scope of the lock determines when it is released and a lock may be applied at both scope levels: transaction locks are released when the transaction is committed or rolled back, session locks are maintained for the length of a session.

Details of these procedures and examples of usage are available in SAP Note 3432435 *Built-In Procedures for Managing Application Locks*.

Monitoring Views

Details of application locks are available in the following monitoring views:

- M_APPLICATION_LOCKS shows the status of currently acquired application locks.
- M_TRANSACTIONS shows the count of acquired application locks for a given transaction (column ACQUIRED_APPLICATION_LOCK_COUNT).
- M_BLOCKED_TRANSACTIONS shows the name of the application lock on which the lock is currently placed (column WAITING_APPLICATION_LOCK).

Related Information

Blocked Transactions (SAP HANA Troubleshooting and Performance Analysis Guide)
Threads (SAP HANA cockpit)
LOCK TABLE Statement (SQL Reference Guide)
M_OBJECT_LOCKS System View
M_RECORD_LOCKS System View
SAP Note 3432435
SAP Note 1999998

8.11.4 Temporal Tables

Temporal tables is a general term for tables which provide functionality to manage historical data: that is, to maintain a set of current values which are valid for a predefined period and also maintain an auditable continuous history of changes.

History Tables were introduced in SAP HANA 1.0 to maintain a history of previous versions of each record and to support time-based queries. SAP History Tables, however, are limited principally because the validity periods are based on system values such as database commit IDs which cannot be maintained or managed by applications.

System-Versioned Tables were introduced in SAP HANA 2.0 SPS 03 to address this, making the validity periods accessible to SQL. For system-versioned tables the data is split across two tables: a current data table and a corresponding history table. They support the tracking of changes on column store tables by capturing the validity period of each record, it is then possible to read ranges of historical data and to query in 'time-travel' mode using the AS OF + <time-stamp> syntax.

Many applications, however, need to be able to manage and manipulate historical business data independent of system time-stamps, for example, to define the validity period of a stock unit or the duration of a customer contract. Application-Time Period Tables have been introduced in SAP HANA 2.0 SPS 04 for this purpose so that records in the database can also have an application-specific validity period and applications can perform time-based queries on business data with optimized access paths.

> **Note**
>
> SAP history tables are still available in the SAP HANA product but you are recommended to use system-versioned and application-time period tables in preference to history tables.

8.11.4.1 SAP HANA History Tables

SAP HANA supports column-based history tables which allow queries on historical data (also known as time-based queries).

History tables are special database tables that only allow inserts. Write operations on history tables do not physically overwrite existing records, instead, they always insert new versions of the data record into the database. The most recent versions in history tables are called current data. All other versions of the same data object contain historical data. Each row in a history table has timestamp-like system attributes that indicate the time period when the record version in this row was the current one. Historical data can be read from history tables by requesting the execution of a time-based query against a historical view of the database (`SELECT ... AS OF` *time*). Alternatively, you can put a database session in history mode, so that all subsequent queries are processed against the historical view.

The history tables in SAP HANA correspond to system-versioned temporary data in SQL. For system-versioned temporal data the timestamps are automatically set and indicate the transaction time when the data was current. New versions are automatically created by the system during updates.

System-versioned tables and application-time period tables have been introduced in more recent versions of SAP HANA and these should now be used in preference to history tables. Refer to the *SAP HANA SQL and System Views Reference* for more details.

Related Information

CREATE TABLE
System-Versioned Tables [page 254]
Application-Time Period Tables [page 256]

8.11.4.2 System-Versioned Tables

System-versioned tables are part of the SQL standard. They support the tracking of changes on column store tables by capturing the validity period of each record.

Specifically in SAP HANA, each record has a *Valid from* and *Valid to* value; these values are maintained by the system so that the mechanism is invisible to the user.

System-versioned tables always consist of two physical tables:

1. The main table of records that are currently valid.
2. A corresponding history table (a one-to-one correspondence) of archived records. A naming convention is not required but may be helpful, for example, to append "_history" to the name of the main table.

> **Note**
>
> Note that the *history table* in this context is not the same as SAP HANA history tables in versions before HANA 2 SPS 03.

When you execute DML commands (insert, delete, update) on a system-versioned table, any records which are deleted or updated (and where the *Valid from* and *Valid to* columns are populated) are archived in the history table. Internal triggers are executed to manage the archiving process.

Full details of system-versioned tables are given in the *SAP HANA SQL and System Views Reference* (see, for example CREATE TABLE <system_versioning_spec>).

Some introductory examples are given here. Note that the history table referred to in the definition of the main table must already exist when the main table is created.

Creating System-Versioned Tables

You can either create new system-versioned tables (using `CREATE`) or you can modify an existing table with the `ALTER` statement so that it becomes a system-versioned table. The key syntax elements are shown here using `CREATE`.

```
CREATE COLUMN TABLE my_table
(…
<data columns>
<validfrom_columnname> TIMESTAMP NOT NULL GENERATED ALWAYS AS ROW START,
<validto_columnname> TIMESTAMP NOT NULL GENERATED ALWAYS AS ROW END,
PERIOD FOR SYSTEM_TIME (valid_from, valid_to)
)
WITH SYSTEM VERSIONING HISTORY TABLE my_table_history [[NOT]VALIDATED];
```

- The `valid_from` and `valid_to` columns are timestamps and are maintained by the system; they have a special generation type: *generated always as row start* and *generated always as row end*. The *Valid to* value of the current record is always set to MAX_TIMESTAMP. These valid from and valid to timestamp columns are then identified as *period for system_time* values.
- The `WITH` clause specifies the corresponding history table for the archived data.
- An optional validation keyword is also supported.

History Tables

The history table has essentially the same structure as the main table but must also meet a number of consistency requirements, specifically: the history table does not have the primary key constraint of the current table, and the `valid_from` and `valid_to` columns are defined without generation clause. If you convert an existing table to a history table you must ensure that all requirements for consistency are met.

Only delete operations on history tables are allowed, this is permitted in order to support housekeeping. Inserts and updates are not allowed and generate a 'feature-not-supported' error. As a workaround to this it would be possible to temporarily drop the system versioning on the table and history table and reapply it later.

Archived records in the history table can still be accessed by referring to their timestamp values. To query a history table one or more timestamp values must be provided with one of the following time operators:

- to select data which was valid at a single point in time you can specify a timestamp with the 'timetravel' operator `AS OF`
- to define time ranges you can use the `FROM - TO` syntax or `BETWEEN - AND`

This example gives basic syntax using `AS OF`.

```
SELECT * FROM <system_versioned_table> FOR SYSTEM_TIME AS OF '<utc-timestamp>'
```

A timetravel period can also be defined for the current session using a session variable so that the AS OF is no longer necessary with each command:

```
SET [SESSION] 'TEMPORAL_SYSTEM_TIME_AS_OF' = '2018-01-01';
SELECT * FROM My_sys_time_table;
```

Deleting Data

To permanently delete data from a system-versioned table it is necessary to delete data from both main and history tables in the following sequence:

1. Submit a delete statement referencing the main table - this will delete the data in the main table and move the deleted versions into the history table.
2. Submit a delete statement referencing the history table - this will then delete the data permanently.

System Views

Temporal tables can be identified in the following system views:

- In the `SYS.TABLES` view `TEMPORAL_TYPE` is used to identify temporal tables. Possible values are: *Temporal* for the current table of a system-versioned table or an application-time period table, *History* or *NULL* for any other table type.
- Details of system-versioned tables and their corresponding history table are available in the view `SYS.TEMPORAL_TABLES` (TABLE_NAME and HISTORY_TABLE_NAME). Temporal tables are further categorized in SYS.TEMPORAL_TABLES.PERIOD_NAME where the value is either `SYSTEM_TIME` for the current table of a system-versioned table or `APPLICATION_TIME` for application-time period tables.
- In the `SYS.TABLE_COLUMNS` view `GENERATION_TYPE` indicates either: *ALWAYS AS ROW START* or *ALWAYS AS ROW END*.

Related Information

CREATE TABLE Statement (Data Definition)

8.11.4.3 Application-Time Period Tables

Application-Time Period Tables allow you to manage and manipulate historical business data based on application-specific time periods which are independent of system time-stamps.

Application-time period tables capture the time in which a record is valid in the business world rather than on the basis of the time it was entered in the database. Using these tables, time periods can reflect the past, present or future as determined by business needs and can be updated and managed from within the application. These tables are only available in the column store.

Some applications may require a history based on both system time stamps and business orientated time stamps. For these scenarios application-time period tables can be combined with system-versioned tables to form bi-temporal tables.

Creating and Updating Application-Time Period Tables

Full details of the SQL commands available for working with application-time period tables are given in the *SAP HANA SQL Reference Guide* (see, for example CREATE TABLE <application_time_period_configuration> and <bi_temporal_table_spec>).

Here we give a brief introduction to the essential commands with examples.

You can create an application-time period table with either CREATE or ALTER table commands using the clause PERIOD FOR APPLICATION_TIME (valid_from, valid_to).

To select records which have been valid at a certain point in time ('timetravel'), use the AS OF operator:

```
select * from <app_time_table> for application_time as of ‘<timestamp>’
```

A timetravel period can also be defined for the current session so that the AS OF is no longer necessary with each command:

```
SET [SESSION] 'TEMPORAL_APPLICATION_TIME_AS_OF' = '2018-01-01';
SELECT * FROM My_app_time_table;
```

You can update historical records within a specified time frame using the UPDATE FOR PORTION OF syntax. The following example sets the value of `customer_email` for the period January 2018:

```
UPDATE customers FOR PORTION OF application_time FROM '2018-01-01' TO
'2018-01-31'
SET customer_email='new_email@acmecompany.de' WHERE customer_id=1;
```

This updates records where the ValidFrom and ValidTo dates overlap with the given portion dates. These records will be updated and the validity periods will be adjusted as follows: the valid to date of the existing record is closed, a new record is inserted for the month of January and a new record is inserted from the end of January:

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	VALIDFROM	VALIDTO
1	ACME Company	info@acmecompany.de	2017-10-01	2018-01-01
1	ACME Company	new_email@acmecompany.de	2018-01-01	2018-01-31
1	ACME Company	info@acmecompany.de	2018-01-31	9999-12-31

System Views

The difference between the current table of a system versioned table and an application-time period table can be seen in the database in SYS.TEMPORAL_TABLES.PERIOD_NAME. The value is either `SYSTEM_TIME` for the current table of a system-versioned table or `APPLICATION_TIME` for application-time period tables. Bi-temporal tables have two entries in this view for both SYSTEM_TIME and APPLICATION_TIME.

The column SYS.TABLES.TEMPORAL_TYPE shows the value `TEMPORAL` if the table is an application-time period table or `HISTORY` for history tables of a system-versioned table.

Related Information

CREATE TABLE Statement (Data Definition)

8.11.5 The JSON Document Store

The SAP HANA Document Store is a store for JSON artifacts and allows native operations on JSON including filtering, aggregation, and joining JSON documents with HANA column or row store tables.

JSON documents (JavaScript Object Notation) are stored in so-called collections. The content of a JSON document may be deeply structured but unlike XML it does not have a schema. This means that any valid JSON data may be inserted without first declaring its structure.

Collections appear to users like tables and users can work with them in SQL in a similar fashion. For example, data can be inserted with the regular INSERT statement and read via SELECT. You can read data from tables and collections in a single statement and you can combine tables and collections by joining as with any other column or row store table.

SAP HANA transactions span all three storage types (Column, Row and Document stores) in a single database which conforms to all the principles of data management: atomicity, consistency, isolation and durability.

Technically, collections are stored in a dedicated binary format; in SQL however, they are defined as tables with their own sub-type. You can see this by selecting collections on the basis of the table type value 'COLLECTION':

```
SELECT * FROM tables WHERE table_type = 'COLLECTION'
```

Full technical details of the Document Store are given in the *SAP HANA Developer Guide for XS Advanced*.

Details of Document Store-specific SQL statements are given in the *SAP HANA JSON Document Store Guide* and the *SAP HANA SQL Reference and System Views Guide*. The following example is included here to illustrate how a collection of customer details can be created, updated and read. The INSERT statement illustrates the syntax and structure of nested JSON data and the corresponding SELECT statements show the use of path expressions to navigate the content of the JSON document:

```
create collection Customers;
```

```
insert into Customers values('{"name": "Paul",
                "address": {
                "street": "Main Street 10",
                "city": "Heidelberg"
                }
                }');
```

```
select * from Customers where "address"."city" = 'Heidelberg';
```

```
select "address"."city", count(*) from Customers group by "address"."city";
```

Because there is no schema to enforce consistent use of data types, you may need to implement your own policy to ensure that data is inserted consistently.

Enabling the JSON Document Store

The JSON Document Store is an optional feature of the SAP HANA database which you have to create for each tenant database. Only a single Document Store process per tenant is possible. Create the Document Store using the ALTER DATABASE statement:

```
ALTER DATABASE <database name> ADD 'docstore'
```

Note that once you have created the Document Store and added collections to it, it can only be removed from the landscape by first removing all collections. The JSON Document Store runs only in the *docstore* server process and collections cannot be moved to any other server (such as the indexserver).

The Document Store does not have a pre-determined SQL port, all communication is routed through ordinary index servers.

The Document Store is available for the SAP HANA Express Edition.

Related Information

SAP HANA JSON Document Store Guide
SAP HANA SQL Reference Guide for SAP HANA Platform
Configure Traces in SAP HANA Studio
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
SAP HANA JSON Document Store Guide

8.11.6 Basic Table Management in SAP HANA Cockpit

SAP HANA database explorer is available from the SAP HANA cockpit and can be used to execute SQL statements and database procedures, as well as query information about the database and database objects.

The database explorer is integrated into SAP HANA cockpit and contains features and functions required by both database administrators and developers. Use the SQL console to query information in the database, view information about your database's catalog objects and execute data definition and data manipulation statements to manage the data. Functionality is available to import and export data from a number of sources and to create and manage remote sources and virtual objects. The topic Getting Started With the SAP HANA Database Explorer in the *SAP HANA Database Explorer* guide gives an illustrated overview of all features with links to detailed information.

Related Information

Getting Started With the SAP HANA Database Explorer

8.11.7 Basic Table Management in SAP HANA Studio

The SAP HANA studio provides several functions for the basic administration and monitoring of tables and views.

Details of all table management features in SAP HANA Studio are given in the *SAP HANA Studio* guide. This includes functionality such as:

- Opening tables and views
- Creating tables and views in runtime
- Importing and exporting tables and other catalog objects
- Importing ESRI shape files

Related Information

Links to SAP HANA Studio guide:
Opening Tables and Views
Viewing Options for Tables and Views
Export Tables and Other Catalog Objects
Import Tables and Other Catalog Objects
Import ESRI Shapefiles
Create a Table in Runtime
Create a View in Runtime

8.11.8 Importing Data from External Sources

A number of options are available for importing external data to tables in the SAP HANA database.

You can import data from external sources to the SAP HANA database; this can be either a complete table or data which is appended to an existing table. Typically the data is in CSV format although binary, GZIP and other formats are supported (refer to the *SAP HANA SQL and System Views Reference* for full syntax details). Importing Microsoft Excel formats is possible within the Database Explorer in SAP HANA cockpit which provides a step-by-step wizard for this.

The two SQL commands for importing data are IMPORT and IMPORT FROM:

- IMPORT is used to import catalog objects (tables, views, synonyms, sequences, and procedures) that have previously been exported with the EXPORT statement.
- IMPORT FROM is used to import external data and append it to an existing table.

Using IMPORT FROM with Network Sources

The basic syntax for using IMPORT FROM is shown in this example:

```
IMPORT FROM CSV FILE '/data/data.csv' INTO "MYTABLE"
WITH ERROR LOG '/data/data.err'
```

The file type can be either a CSV file or, for statistics objects, a control file. The file path and the target object are required and a number of options are supported in the WITH <import_from_option_list> clause, these include the location of the error log.

For full details of the options and examples of this command refer to the *SAP HANA SQL and System Views Reference*.

Security

Users need the IMPORT privilege to access files on the file server, this is not granted by default to any user. Access is limited to files which can be read by the <sid>adm of the HANA system.

By default, only three locations can be used for the CSV source data and the error log:

- $DIR_INSTANCE/work
- $DIR_INSTANCE/backup
- $SAP_RETRIEVAL_PATH/trace

If you use a different location in the command it will generate a warning message that the path is invalid. If you wish to use a different location it must be specified in the configuration parameter *csv_import_path_filter* in the `import_export` section of the indexserver.ini configuration file.

For more information on Security see the following SAP Notes:

- 2109565 - Potential information disclosure relating to IMPORT FROM statement in SAP HANA
- 2876090 - Security Restriction on Setting Error Log Option of IMPORT/EXPORT FROM Command.

Named Pipes

Typically, the data must be copied to the specified file path on the HANA server, however, the <file path> clause of this command also supports named pipes. A persistent pipe, also referred to in Linux as 'FIFO', transfers data in memory either from a local or network source and makes it unnecessary to manually copy the CSV file to a local disk. The performance for this type of import is also significantly faster.

In the following example, the HANA server is set up to listen (steps 1-3) and an external node on the network is set up to send data (step 4). This example uses netcat (nc) to set up the named pipe but appropriate security precautions must be taken - see following note.

> ⚠ Caution
>
> You must take responsibility for security implications if using tools such as netcat which provides no encryption and no authentication.

1. Create the named pipe on the HANA server:

```
$ mkfifo /tmp/myfifo
```

2. Start the netcat listener and redirect its output to the named pipe:

```
$ nc -l 8080 > /tmp/myfifo
```

3. Execute the IMPORT FROM command specifying the named pipe as the file path:

```
IMPORT FROM CSV FILE '/tmp/myfifo' INTO "SYSTEM"."TBL_A";
```

4. On the external host start netcat to send the CSV data, specifying the target server and port number and the source of the data:

```
$ nc MyHANAServer:8080 < /path/to/csvfile
```

Once this has been set up it remains open and HANA polls the pipe continuously until either the end of the data stream is reached or until the session is canceled.

The named pipe can also be used for local data and the following example shows a command to redirect local text data to the pipe:

```
$ cat /path/to/csvfile > /tmp/myfifo
```

Related Information

Import Data Into a New or Existing Table (SAP HANA Database Explorer)
HANA SQL Reference Guide: IMPORT FROM Statement (Data Import Export)
SAP Note 2109565
SAP Note 2876090

8.11.9 Importing and Exporting Data With Cloud Sources

You can use SQL commands to import and export data directly from a number of cloud storage services.

Options are available to link to cloud storage systems as the source or target for IMPORT and EXPORT statements. You can work with either SAP catalog objects (optionally using archive files) or you can import or export data in CSV format for a single database object from or to a table in the SAP HANA database.

Related Information

Importing and Exporting with Amazon Web Services [page 267]
Importing and Exporting with Microsoft Azure [page 264]
Importing and Exporting with Alibaba Cloud OSS [page 271]

Importing and Exporting with Google Cloud [page 275]
Importing and Exporting with SAP HANA Cloud Data Lake Storage [page 279]

8.11.9.1 Object Naming Convention

Permitted characters for the object path value in the Import/Export cloud storage path.

The following table gives the list of safe Unicode characters, which can be used in the object path value in the Import/Export cloud storage path (identified in the following sections as <object_id>). Some hyperscalers may support additional characters, but these characters in the range of U+0020 ~ U+007E are guaranteed to be supported for all hyperscalers.

Supported characters in the object_path

Unicode	Character	Character Name
U+0020	' '	Space
U+0021	'!'	Exclamation mark
U+0024	'$'	Dollar sign
U+0026	'&'	Ampersand
U+0027	'''	Apostrophe
U+0028	'('	Left parenthesis
U+0029	')'	Right parenthesis
U+002A	'*'	Asterisk
U+002B	'+'	Plus sign
U+002C	','	Comma
U+002D	'-'	Hyphen-minus
U+002E	'.'	Full stop
U+002F	'/'	Solidus
U+003A	':'	Colon
U+003D	'='	Equal sign
U+003F	'?'	Question mark
U+0040	'@'	At sign
U+005F	'_'	Low line
U+0030~U+0039	'0'~'9'	Digit zero ~digit nine
U+0041~U+005A	'A'~'Z'	Capital letters
U+0061~U+007A	'a'~'z'	Latin small letters

8.11.9.2 Importing and Exporting with Microsoft Azure

You can use SQL commands to import and export data directly from Microsoft Azure storage.

You can work with either SAP catalog objects (optionally using archive files) or you can import or export data in CSV format for a single database object from or to a table in the SAP HANA database.

Prerequisites

Required Token Permissions

For developers who access Azure Cloud Storage through APIs the following tables show the API call names for which shared access signature (SAS) token permissions are required. You must ensure that appropriate read and write permissions are granted for these API calls.

Export: for use with the statements 'Export' and 'Export Into'

API (JSON Format)	Required for...
List Blobs	Get list of objects
DLSGen2. Filesystem Get Properties • DLSGen2. Path Delete • Delete Blob * n	Delete Objects (only required when using EXPORT with the REPLACE option). When the option REPLACE is used then 'Get Properties' is used first to detect if the storage account is using hierarchical namespaces. If the account is using hierarchical namespaces then 'Path Delete' is used, otherwise 'Delete Blob' is used multiple times to remove individual objects.
Put Blob	Write / truncate an object
Put Block	Upload each part for an object
Put Block List	Complete Multipart Upload
Delete Blob	Abort Multipart Upload

Import: for use with the statements 'Import', 'Import From' and 'Import Scan'

API (JSON Format)	Required for...
List Blobs	Get list of objects
Get Blob Properties	Get the metadata of an object
Get Blob	Read data of an object

Preparation

The Microsoft Azure path is made up of the following components:

- Storage account name
- Shared access signature (SAS) token
- Container name
- Object id

```
azure://<storage_account>:<SAS-token>@<container>/<object_id>
```

The signature has two parts: a signed storage resource URI and a token with a set of query parameters, for example:

```
'azure://hanaadlsregression:sv=2019-02-02&ss=bfqt&srt=sco
&sp=rwdlacup&se=2021-02-17T15:15:08Z&st=2020-02-17T07:15:08Z
&spr=https&sig=5WNoL4YEZubOvbXXXXXXX@my-container/IMEX_DEMO'
```

Microsoft Azure Storage Certificate

Additionally, a Microsoft Azure storage SSL certificate is required. Certificates may be specific to a location and you are recommended to contact Microsoft Azure Support to get correct certificate information. The certificate (in PEM format) contains the text to be registered in SAP HANA, copy the text and create a certificate which can be saved in a personal security environment (PSE). The SQL commands to create a PSE, create a certificate, add the certificate to the PSE and set the purpose to REMOTE SOURCE are given here:

```
create pse HTTPS;
create certificate from '<CERTIFICATE_TEXT_FROM_AZURE>' comment 'AZURE';
select CERTIFICATE_ID from CERTIFICATES where COMMENT = 'AZURE'
alter pse HTTPS add certificate <SELECTED_CERTIFICATE_ID>;
set pse HTTPS purpose REMOTE SOURCE;
```

Creating a Credential

To avoid the need to repeatedly enter the private key in plain text a credential for the SAPHANAIMPORTEXPORT component can be created which uses the private key value as a password. Once this has been created it can be more easily used in import and export commands. The syntax for the create credential command is as follows:

```
create credential for [user <userName>] component 'SAPHANAIMPORTEXPORT' purpose
'<purposeName>' type 'PASSWORD'
using 'user=<serviceAccount>;password=<privateKey>';
```

The following example creates a credential named Azure_ImportExport (defined in the purpose):

```
create credential for component 'SAPHANAIMPORTEXPORT' purpose
'Azure_ImportExport' type 'PASSWORD'
using 'user=azure://<path details>@my-container/IMEX_DEMO;password=-----BEGIN
PRIVATE KEY-----...-----END PRIVATE KEY-----';
```

Once the credential has been created the name can be used in the import and export commands as shown in the examples which follow.

SQL Commands

The import and export SQL commands are: IMPORT, IMPORT FROM, EXPORT, EXPORT INTO, IMPORT SCAN. These support a number of options which can be used to specify, for example, delimiter characters, the

number of threads to use during the process, if the data includes a header row, date formats, or an action to take in the event of failure due to invalid data. For full details of these commands and all the options supported refer to the corresponding topics in the *SAP HANA SQL Reference Guide for SAP HANA Platform* (see Related Links).

For IMPORT you must have the IMPORT system privilege, and the CREATE ANY privilege on the schema that is being imported into.

For EXPORT you must have the EXPORT system privilege and have the SELECT privilege on the objects being exported.

IMPORT / EXPORT Catalog Objects

Use the IMPORT and EXPORT commands to work with SAP catalog objects. Objects are exported in either binary or CSV formats, optionally you can export and import data from / to archive files. Import is used to re-import objects which have been exported using the EXPORT command, the file format binary or CSV is detected automatically, if not specified. Note, however, that it is not possible to import binary or CSV objects which have been exported from the on-premise SAP HANA Platform because the two formats and certain data types are not compatible.

The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT <object_name_list> FROM '<Azure_path>' [with options];
```

```
EXPORT <object_name_list> AS CSV INTO '<Azure_path>' [with options];
```

These commands require temporary disk space: IMPORT requires enough disk space to download objects from Microsoft Azure and/or extract archive files and EXPORT requires disk space as it exports data into the work directory before uploading and/or archiving.

Examples

This example exports all objects of the schema IMEX_DEMO:

```
export IMEX_DEMO."*" into 'azure://hanaadlsregression:sv=2019-02-02
&ss=bfqt&srt=sco&sp=rwdlacup&se=2021-02-17T15:15:08Z&st=2020-02-17T07:15:08Z&spr=
https
&sig=5WNoL4YEZubOvbXXXXXXX@my-container/IMEX_DEMO' with threads 4 replace;
```

This example imports objects from the exported location:

```
import all from 'azure://hanaadlsregression:sv=2019-02-02
&ss=bfqt&srt=sco&sp=rwdlacup&se=2021-02-17T15:15:08Z&st=2020-02-17T07:15:08Z&spr=
https
&sig=5WNoL4YEZubOvbXXXXXXX@my-container/IMEX_DEMO' with threads 4 replace;
```

IMPORT / EXPORT CSV Files

You can use the IMPORT FROM and EXPORT INTO commands to work with single database objects in CSV format. Note that exporting objects in this format does not include table information like index or column types. The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT FROM CSV FILE '<Azure_path.csv>' INTO <TARGET_TABLE_NAME> [with options];
```

```
EXPORT INTO '<Azure_path.csv>' FROM <SOURCE_TABLE_OR_VIEW_NAME> [with options];
```

IMPORT/EXPORT of CSV files does not require additional temporary space as these commands buffer I/O directly from/into Microsoft Azure storage.

Examples

This example exports a table into a CSV file:

```
export into 'azure://hanaadlsregression:sv=2019-02-02&ss=bfqt&srt=sco
&sp=rwdlacup&se=2021-02-17T15:15:08Z&st=2020-02-17T07:15:08Z&spr=https
&sig=5WNoL4YEZubOvbXXXXXXX@my-container/DEMO_TBL1.csv'
from "IMEX_DEMO"."DEMO_TBL1" with field delimited by ',' threads 4;
```

This example imports data from the exported CSV file:

```
import from csv file 'azure://hanaadlsregression:sv=2019-02-02&ss=bfqt&srt=sco
&sp=rwdlacup&se=2021-02-17T15:15:08Z&st=2020-02-17T07:15:08Z&spr=https
&sig=5WNoL4YEZubOvbXXXXXXX@my-container/DEMO_TBL1.csv'
into "IMEX_DEMO"."DEMO_TBL1" with field delimited by ',' threads 4;
```

IMPORT SCAN

The IMPORT SCAN statement searches a given path for objects exported with the EXPORT statement. It stores the results in the session-local temporary table #IMPORT_SCAN_RESULT in the current schema. The following example illustrates the usage:

```
import scan 'azure://hanaadlsregression:sv=2019-02-02&ss=bfqt&srt=sco
&sp=rwdlacup&se=2021-02-17T15:15:08Z&st=2020-02-17T07:15:08Z&spr=https
&sig=5WNoL4YEZubOvbXXXXXXX@my-container/IMEX_DEMO'
```

Select the data in the #IMPORT_SCAN_RESULT table to access the result of the scan:

```
select * from #IMPORT_SCAN_RESULT;
```

Related Information

IMPORT Statement (Data Import Export)
IMPORT FROM Statement (Data Import Export)
IMPORT SCAN Statement (Data Import Export)
EXPORT Statement (Data Import Export)
EXPORT INTO Statement (Data Import Export)
Import Certificates for SSL Connections to Remote Sources [page 1551]

8.11.9.3 Importing and Exporting with Amazon Web Services

You can use SQL commands to import and export data directly from Amazon Web Services (AWS) by specifying the AWS S3 path and the required credentials.

You can work with either SAP catalog objects (optionally using archive files) or you can import or export data in CSV format for a single database object from or to a table in the SAP HANA database.

Prerequisites

Required Token Permissions

For developers who access AWS S3 Cloud Storage through APIs the following tables show the API call names for which permissions (access keys - see below) are required. You must ensure that appropriate read and write permissions are granted for these API calls.

Export: for use with the statements 'Export' and 'Export Into'

API (JSON Format)	Required for...
ListObjectsV2	Get list of objects
DeleteObjects	Delete Objects (only required when using EXPORT with the REPLACE option).
PutObject	Write / truncate an object
CreateMultiplartUpload	Start Multipart Upload
UploadPart	Upload each part for an object
CompleteMultipartUpload	Complete Multipart Upload
AbortMultipartUpload	Abort Multipart Upload

Import: for use with the statements 'Import', 'Import From' and 'Import Scan'

API (JSON Format)	Required for...
ListObjectsV2	Get list of objects
HeadObject	Get the metadata of an object
GetObject	Read data of an object

Preparation

The AWS S3 path must include the region, the access credentials and the bucket name:

```
's3-<region>://<access_key>:<secret_key>@<bucket>/<object_id>'
```

Credentials required to access AWS S3 through an API are made up of a pair of values: an access key ID and a corresponding secret key, these are issued in the Amazon Identity and Access Management (IAM) console under *Access keys for API Access*. Currently access with a key pair for an enforced multi-factor authentication account is not supported.

The object ID specifies a path or file name within the named bucket. You can look up geographical regions and endpoint details in the *Amazon Relational Database Services User Guide* under *Regions and Availability Zones*.

A path for the European Central region, for example, may look like this:

```
's3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-demo/MySchema'
```

Note that the use of custom or customer-specific domains (such as `<bucket_name>.sap.com`) is not supported and are not accessible by AWS S3 endpoints.

> → Tip
>
> Bear in mind the need to flush the bucket of unwanted data to avoid unnecessary data charges. The SAP HANA IMPORT/EXPORT statement does not support a DELETE option on AWS S3 but you can limit the lifespan of a bucket if necessary by setting an expiration date for the bucket in the Amazon Management Console.

Amazon Web Services Certificate

Additionally, an AWS S3 SSL certificate is required. Certificates may be specific to a region and you are recommended to contact AWS Support to get correct certificate information. The certificate (in PEM format) contains the text to be registered in SAP HANA, copy the text and create a certificate which can be saved in a personal security environment (PSE). The SQL commands to create a PSE, create a certificate, add the certificate to the PSE and set the purpose to REMOTE SOURCE are given here:

```
create pse HTTPS;
create certificate from '<CERTIFICATE_TEXT_FROM_S3>' comment 'S3';
select CERTIFICATE_ID from CERTIFICATES where COMMENT = 'S3'
alter pse HTTPS add certificate <SELECTED_CERTIFICATE_ID>;
set pse HTTPS purpose REMOTE SOURCE;
```

Creating a Credential

To avoid the need to repeatedly enter the private key in plain text a credential for the SAPHANAIMPORTEXPORT component can be created which uses the private key value as a password. Once this has been created it can be more easily used in import and export commands. The syntax for the create credential command is as follows:

```
create credential for [user <userName>] component 'SAPHANAIMPORTEXPORT' purpose
'<purposeName>' type 'PASSWORD'
using 'user=<serviceAccount>;password=<privateKey>';
```

The following example creates a credential named S3_ImportExport (defined in the purpose):

```
create credential for component 'SAPHANAIMPORTEXPORT' purpose 'S3_ImportExport'
type 'PASSWORD'
using 'user=s3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-demo/
MySchema;password=-----BEGIN PRIVATE KEY-----...-----END PRIVATE KEY-----';
```

Once the credential has been created the name can be used in the import and export commands as shown in the examples which follow.

SQL Commands

The import and export SQL commands are: IMPORT, IMPORT FROM, EXPORT, EXPORT INTO, IMPORT SCAN. These support a number of options which can be used to specify, for example, delimiter characters, the

number of threads to use during the process, if the data includes a header row, date formats, or an action to take in the event of failure due to invalid data. For full details of these commands and all the options supported refer to the corresponding topics in the *SAP HANA SQL Reference Guide for SAP HANA Platform* (see Related Information links).

For IMPORT you must have the IMPORT system privilege, and the CREATE ANY privilege on the schema that is being imported into.

For EXPORT you must have the EXPORT system privilege and have the SELECT privilege on the objects being exported.

IMPORT / EXPORT Catalog Objects

Use the IMPORT and EXPORT commands to work with SAP catalog objects. Objects are exported in either binary or CSV formats, optionally you can export and import data from / to archive files. Import is used to re-import objects which have been exported using the EXPORT command, the file format binary or CSV is detected automatically, if not specified. Note, however, that it is not possible to import binary or CSV objects which have been exported from the on-premise SAP HANA Platform because the two formats and certain data types are not compatible.

The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT <object_name_list> FROM '<S3_path>' [with options];
```

```
EXPORT <object_name_list> AS CSV INTO '<S3_path>' [with options];
```

These commands require temporary disk space: IMPORT requires enough disk space to download objects from S3 and/or extract archive files and EXPORT requires disk space as it exports data into the work directory before uploading and/or archiving.

Examples

This example exports all objects of the schema MySchema:

```
export MySchema."*" into 's3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-
demo/MySchema' with threads 4 replace;
```

This example imports objects from the exported location:

```
import all from 's3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-demo/
MySchema' with threads 4 replace;
```

IMPORT / EXPORT CSV Files

You can use the IMPORT FROM and EXPORT INTO commands to work with single database objects in CSV format. Note that exporting objects in this format does not include table information like index or column types. The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT FROM CSV FILE '<S3_path.csv>' INTO <TARGET_TABLE_NAME> [with options];
```

```
EXPORT INTO '<S3_path.csv>' FROM <SOURCE_TABLE_NAME> [with options];
```

IMPORT/EXPORT of CSV files does not require additional temporary space as these commands buffer I/O directly from/into AWS S3.

Examples

This example exports a table to a CSV file:

```
export into 's3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-demo/
My_lineitem.csv'
from MySchema.LINEITEM with field delimited by ',' threads 4;
```

This example imports a table in CSV format:

```
import from csv file 's3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-demo/
My_lineitem.csv' into MySchema.LINEITEM
with field delimited by ',' threads 4;
```

IMPORT SCAN

The IMPORT SCAN statement searches a given path for objects exported with the EXPORT statement. It stores the results in the session-local temporary table #IMPORT_SCAN_RESULT in the current schema. The following example illustrates the usage:

```
import scan 's3-eu-central-1://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@my-demo'
```

Select the data in the #IMPORT_SCAN_RESULT table to access the result of the scan:

```
select * from #IMPORT_SCAN_RESULT;
```

Related Information

IMPORT Statement (Data Import Export)
IMPORT FROM Statement (Data Import Export)
EXPORT Statement (Data Import Export)
EXPORT INTO Statement (Data Import Export)
IMPORT SCAN Statement (Data Import Export)
Import Certificates for SSL Connections to Remote Sources [page 1551]

8.11.9.4 Importing and Exporting with Alibaba Cloud OSS

You can use SQL commands to import and export data directly from Alibaba Cloud OSS (Object Storage Service) by specifying the OSS path and the required credentials.

You can work with either SAP catalog objects and archive files (import from archive) or you can import or export data in CSV format for a single database object from or to a table in the SAP HANA database.

Prerequisites

Required Token Permissions

For developers who access Alibaba Cloud Storage through APIs the following tables show the API call names for which permissions are required. You must ensure that appropriate read and write permissions are granted for these API calls.

Export: for use with the statements 'Export' and 'Export Into'

API (JSON Format)	Required for...
GetBucket	Get list of objects
DeleteObject	Delete Objects (only required when using EXPORT with the REPLACE option).
PutObject	Write / truncate an object
InitiateMultiplartUpload	Start Multipart Upload
UploadPart	Upload each part for an object
CompleteMultipartUpload	Complete Multipart Upload
AbortMultipartUpload	Abort Multipart Upload

Import: for use with the statements 'Import', 'Import From' and 'Import Scan'

API (JSON Format)	Required for...
GetBucket	Get list of objects
HeadObject	Get the metadata of an object
GetObject	Read data of an object

Preparation

The Alibaba Cloud path must include the region, the access credentials and the bucket name:

```
oss://<access_key_id>:<access_key_secret>@<region>.aliyuncs.com/<bucket>/
<object_id>
```

Credentials required to access Alibaba Cloud through an API are made up of a pair of values: an access key ID and a corresponding secret key. Currently access with a key pair for an enforced multi-factor authentication account is not supported.

The object ID specifies a path or file name within the named bucket. You can look up geographical regions and endpoint details in the Alibaba Cloud documentation *Object Storage Service* under *Regions and endpoints*.

A path for the Shanghai region, for example, may look like this:

```
oss://AKIAXXXXXXXXX:x16WWxxxxxxxxxxx@oss-cn-shanghai.aliyuncs.com/mybucket/
regression_test/BUG000000/export_path
```

Note that the use of custom or customer-specific domains (such as `<bucket_name>.sap.com`) is not supported and are not accessible by Alibaba Cloud endpoints.

Alibaba Cloud OSS Certificate

Additionally, an Alibaba Cloud OSS certificate is required. The certificate (in PEM format) contains the text to be registered in SAP HANA, copy the text and create a certificate which can be saved in a personal security environment (PSE). The SQL commands to create a PSE, create a certificate, add the certificate to the PSE and set the purpose to REMOTE SOURCE are given here:

```
create pse HTTPS;
create certificate from '<CERTIFICATE_TEXT_FROM_AliCloud>' comment 'AliCloud';
select CERTIFICATE_ID from CERTIFICATES where COMMENT = 'AliCloud'
alter pse HTTPS add certificate <SELECTED_CERTIFICATE_ID>;
set pse HTTPS purpose REMOTE SOURCE;
```

Creating a Credential

To avoid the need to repeatedly enter the private key in plain text a credential for the SAPHANAIMPORTEXPORT component can be created which uses the private key value as a password. Once this has been created it can be more easily used in import and export commands. The syntax for the create credential command is as follows:

```
create credential for [user <userName>] component 'SAPHANAIMPORTEXPORT' purpose
'<purposeName>' type 'PASSWORD'
using 'user=<serviceAccount>;password=<privateKey>';
```

The following example creates a credential named OSS_ImportExport (defined in the purpose):

```
create credential for component 'SAPHANAIMPORTEXPORT' purpose 'OSS_ImportExport'
type 'PASSWORD'
using 'user=oss://AKIAXXXXXXXXX:x16WWxxxxxxxxxxx@oss-cn-shanghai.aliyuncs.com/
mybucket/regression_test/BUG000000/export_path;password=-----BEGIN PRIVATE
KEY------..-----END PRIVATE KEY-----';
```

Once the credential has been created the name can be used in the import and export commands as shown in the examples which follow.

SQL Commands

SAP HANA SQL Reference Guide for SAP HANA Platform (see Related Information links).

For IMPORT you must have the IMPORT system privilege, and the CREATE ANY privilege on the schema that is being imported into. The import and export SQL commands are: IMPORT, IMPORT FROM, EXPORT, EXPORT INTO, IMPORT SCAN. These support a number of options which can be used to specify, for example, delimiter characters, the number of threads to use during the process, if the data includes a header row, date formats, or an action to take in the event of failure due to invalid data. For full details of these commands and all the options supported refer to the corresponding topics in the

For EXPORT you must have the EXPORT system privilege and have the SELECT privilege on the objects being exported.

IMPORT / EXPORT Catalog Objects

Use the IMPORT and EXPORT commands to work with SAP catalog objects. Objects are exported in either binary or CSV formats, you can also import data from archive files. Import is used to re-import objects which have been exported using the EXPORT command; the file format, binary or CSV, is detected automatically if not specified. Note, however, that it is not possible to import binary or CSV objects which have been exported from the on-premise SAP HANA Platform because the two formats and certain data types are not compatible.

The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT <object_name_list> FROM '<storage_path>' [with options];
```

```
EXPORT <object_name_list> AS CSV INTO '<storage_path>' [with options];
```

Examples

This example exports all objects of the schema MySchema:

```
export MySchema."*" into 'oss://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@oss-eu-
central-1.aliyuncs.com/imex-demo/MySchema' with threads 4 replace;
```

This example imports objects from the exported location:

```
import all from 'oss://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@oss-eu-
central-1.aliyuncs.com/imex-demo/MySchema' with threads 4 replace;
```

IMPORT / EXPORT CSV Files

You can use the IMPORT FROM and EXPORT INTO commands to work with single database objects in CSV format. Note that exporting objects in this format does not include table information like index or column types. The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT FROM CSV FILE '<AliCloud_path.csv>' INTO <TARGET_TABLE_NAME> [with
options];
```

```
EXPORT INTO '<AliCloud_path.csv>' FROM <SOURCE_TABLE_NAME> [with options];
```

IMPORT/EXPORT of CSV files does not require additional temporary space as these commands buffer I/O directly from/into Alibaba Cloud.

Examples

This example exports a table to a CSV file:

```
The import and export SQL commands are:export
into 'oss://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@oss-eu-central-1.aliyuncs.com/imex-
demo/My_lineitem.csv'
from MySchema.LINEITEM with field delimited by ',' threads 4;
```

This example imports a table in CSV format:

```
import from csv file 'oss://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@oss-eu-
central-1.aliyuncs.com/imex-demo/My_lineitem.csv'
into MySchema.LINEITEM with field delimited by ',' threads 4;
```

IMPORT SCAN

The IMPORT SCAN statement searches a given path for objects exported with the EXPORT statement. It stores the results in the session-local temporary table #IMPORT_SCAN_RESULT in the current schema. The following example illustrates the usage:

```
import scan 'oss://AKIAxxxxxxxxxx:xl6WWxxxxxxxxxx@oss-eu-central-1.aliyuncs.com/
imex-demo/'
```

Select the data in the #IMPORT_SCAN_RESULT table to access the result of the scan:

```
select * from #IMPORT_SCAN_RESULT;
```

Related Information

IMPORT Statement (Data Import Export)
IMPORT FROM Statement (Data Import Export)
EXPORT Statement (Data Import Export)
EXPORT INTO Statement (Data Import Export)
IMPORT SCAN Statement (Data Import Export)
Import Certificates for SSL Connections to Remote Sources [page 1551]

8.11.9.5 Importing and Exporting with Google Cloud

You can use SQL commands to import and export data directly from Google Cloud.

You can work with either SAP catalog objects (optionally using archive files) or you can import or export data in CSV format for a single database object from or to a table in the SAP HANA database. The SQL statements require the path to the GCS storage and also a reference to a credential identifying a password which must be created in advance. This feature can be used in all Import/Export statements: EXPORT, IMPORT, EXPORT INTO, IMPORT FROM, and IMPORT SCAN.

Prerequisites

Required Token Permissions

For developers who access Google Cloud Storage through APIs the following tables show the API call names for which permissions are required. You must ensure that appropriate read and write permissions are granted for these API calls.

Export: for use with the statements ‘Export’ and ‘Export Into’

API (XML Format)	Required for...
GET Bucket	Get list of objects
DELETE Object * n	Delete Objects (only required when using EXPORT with the REPLACE option).
PUT Object	Write / truncate an object
POST Object	Start Multipart Upload
PUT Object	Upload each part for an object
POST Object	Complete Multipart Upload
DELETE Object	Abort Multipart Upload

Import: for use with the statements ‘Import’, ‘Import From’ and ‘Import Scan’

API (XML Format)	Required for...
GET Bucket	Get list of objects
HEAD Object	Get the metadata of an object
GET Object	Read data of an object

Preparation

The Google Cloud Storage (GCS) path is specified as follows:

```
gs://<service_account>:<private_key>@<bucket>/<object_id>
```

The credential pair, made up of service account ID and private key, is required for API access and is issued in the Google Identity and Access Management (IAM) Console. Refer to Google documentation on Service Accounts for more information. The bucket name is the GCS source or target for the import/export command. The full path in plain text is therefore very long, the private key value is truncated in the following example:

```
'gs://gc-storage@sap-bigquery-access.iam.gserviceaccount.com:-----BEGIN PRIVATE
KEY-----  -----END PRIVATE KEY-----
@imex-regression-test/afae087c-ef55-4c63-bcbd-edfb4d251581'
```

Google Cloud Storage SSL Certificate

A GCS SSL certificate is required. Certificates may be specific to a location. Information about endpoints to download certificates is available from Google, for example: Request endpoints . The certificate (in PEM format) contains the text to be registered in SAP HANA, copy the text and create a certificate which can be saved in a personal security environment (PSE). The SQL commands to create a PSE, create a certificate, add the certificate to the PSE and set the purpose to REMOTE SOURCE are given here:

```
create pse HTTPS;
```

```
create certificate from '<CERTIFICATE_TEXT_FROM_GOOGLE>' comment 'GOOGLE_CERT';
select CERTIFICATE_ID from CERTIFICATES where COMMENT = 'GOOGLE_CERT'
alter pse HTTPS add certificate <SELECTED_CERTIFICATE_ID>;
set pse HTTPS purpose REMOTE SOURCE;
```

Creating a Credential

To avoid the need to repeatedly enter the private key in plain text a credential for the SAPHANAIMPORTEXPORT component can be created which uses the private key value as a password. Once this has been created it can be more easily used in import and export commands. The syntax for the create credential command is as follows:

```
create credential for [user <userName>] component 'SAPHANAIMPORTEXPORT' purpose
'<purposeName>' type 'PASSWORD'
using 'user=<serviceAccount>;password=<privateKey>';
```

The following example creates a credential named GCS_ImportExport (defined in the purpose):

```
create credential for component 'SAPHANAIMPORTEXPORT' purpose
'GCS_ImportExport' type 'PASSWORD' using 'user=gc-storage@sap-bigquery-
access.iam.gserviceaccount.com;password=-----BEGIN PRIVATE KEY-----...-----END
PRIVATE KEY-----';
```

Once the credential has been created the name can be used in the import and export commands as shown in the examples which follow.

SQL Commands

The import and export SQL commands are: IMPORT, IMPORT FROM, EXPORT, EXPORT INTO, IMPORT SCAN. These support a number of options which can be used to specify, for example, delimiter characters, the number of threads to use during the process, if the data includes a header row, date formats, or an action to take in the event of failure due to invalid data. For full details of these commands and all the options supported refer to the corresponding topics in the *SAP HANA SQL Reference Guide for SAP HANA Platform* (see Related Links).

For IMPORT you must have the IMPORT system privilege, and the CREATE ANY privilege on the schema that is being imported into.

For EXPORT you must have the EXPORT system privilege and have the SELECT privilege on the objects being exported.

IMPORT / EXPORT Catalog Objects

Use the IMPORT and EXPORT commands to work with SAP catalog objects. Objects are exported in either binary or CSV formats, optionally you can export and import data from / to archive files. Import is used to re-import objects which have been exported using the EXPORT command, the file format binary or CSV is detected automatically, if not specified. Note, however, that it is not possible to import binary or CSV objects which have been exported from the on-premise SAP HANA Platform because the two formats and certain datatypes are not compatible.

The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
IMPORT <object_name_list> FROM '<GCS_path>' [with options];
```

```
EXPORT <object_name_list> AS CSV INTO '<GCS_path>' [with options];
```

These commands require temporary disk space: IMPORT requires enough disk space to download objects and/or extract archive files and EXPORT requires disk space as it exports data into the work directory before uploading and/or archiving.

IMPORT / EXPORT CSV Files

You can use the IMPORT FROM and EXPORT INTO commands to work with single database objects in CSV format. Note that exporting objects in this format does not include table information like index or column types. The basic syntax of the commands is shown here, the examples below illustrate the usage:

```
import <tables> into 'gs://<bucket>/<object_id>' with credential '<purposeName>';
```

```
export <tables> into 'gs://<bucket>/<object_id>' with credential
'<purposeName>';
```

IMPORT/EXPORT of CSV files does not require additional temporary space as these commands buffer I/O directly from/into GCS storage.

Examples

This example exports a table into a CSV file using the credential already created:

```
export "IMEX_RESTTEST"."*" as CSV into 'gs://imex-regression-test/
71ef7736-3927-44e8-bd0d-8ec629cefc0b' with credential 'GCS_ImportExport';
```

This example imports data from the exported CSV file using the credential already created:

```
import all from 'gs://imex-regression-test/71ef7736-3927-44e8-bd0d-8ec629cefc0b'
with credential 'GCS_ImportExport';
```

IMPORT SCAN

The IMPORT SCAN statement searches a given path for objects exported with the EXPORT statement. It stores the results in the session-local temporary table #IMPORT_SCAN_RESULT in the current schema. The following example illustrates the usage:

```
import scan 'gs://imex-regression-test/71ef7736-3927-44e8-bd0d-8ec629cefc0b'
with credential 'GCS_ImportExport'
```

Select the data in the #IMPORT_SCAN_RESULT table to access the result of the scan:

```
select * from #IMPORT_SCAN_RESULT;
```

Related Information

IMPORT Statement (Data Import Export)
IMPORT FROM Statement (Data Import Export)
IMPORT SCAN Statement (Data Import Export)
EXPORT Statement (Data Import Export)
EXPORT INTO Statement (Data Import Export)

8.11.9.6 Importing and Exporting with SAP HANA Cloud Data Lake Storage

You can use SQL commands to import and export data directly from SAP HANA Cloud data lake file storage by specifying the path to the file store and creating the required credentials for users who need to access data lake.

Prerequisites

Required Token Permissions

For developers who access Data Lake Files Storage through APIs the following tables show the API call names for which permissions are required. You must ensure that appropriate read and write permissions are granted for these API calls.

Export: for use with the statements 'Export' and 'Export Into'

API (JSON Format)	Required for...
LISTSTATUS_RECURSIVE	Get list of objects
DELETE_BATCH	Delete Objects (only required when using EXPORT with the REPLACE option).
GETOPERATIONSTATUS	Write / truncate an object
CREATE	Upload each part for an object
MERGE	Complete Multipart Upload
DELETE_BATCH	Abort Multipart Upload

Import: for use with the statements 'Import', 'Import From' and 'Import Scan'

API (JSON Format)	Required for...
LISTSTATUS_RECURSIVE	Get list of objects
GETFILESTATUS	Get the metadata of an object
OPEN	Read data of an object

Preparation

A valid credential name and the path to the file store are required in the import and export statements. To access SAP HANA Cloud data lake file storage the use of the WITH CREDENTIAL clause is mandatory for execution of Import and Export statements. The credential uses the CA certificate key and is uploaded to a PSE certificate store. This avoids the need to repeatedly enter the certificate as plain text for each Import/Export statement.

The following Import/Export statements are supported: EXPORT, IMPORT, EXPORT INTO, IMPORT FROM, and IMPORT SCAN.

You can work with either SAP catalog objects or you can import or export data in CSV format for a single database object from or to a table in the SAP HANA database.

There are two use cases which are illustrated in the examples given below:

1. HANA uses an external file container (not necessarily its SAP HANA Cloud data lake scope and the endpoint value requires the full server address including the file container ID).
2. Optional clause AUTHENTICATION FACTOR is available for 'x-sap-trusted-user'.

File Path

The file path which is required to access an external file container is specified as follows:

```
hdlfs://<endpoint address>/<object_id>
```

The SAP HANA Cloud data lake file storage endpoint value is the server address which consists of file container id and landscape (see examples below).

Certificates

The OpenSSL tool can be used to generate certificates.

The client certificate needs to be registered to the file container. For instructions on uploading a certificate refer to the 'Manage Trusts in the Data Lake File Container' section of the document *SAP HANA Cloud Administration with SAP HANA Cloud Central*.

The steps required to set up the certificate in a PSE (personal security environment) are as follows:

1. Create the certificate collection; TRUST ADMIN privilege is required for this:

```
Create PSE myPSE
```

2. Create the certificate (system privilege CERTIFICATE ADMIN is required) and set a comment value which can be used to identify the certificate, for example 'SSLRootCA':

```
CREATE CERTIFICATE FROM '-----BEGIN CERTIFICATE-----……-----END
CERTIFICATE-----' COMMENT 'SSLRootCA'
```

3. Use the comment value to retrieve the certificate ID:

```
SELECT CERTIFICATE_ID FROM CERTIFICATES WHERE COMMENT = 'SSLRootCA'
```

4. Add the certificate ID (for example 123456) to the certificate collection:

```
ALTER PSE myPSE ADD CERTIFICATE 123456
```

5. Set the own certificate with the following values in plain text:

- Client private key
- Client certificate
- Root Certification Authority of the client certificate

```
ALTER PSE myPSE SET OWN CERTIFICATE '-----BEGIN PRIVATE KEY-----….
.-----END PRIVATE KEY----------BEGIN CERTIFICATE-----…
…-----END CERTIFICATE----------BEGIN CERTIFICATE----------END CERTIFICATE-----'
```

6. Give access to the user with the object privilege REFERENCES:

```
GRANT REFERENCES ON PSE myPSE TO myDBUser
```

7. Use the CREATE CREDENTIAL FOR USER statement to enable the database user. The component value is SAPHANAIMPORTEXPORT. The credential name is given as the value for 'purpose' and the type must be X509 for data lake access. You can name a specific user as shown here, but this is optional and if no user is named then the credential is public.

```
CREATE CREDENTIAL FOR USER myDBUser COMPONENT 'SAPHANAIMPORTEXPORT' PURPOSE
'myCredential' TYPE 'X509' PSE myPSE
```

Once the setup is complete the authorized user can use this credential name in the WITH CREDENTIAL clause of all execute import and export statements.

Example SQL Statements

The basic specification is shown here. Examples here also show the use of the optional AUTHENTICATION FACTOR for 'x-sap-trusted-user'.

```
IMPORT/EXPORT …<hdlfs path>… WITH CREDENTIAL <credentialName>
[AUTHENTICATION FACTOR 'x-sap-trusted-user=<trustedUser>;x-sap-trusted-user-
encoding=<encodingType>;x-sap-trusted-user-roles=<roles>']
```

Basic statements to import CSV data into SAP HANA and export data in CSV format from SAP HANA are as follows using the file store path and the user credentials:

```
IMPORT FROM CSV FILE 'hdlfs://example-file-container.files.hdl.demo-hc-3-hdl-
hc-dev.dev-aws.hanacloud.ondemand.com/directory/data.csv' INTO myTable WITH
CREDENTIAL 'myCredential'
```

```
EXPORT myTable AS CSV into 'hdlfs://example-file-container.files.hdl.demo-hc-3-
hdl-hc-dev.dev-aws.hanacloud.ondemand.com/directory/data.csv' WITH CREDENTIAL
'myCredential'
```

The same examples using authentication factor:

```
IMPORT FROM CSV FILE 'hdlfs://example-file-container.files.hdl.demo-
hc-3-hdl-hc-dev.dev-aws.hanacloud.ondemand.com/directory/data.csv' INTO
myTable WITH CREDENTIAL 'myCredential' AUTHENTICATION
FACTOR 'x-sap-trusted-user=myTrustedUser;x-sap-trusted-user-encoding=b64_utf;x-
sap-trusted-user-roles=myCustomRole'
```

```
EXPORT myTable AS CSV into 'hdlfs://example-file-container.files.hdl.demo-hc-3-
hdl-hc-dev.dev-aws.hanacloud.ondemand.com/directory/data.csv' WITH CREDENTIAL
'myCredential' AUTHENTICATION FACTOR 'x-sap-trusted-user=myTrustedUser;x-sap-
trusted-user-encoding=b64_utf;x-sap-trusted-user-roles=myCustomRole'
```

Related Information

Manage Trusts in the Data Lake File Container

IMPORT Statement (Data Import Export)
IMPORT FROM Statement (Data Import Export)
IMPORT SCAN Statement (Data Import Export)
EXPORT Statement (Data Import Export)
EXPORT INTO Statement (Data Import Export)

8.11.10 Table and Catalog Consistency Checks

Using stored procedures and commands available in the SAP HANA database, you can perform a range of consistency checks on the database catalog and on database tables.

You are recommended to integrate consistency checks into your routine maintenance schedule so that any problems can be detected as soon as they occur.

Two command line procedures are available to check table consistency and the database catalog:

```
CALL CHECK_TABLE_CONSISTENCY()
```

```
CALL CHECK_CATALOG()
```

Optionally, the table consistency check can be scheduled within the embedded statistics service.

For each procedure a list of checking actions is available, for example, CHECK_COLUMN_TABLES, CHECK_ROW_TABLES, CHECK_PARTITIONING_DATA, and so on; these can all be individually activated or omitted from the check as required. For some of these checks a repair option is supported, for example REPAIR_PARTITIONING_DATA. Additional privileges are required for repair actions, these actions must be explicitly specified and must be run separately from check actions. A complete list of all check and repair actions for the two procedures is available by running `GET_CHECK_ACTIONS()`. Details of these commands, configuration options and the statistics features for table consistency checks are given in the sections which follow.

> → Recommendation
>
> Running database checks affects system performance therefore the checks should be run in a timeframe when the system is not at high load. If you are running an active/active (read enabled) system you can run the checks on the read enabled secondary system, or possibly on a system copy.

Related Information

Table Consistency Check [page 283]
Catalog Consistency Check [page 286]
Configuration Parameters for the Table Consistency Check [page 287]
Active/Active (Read Enabled) [page 852]
The Statistics Service [page 143]

8.11.10.1 Table Consistency Check

The table consistency check is a procedure available in the SAP HANA database that performs a range of consistency check actions on database tables. It can be run from the command line or scheduled within the statistics service.

Manual Execution

To execute the procedure manually, you must have the following system privileges:

- CATALOG READ for check actions
- DATA ADMIN for repair actions

Input Parameters

To see details of all check actions which relate to table consistency, including a description of what they do, call the procedure **GET_CHECK_ACTIONS**:

```
CALL GET_CHECK_ACTIONS('CHECK_TABLE_CONSISTENCY')
```

Syntax

The syntax of the table consistency check procedure is as follows:

```
CALL CHECK_TABLE_CONSISTENCY ('<check_action1>[,<check_action2>]',
'<schema_name>', '<table_name>')
```

This procedure is also available for the Dynamic Tiering option but the syntax and options supported are different. Refer to the SAP HANA SQL Reference Guide for details.

Use the parameter `check_action` to define one or more specific check actions, or enter **CHECK** as the value to execute all default check actions. Use the parameters `schema_name` and `table_name` to define specific schemas and tables to check, or enter **NULL** as the value for these parameters to check all tables in all schemas.

> ### Example
>
> The first example performs all default check actions on all tables. The second example performs only two specific checks (primary key and column tables) on two named tables:
>
> ```
> CALL CHECK_TABLE_CONSISTENCY ('CHECK', NULL, NULL)
> ```
>
> ```
> CALL CHECK_TABLE_CONSISTENCY('CHECK, CHECK_PRIMARY_KEY, CHECK_COLUMN_TABLES',
> 'MYSCHEMA', 'MYTABLE01, MYTABLE02');
> ```

Check actions can be excluded by using a minus character before the action name. This command for example, executes all checks which are included in the summary action 'CHECK' except for CHECK_PRIMARY_KEY:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK, -CHECK_PRIMARY_KEY', NULL, NULL);
```

The following three actions cannot be excluded: 'CHECK', 'CHECK_COLUMN_TABLES', 'CHECK_ROW_TABLES'.

It is also possible to use the minus character to exclude specific tables from the consistency check. This command for example, executes all checks on all tables except for tables 'T1' and 'T2':

```
CALL CHECK_TABLE_CONSISTENCY('CHECK', 'SYSTEM', '-T1, -T2');
```

Note that you cannot mix excluding tables and naming specific tables in the same command; this would generate an error.

The results returned are available in two monitoring views as described below under Statistics Service - Results.

> **Note**
>
> Some check actions are contained within others and are therefore not explicitly executed when you execute the CHECK action. Repair actions make changes to the data and are excluded from the CHECK action.

Lower case characters and special characters in schema and table names must be enclosed in double quotes. The syntax, for example, for a table named "ABC/abc" in the SYSTEM schema must be as follows:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK', 'SYSTEM', '"ABC/abc"');
```

Results

Results of all consistency checks, whether executed manually or scheduled in the statistics service, are available in two monitoring views:

- `M_CONSISTENCY_CHECK_HISTORY`
- `M_CONSISTENCY_CHECK_HISTORY_ERRORS`

The first view gives high level information about parameters used for the check, runtime duration and the total number of tables with errors; each run is identified by a CHECK_EXECUTION_ID value. The errors table lists and gives details of all errors found within each check run. If errors are found you may wish to contact SAP Support to analyze the results and advise on any required action.

In order to avoid storing redundant data, if a manually executed run is repeated by the same user and returns the same results the existing entry in the history table is updated with statistical information instead of creating a new entry for each run. The history table contains statistical information of all related runs, including minimum/maximum/average runtime, number of invocations, timestamp of first/last invocation.

To save disk space you can also clear these tables by executing the following statement:

```
ALTER SYSTEM CLEAR CONSISTENCY CHECK HISTORY [UNTIL <timestamp>]
```

This statement cleans up all entries in the SYS.CONSISTENCY_CHECK_HISTORY_ and SYS.CONSISTENCY_CHECK_HISTORY_ERRORS_ tables up to the specified point in time.

Configuration

A set of ini parameters in the indexserver.ini file is available to control the command line table consistency check. These include, for example: startup behavior, timeout values, and 'smart' job scheduling parameters to skip large jobs which may severely impact performance. These are described in detail in a separate subsection.

Two SAP Notes on consistency checks are available: 1977584 and an FAQ Note 2116157.

Table Consistency Checks in the Statistics Service

You are recommended to schedule the table consistency check so that it runs automatically at regular intervals. The frequency depends on your scenario.

Table consistency checking can be scheduled in the embedded statistics service using collector `_SYS_STATISTICS.Collector_Global_Table_Consistency`. Run-time parameters are maintained as key-value pairs in the `_SYS_STATISTICS.STATISTICS_PROPERTIES` table and the results of the check (details of any errors which are found) are available in the statistics view `GLOBAL_TABLE_CONSISTENCY`. The statistics server also includes a configurable Table Consistency alert (#83) which checks the number of errors and affected tables detected by the consistency check.

The following property values (run-time parameters) can be defined:

Key	Default Value
internal.table_consistency.check_actions	check_variable_part_sanity, check_data_container, check_variable_part_double_reference_global, check_partitioning, check_replication, check_table_container_no_load
internal.table_consistency.target_schema	NULL (all schemas)
internal.table_consistency.target_table	NULL (all tables)
internal.table_consistency.max_duration	0 (no maximum duration)

Note that by default a relatively small number of check actions is carried out to avoid overloading the system, however, the keyword 'check' is available here which acts as a wildcard to execute all check actions. To activate all check actions update this value as shown here.

```
update _SYS_STATISTICS.STATISTICS_PROPERTIES set value = ‘check’ where key =
‘internal.table_consistency.check_actions’
```

The processing sequence for column store tables (row store tables are typically much smaller) is done in strict rotation based on the value of a last check timestamp (oldest first). This ensures that even if the check is canceled before it completes, all tables will eventually be checked over a number of procedure runs.

Result

Results of all consistency checks, whether executed manually or scheduled in the statistics service, are available in two monitoring views as described above.

Disabling Automatic Checks

You can temporarily disable the statistics collector and alert by executing the following statements:

```
CALL CHECK_TABLE_CONSISTENCY('SET_COLLECTOR_SCHEDULE', 'status', 'inactive')
CALL CHECK_TABLE_CONSISTENCY('SET_ALERT_SCHEDULE', 'status', 'inactive')
```

You can re-enable the statistics collector and alert by repeating these calls and setting the 'inactive' value to 'idle'.

Related Information

SAP Note 1977584

SAP Note 2116157

(Deprecated) CALL CHECK_TABLE_CONSISTENCY Statement (Multistore Table) [Dynamic Tiering]

Configuration Parameters for the Table Consistency Check [page 287]

8.11.10.2 Catalog Consistency Check

The catalog consistency check can be run from the command line or be scheduled at the operating system level to perform a range of consistency check actions on the database catalog. The frequency with which you do this depends on your scenario.

> → Recommendation
>
> Do not simultaneously run the catalog check and perform DDL operations (for example, dropping users) since this may cause the check to return multiple errors. Either run the catalog check on the system copy or wait until other operations have completed. Only if you continue to receive errors should you contact SAP Support.

Manual Execution

To execute this procedure, you must have the system privilege CATALOG READ (or DATA ADMIN).

The syntax of the table consistency check call is as follows:

```
CALL CHECK_CATALOG
('<action>','<schema_name>','<object_name>','<catalog_object_type>')
```

The `action` parameter specifies the check action(s) to be performed.

To see details of all check actions which relate to catalog consistency, including a description of what they do, call the procedure **`GET_CHECK_ACTIONS`**:

```
CALL GET_CHECK_ACTIONS('CHECK_CATALOG')
```

Use the parameter `action` to define one or more specific check actions, or enter **`CHECK`** as the value to execute all available actions. Use the parameters `schema_name` and `object_name` to define specific schemas and objects to check, or enter **`NULL`** as the value for these parameters to check all objects in all schemas.

Specify **`NULL`** as the value for the parameter `catalog_object_type`. This parameter is not currently effective and is reserved for future use.

> ✤ Example
>
> To perform all check actions on all objects of all types, execute the statement:
>
> ```
> CALL CHECK_CATALOG ('CHECK', NULL, NULL, NULL)
> ```
>
> Object names are case sensitive and this example shows the use of quotation marks to submit a lower-case table name:
>
> ```
> CALL CHECK_CATALOG ('CHECK', 'SYSTEM', '"mytest"', 'TABLE');
> ```

Result

If errors are found the procedure returns a set of results with the following columns: SCHEMA, NAME, OBJECT_TYPE, ERROR_CODE, ERROR_MESSAGE.

If errors are found, you may wish to contact SAP Support to analyze the results and advise on the required action.

8.11.10.3 Configuration Parameters for the Table Consistency Check

A set of configuration parameters in the indexserver.ini file is available to control the manual table consistency check.

The configuration parameters that control startup behavior are in the `metadata` and `row_engine` sections of the configuration file. Other parameters for timeout values and parameters to skip large jobs which may severely impact performance are available in the `table_consistency_check` section.

[metadata]

Parameter	enable_startup_consistency_check
Short Description	Enable/disable metadata consistency check during SAP HANA startup
Full Description	The metadata consistency check during startup includes CHECK_CATALOG is executed with check action CHECK_OBJECT_REFERENTIAL_INTEGRITY.
Type	Boolean
Change	Offline
Default	True

[row_engine]

Parameter	consistency_check_at_startup
Short Description	Configure row store consistency check during SAP HANA startup.
Full Description	This parameter is used to configure a row store consistency check via CHECK_TABLE_CONSISTENCY during SAP HANA startup. It's a list parameter and the allowed values are 'table', 'page' and 'index', which perform consistency checks on 'table', 'page' and 'index' respectively. This consistency check can be disabled by setting the parameter value to 'none'.
Type	List of strings
Change	Offline

Parameter	consistency_check_at_startup
Default	table,page,index

Parameter	startup_consistency_check_timeout
Short Description	Maximum duration of consistency check at SAP HANA startup
Full Description	This parameter controls the maximum duration of the row store consistency check executed during SAP HANA startup (see `consistency_check_at_startup`).
Type	Integer
Unit	Second
Change	Offline
Default	600

[table_consistency_check]

Parameter	check_max_concurrency_percent
Short Description	Maximum concurrency for table consistency check
Full Description	This parameter controls the overall CPU and thread resource consumption of CHECK_TABLE_CONSISTENCY, defined as a percentage of `max_concurrency_hint` in the global.ini file which is the general limit for the number of concurrently running threads.
Type	Integer
Range	1-100
Change	Online
Default	80

Parameter	enable_table_consistency_check_trace
Short Description	Enable/disable table consistency check tracing
Full Description	This parameter controls if the output of CHECK_TABLE_CONSISTENCY is dumped to trace files with the following naming convention: <service>_<host>.<port>.table_consistency_check.<timestamp>.trc
Type	Boolean
Change	Online

Parameter	enable_table_consistency_check_trace
Default	True

Parameter	large_job_threshold
Short Description	Threshold for large jobs
Full Description	The parameters `large_job_threshold` and `max_num_large_jobs` can be used to make sure that not too many large tables are checked in parallel. Tables exceeding the defined number of rows (default: 100 million) are considered as large tables.
Type	Integer
Unit	Number of rows
Change	Online
Default	100000000

Parameter	max_duration
Short Description	Define the maximum duration of a CHECK_TABLE_CONSISTENCY call
Full Description	This parameter controls the maximum duration of a CHECK_TABLE_CONSISTENCY call. After the specified time the CHECK_TABLE_CONSISTENCY call is implicitly canceled. The default value 0 refers to unlimited duration.
Type	Integer
Unit	Seconds
Change	Online
Default	0

Parameter	max_num_large_jobs
Short Description	Maximum number of large jobs running in parallel
Full Description	The parameters `large_job_threshold` and `max_num_large_jobs` can be used to make sure that not too many large tables are checked in parallel. This parameter controls the maximum number of large tables being checked at the same time.
Type	Integer
Change	Online
Default	4

Parameter	max_result_entry
Short Description	Maximum number of generated errors per node
Full Description	Maximum number of generated errors per SAP HANA node; if more errors are found, the following information is provided: <error_count> errors were found in total. <error_count - max_result_entry> errors are suppressed.
Type	Integer
Change	Online
Default	1000000

Parameter	max_result_entry_per_entity
Short Description	Maximum number of generated error per table
Full Description	Maximum number of generated error per table (partition). If more errors are found, the following information is provided: <error_count> errors were found in this table(partition). <error_count - max_result_entry_per_entity> errors are suppressed
Type	Integer
Change	Online
Default	1000

Parameter	remote_check_timeout
Short Description	Maximum wait time for remote checks
Full Description	CHECK_TABLE_CONSISTENCY terminates if checks on remote tables on secondary node aren't finished within the specified time range. The default value of 86400000 milliseconds refers to a maximum duration of one day.
Type	Integer
Unit	Millisecond
Change	Online
Default	86400000

8.11.11 Row Store Reorganization

Row store reorganization may be required from time to time to recover data fragments in memory; you can carry out reorganization either online or offline.

Row store reorganization recovers data fragments in memory that result from the way in which memory segments and pages are allocated. During reorganization fragmented pages are moved to other segments and the resultant empty segments are freed.

You can run the process online when the database is running or offline when the database is restarted:

- The offline process is run by setting the values of a series of 'row_engine' configuration file parameters in the indexserver.ini file and then restarting the database. Generally, a better compaction ratio is achieved by running reorganization offline as there is no transaction update activity.
- The online process can be executed either by using the SQL system management statement RECLAIM DATA SPACE, or by executing the stored procedure REORGANIZE_ROWSTORE(). These processes do not require exclusive access to tables, this means that tables can be updated while the reorganization process is running.

The stored procedure REORGANIZE_ROWSTORE() can also be used to estimate the benefit of offline row store reorganization as described below.

SAP Note 1813245 - *SAP HANA DB: Row Store Reorganization* gives details of all prerequisites and the preparation steps required before running either online or offline reorganization and gives details of the configuration changes required for the offline process.

SQL RECLAIM DATA SPACE

The syntax for the SQL statement is as follows:

```
ALTER SYSTEM RECLAIM [ROW] DATA SPACE [IMMEDIATE] [<host_port>]
```

For reorganization on a remote node optional parameter values can be provided to specify the host and the port of the server where reorganization is to be carried out. If these are omitted reorganization is carried out on the server where the current connection is made.

Refer to the *SAP HANA SQL and System Views Reference* for further details of the RECLAIM DATA SPACE statement.

Stored Procedure REORGANIZE_ROWSTORE()

An advantage of using the stored procedure in preference to the SQL statement is that the stored procedure automatically distributes the command to all running nodes. The basic syntax of the stored procedure is:

```
CALL REORGANIZE_ROWSTORE (‘REORGANIZE’)
```

With the offline mode you can firstly estimate the benefit of row store reorganization and assess how much memory might be recovered; use the syntax illustrated in the following examples.

Offline Mode Estimate

The syntax for the estimate option is as follows:

```
CALL REORGANIZE_ROWSTORE ('ESTIMATE_MEMORY_SAVING', 'OFFLINE');
```

The procedure returns the estimated saved memory size in MB, the estimated moved page size in MB and the estimated moved page count.

Reorganize a Schema / Table / Segments

The syntax for the reorganize function is given here, the procedure can also be run for a specific named schema and table or for a specific segment:

```
CALL REORGANIZE_ROWSTORE (‘REORGANIZE’, ‘TABLE’, ‘MySchema’, [‘<TABLE_NAME>’])
```

If the optional parameter TABLE_NAME is not given, all row store tables in the specified schema will be reorganized.

Use the SEGMENTS key word to identify one or more segments to reorganize. In this case a json string is required to specify the segments as shown in the following example:

```
CALL REORGANIZE_ROWSTORE(‘REORGANIZE’, ‘SEGMENT’, ‘[{"location”: ”host1:port1”,
"segments”: [3, 4]}, {"location”: "host2:port2”, "segments”: [5, 6, 7]}]”)
```

This reorganizes segment 3, 4 on ‘host1:port1’ and segment 5, 6, 7 on ‘host2:port2’

The procedure also includes a documentation parameter: `REORGANIZE_ROWSTORE ('HELP')`

Related Information

SAP Note 1813245

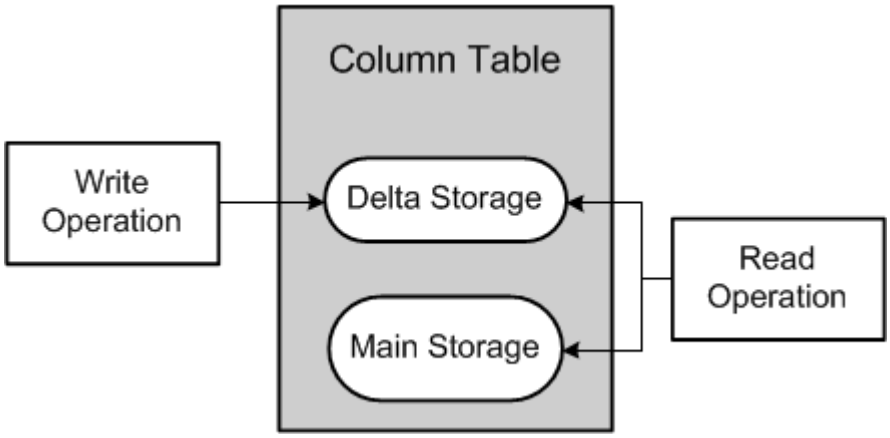
8.11.12 Memory Management in the Column Store

The column store is the part of the SAP HANA database that manages data organized in columns in memory. Tables created as column tables are stored here.

The column store is optimized for read operations but also provides good performance for write operations. This is achieved through 2 data structures: main storage and delta storage.

The main storage contains the main part of the data. Here, efficient data compression is applied to save memory and speed up searches and calculations. Write operations on compressed data in the main storage would however be costly. Therefore, write operations do not directly modify compressed data in the main storage. Instead, all changes are written to a separate data structure called the delta storage. The delta storage uses only basic compression and is optimized for write access. Read operations are performed on both structures, while write operations only affect the delta.

Main Storage and Delta Storage



The purpose of the delta merge operation is to move changes collected in the delta storage to the read-optimized main storage. After the delta merge operation, the content of the main storage is persisted to disk and its compression recalculated and optimized if necessary.

A further result of the delta merge operation is truncation of the delta log. The delta storage structure itself exists only in memory and is not persisted to disk. The column store creates its logical redo log entries for all operations executed on the delta storage. This log is called the delta log. In the event of a system restart, the delta log entries are replayed to rebuild the in-memory delta storages. After the changes in the delta storage have been merged into the main storage, the delta log file is truncated by removing those entries that were written before the merge operation.

> ⓘ Note
>
> As only data in memory is relevant, the load status of tables is significant. A table can have one of the following load statuses:
>
> - Unloaded, that is, none of the data in the table is loaded to main memory
> - Partly loaded, that is, some of the data in the table is loaded to main memory, for example, a few columns recently used in a query
> - Fully loaded, that is, all the data in the table is loaded into main memory

However, data that is in the delta storage can only be fully loaded or unloaded. Partial loading is not possible. Therefore, if a delta merge has not been performed and the table's entire data is in the delta storage, the table is either fully loaded or unloaded.

Loading and Unloading of Data in the Column Store

The SAP HANA database aims to keep all relevant data in memory. Standard row tables are loaded into memory when the database is started and remain there as long as it is running. They are not unloaded. Column tables, on the other hand, are loaded on demand, column by column when they are first accessed. This is sometimes called lazy loading. This means that columns that are never used are not loaded and memory waste is avoided.

> ⓘ Note
>
> This is the **default** behavior of column tables. In the metadata of the table, it is possible to specify that individual columns or the entire table are loaded into memory when the database is started.

The database may actively unload tables or individual columns from memory, for example, if a query or other processes in the database require more memory than is currently available. It does this based on a least recently used algorithm. This unload mechanism can be combined with manually-applied unload priority values. The priority value is applied to a table as a number between 0 and 9; tables with a higher priority are unloaded earlier than other tables.

For more information you can also refer to SAP Note 2127458 *FAQ: SAP HANA Loads and Unloads*

You can also configure columns to allow access to the main storage one page at a time instead of requiring the whole column to be in memory. This enables you to save memory and query a single value in the main storage when certain individual columns or the entire table reside on disk. To enable this feature, specify column description clauses PAGE LOADABLE or COLUMN LOADABLE in the `<column_desc>` of a CREATE TABLE or ALTER TABLE statement.

Related Information

CREATE TABLE Statement (Data Definition)
The Delta Merge Operation [page 301]
SAP Note 2127458

8.11.12.1 Load/Unload a Column Table into/from Memory

Under normal circumstances, the SAP HANA database manages the loading and unloading of tables into and from memory automatically, the aim being to keep all relevant data in memory. However, you can manually load and unload individual tables, as well as load table columns if necessary.

Prerequisites

You have one of the following privileges:

- System privilege TABLE ADMIN
- SQL object privilege UPDATE for the table or the schema in which the table is located

Context

As the SAP HANA database automatically manages the loading and unloading of tables it is not normally necessary to manually load and unload individual tables and table columns. However, this may be necessary for example:

- To precisely measure the total or “worst case” amount of memory used by a particular table (load)
- To actively free up memory (unload)

Load and Unload a Table Using Administration Tools

Context

The procedure is the same in both SAP HANA studio and database explorer using the context menu options available for a selected table in the catalog. In SAP HANA studio you can see detailed information about a table's current memory usage and load status on the *Runtime Information* tab of the table definition. See also monitoring view M_CS_TABLES below.

Procedure

1. In the *Systems* view, navigate to the table in the catalog.
2. In the context menu of the table, choose *Load into Memory* or *Unload from Memory* as required.
3. To start the operation accept the confirmation prompt which is displayed.

Results

If you loaded a table, the complete data of the table, including the data in its delta storage, is loaded into main memory. Depending on the size of the table, this may take some time. The table's load status is FULL.

If you unloaded a table, the complete data of the table, including the data in its delta storage, is unloaded from main memory. Subsequent access to this table will be slower as the data has to be reloaded into memory. The table's load status is NO.

Load and Unload a Table Using SQL

Procedure

1. Open the SQL console and execute the required statement. The LOAD statement supports options to specify one or more table columns, just delta data or all data. You can query the monitoring view M_CS_TABLES for full details of the table (example below).
   - Load MyTable into memory: `LOAD MyTable ALL;`
   - Load columns A and B of MyTable into memory: `LOAD MyTable (A, B);`
   - Unload MyTable from memory: `UNLOAD MyTable;`
2. Query the load status of MyTable: `SELECT loaded FROM m_cs_tables WHERE table_name = 'MyTable';`

Results

If you load only selected columns then the table's load status is PARTIALLY. If you unload a table, the complete data of the table, including the data in its delta storage, is unloaded from main memory.

Related Information

Memory Sizing [page 177]
Memory Management in the Column Store [page 292]
Table Definition
LOAD Statement (Data Manipulation)
UNLOAD Statement (Data Manipulation)

8.11.12.2 Managing Memory by Object Usage

You can use the Unused Retention Period feature to automatically unload objects from memory which are not being used.

SAP HANA has a built-in memory management system which automatically unloads swappable resources from memory if the level of available memory gets too low. An additional method for managing memory is the Unused Retention Period, this automatically unloads objects from memory which are not being used.

> **Note**
>
> You can check the total number of objects and the usage of swappable and non-swappable size using the view M_MEMORY_OBJECTS.

Unused Retention Period

To proactively manage memory, even if no low memory situation is present, you can automatically unload swappable objects from memory on the basis of how frequently objects are used. The time-based parameter `unused_retention_period`. is available for this in the global.ini file.

To use this feature change the default value (initially set to 0) to a number of seconds, for example 7200 (2 hours). Objects which are not used within this time period are flagged as being eligible to unload.

Objects which have exceeded the retention period are not immediately unloaded from memory, an additional checking process which runs at a pre-defined interval initiates the unload. The frequency of the check is controlled by the `unused_retention_period_check_interval` configuration parameter. This is set by default to 7200 seconds (2 hours).

Retention Times and Priorities for Objects

In addition to the general configuration parameter you can apply a retention period and an unload priority value to the table or partition definition itself (see the *SAP HANA SQL and System Views Reference* for details). The following example applies an unload retention period of 60 seconds to table `myTable`.

```
ALTER TABLE "myTable" WITH PARAMETERS('UNUSED_RETENTION_PERIOD'='60')
```

Note that retention period values can only be applied to tables if a configuration value has been set for `unused_retention_period` in the global.ini file.

The unload priority value is a number from 0 to 9 where 0 means the object can never be automatically unloaded and 9 means the earliest unload. The following example reduces the default value of 5 to 2:

```
ALTER TABLE "myTable" UNLOAD PRIORITY 2
```

The unload priority of a table is saved in the TABLES table:

```
SELECT UNLOAD_PRIORITY FROM TABLES WHERE TABLE_NAME = "myTable"
```

> **Note**
>
> Note that changes in unload priority are not immediately effective; the change only takes effect the next time the table is loaded into memory. Therefore if you want to apply the change immediately you should unload and reload the table with the following two statements:
>
> ```
> UNLOAD "myTable";
> ```
>
> ```
> LOAD "myTable" ALL;
> ```
>
> As these operations may take a long time for large tables, consider running these jobs at a suitable time outside business hours.

Resources have a retention disposition weighting value which also influences the sequence in which objects are unloaded from memory. The weighting is used with the time value since the last access of a resource to calculate a disposition value. Priorities 6-9 correspond to 'early unload' disposition, priorities 1-5 correspond to a 'long term' disposition, and tables with a priority of zero are 'non swappable'. Disposition values are configurable by a set of parameters in the `memoryobjects` section of the global.ini file although this is not normally necessary. See also SAP Note 1999997 - FAQ: SAP HANA Memory.

Related Information

SAP Note 1999997

8.11.12.3 Hybrid LOBs (Large Objects)

You can choose to store data for large objects either on disk, or in memory, or you can use the flexible hybrid LOB feature which determines how an object is stored depending on its size.

SAP HANA can store large binary objects (LOBs) such as images or videos on disk rather than inside column or row structures in main memory. An LOB saved on disk is referenced only by an ID in the corresponding table column and is loaded into memory on demand. This significantly reduces main memory consumption and therefore affects start up and takeover times.

Note that LOB data has a short-term disposition setting and if there are memory shortages it is removed from memory before column or table data needs to be unloaded (see Cache Consumption below).

The basic options for managing storage of LOBs are:

- Save all LOBs in memory (LOB size of up to 1GB supported).
- Save all LOBs on disk (a virtual file is created per LOB).
- Use the hybrid feature which uses three storage types for LOBs depending on their size:
  - Small objects up to 1000 bytes are saved in memory.
  - Medium sized objects up to 16000 bytes are 'packed', that is, saved in a single LobContainer and loaded into memory when required.
  - Large objects above this threshold are stored in individual virtual files on disk and loaded into memory when required.

When a new LOB is added to the database its size is assessed and the thresholds are applied to determine how the object is stored.

The packed storage type reduces the number of small LOBs being represented as single files. This helps to optimize restart times since the need to load and initialize many LOBs during a system restart has a considerable negative impact on the restart time. It may be particularly advantageous for scenarios using persistent memory and the fast restart option. The hybrid LOB feature is available in the row and column store but the packed type is only implemented for column store tables.

The table consistency check offers a generic check for LOBs (CHECK_LOBS option). Additionally check and repair options are available for packed LOBs:

- CHECK_HYBRID_LOB_OVERHEAD checks all packed LOB containers for a high ratio of physical size to binary size.
- REPAIR_HYBRID_LOB_OVERHEAD iterates over all specified MidSizeLobContainers and removes empty pages.

SQL Statements

The default thresholds for hybrid LOBs are preconfigured, however, it is also possible to apply a user-defined memory threshold value for a table or column using the ALTER or CREATE TABLE statements by specifying details for the LOB data type. Values applied from the SQL command line take precedence over default system settings. Some examples are given here; refer to the `<lob_data_type> [<memory_threshold_value>]` clause for the CREATE and ALTER Table statements for further details.

An additional option to reorganize storage of LOB data is to use the ALTER TABLE `<lob_reorganize>` clause, see example below.

Note that the LOAD UNIT setting for a column or table (<table_load_unit_clause> COLUMN LOADABLE or PAGE LOADABLE) determines how the whole table or column is handled not the individual objects. Refer also to the Native Storage Extension topic *SAP HANA NSE and Paged Large Object Data Types*.

Memory Threshold Values

If you create a table with no memory threshold, the system settings for hybrid LOBs are applied:

```
create column table <table> (id int, data blob);
```

A memory threshold of zero saves all LOBs on disk and a memory threshold of null saves all LOBs in memory:

```
create column table <table> (id int, data blob memory threshold 0);
```

```
create column table <table> (id int, data nclob memory threshold null);
```

Setting a value (unit bytes) for the memory threshold applies this to the column. A value of 1000, for example, means that all lobs <= 1000 bytes are stored in memory and larger lobs are on disk:

```
create column table <table> (id int, data clob memory threshold 1000);
```

The `memory threshold` value is interpreted as smaller or equal (<=).

LOB Reorganize

The following example uses the reorganize clause to apply the default settings for two specified columns of a table:

```
ALTER TABLE 'MyTable' LOB REORGANIZE ('Attachments', 'Pictures')
```

If no columns are specified the thresholds are applied to all LOB columns in the table.

The LOB REORGANIZE clause also supports an ONLINE mode (or ONLINE PREFERRED) which does not require an exclusive lock on tables so that data updates can still be applied. See the ALTER TABLE `<lob_reorganize>` clause in the *SAP HANA SQL Reference Guide* for details.

Memory Consumption per Table

Even when LOB data is stored on disk you still need to store some information in memory. This information can be retrieved by querying M_TABLES, M_CS_TABLES or M_CS_COLUMNS:

```
SELECT * FROM M_CS_TABLES WHERE SCHEMA_NAME = '<schema>' ORDER BY
MEMORY_SIZE_IN_TOTAL DESC;
SELECT * FROM M_CS_COLUMNS WHERE SCHEMA_NAME = '<schema>' AND TABLE_NAME =
'<table>' ORDER BY MEMORY_SIZE_IN_TOTAL DESC;
```

System HEAP_MEMORY

To see how much main memory is consumed by hybrid LOB data stored on disk that is actually loaded into main memory, you can read memory allocator statistics data from the view M_HEAP_MEMORY. The allocator is identified in the view as the Category value; a number of allocators are available, for example:

```
SELECT * FROM M_HEAP_MEMORY WHERE CATEGORY = 'Pool/PersistenceManager/
LOBContainerDirectory';
```

Available allocators are listed here; the full path begins with `Pool/PersistenceManager/` as shown in the above example.

Allocator	Content
LOBContainerDirectory	Information about persistent LOB containers stored on disk.

Allocator	Content
ContainerFileIDMapping	Mapping information for fast access to persistent LOB containers and other non-LOB containers.
MidSizeLOBContainerDirectory	Information about persistent midsize LOB containers stored on disk.
MidSizeLOBContainerFileIDMapping	Transient mapping structures for fast access to midsize LOBs.
PersistentSpace/DefaultLPA/Page	Page cache allocations among others including also LOBs.
PersistentSpace/DefaultLPA/LOBPage	Page cache allocations dedicated for LOBs only.

Cache Consumption

To speed up access when LOBs are stored on disk, LOB data is cached inside SAP HANA page cache with short term disposition.

> **Note**
>
> Do not change this as it will cause performance issues.

During high load HEAP_MEMORY might increase significantly (until SAP HANA's general memory limit is reached). This is no problem as LOB data is unloaded first from the page cache as it uses short term disposition.

For memory analysis the cache may be cleaned or SAP HANA is restarted in order to free caches. Both options should be used carefully as this unloads all tables and reload might be expensive (meaning it may require a downtime).

For an overview of the cache use: `SELECT * FROM M_MEMORY_OBJECT_DISPOSITIONS`

M_MEMORY_OBJECT_DISPOSITIONS shows which component holds which kind of disposition resource (whether the memory objects are short, mid, long-term or non-swappable). It does not tell you what data is stored in cached pages.

Related Information

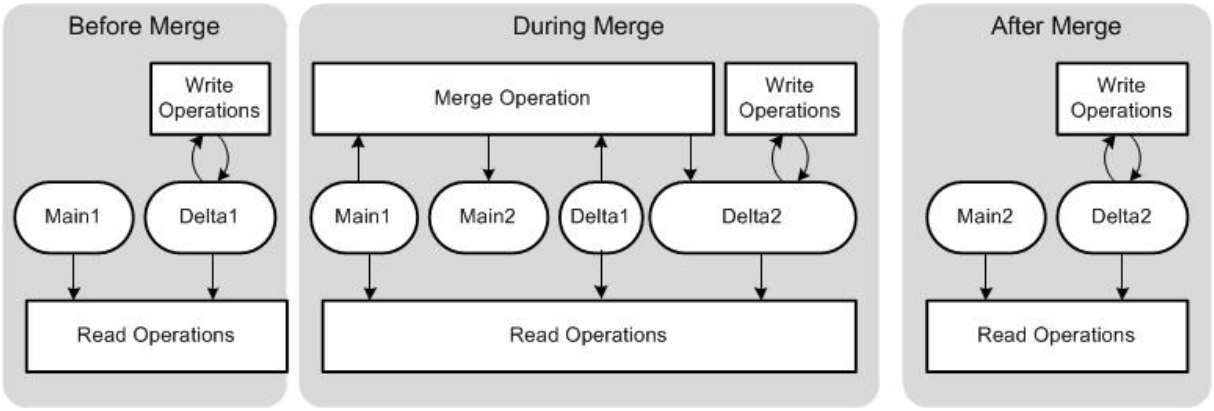
CREATE TABLE Statement (Data Definition)
ALTER TABLE Statement (Data Definition)
M_MEMORY_OBJECT_DISPOSITIONS System View
M_TABLE_LOB_FILES System View
M_HEAP_MEMORY System View
SAP HANA NSE and Paged Large Object Data Types [page 224]
Table and Catalog Consistency Checks [page 282]

8.11.13 The Delta Merge Operation

Write operations are only performed on the delta storage. In order to transform the data into a format that is optimized in terms of memory consumption and read performance, it must be transferred to the main storage. This is accomplished by the delta merge operation.

The following figure shows the different steps in the merge process, which objects are involved, and how they are accessed:

The Delta Merge Process



1. Before the merge operation, all write operations go to Delta 1 storage and all read operations read from Main 1 and Delta 1 storages.
2. While the merge operation is running, the following happens:
   1. All write operations go to the second delta storage, Delta 2.
   2. Read operations read from the original main storage, Main 1, and from both delta storages, Delta 1 and Delta 2.
   3. The content of Main 1 and Delta 1 are merged into the new main storage, Main 2.
3. After the merge operation has completed, the following happens:
   1. Main1 and Delta1 storages are deleted.
   2. The content of the complete main storage is persisted to disk.

> **Note**
>
> With this double buffer concept, the table only needs to be locked for a short time: at the beginning of the process when open transactions are moved to Delta2, and at the end of the process when the storages are “switched”.

> **Caution**
>
> The minimum memory requirement for the delta merge operation includes the current size of main storage plus future size of main storage plus current size of delta storage plus some additional memory. It is important to understand that even if a column store table is unloaded or partly loaded, the whole table is loaded into memory to perform the delta merge.

The delta merge operation can therefore be expensive for the following main reasons:

- The complete main storages of all columns of the table are re-written in memory. This consumes some CPU resources and at least temporarily duplicates the memory needed for the main storages (while Main 1 and Main 2 exist in parallel).
- The complete main storages are persisted to disk, even if only a relatively small number of records were changed. This creates disk I/O load.

This potentially negative impact on performance can be mitigated by splitting tables. The size of the main storage can be reduced by splitting the table into multiple partitions, each with its own main and delta storages. The delta merge operation is performed at partition level and only for partitions that actually require it. This means that less data needs to merged and persisted. Note that there are disadvantages to partitioning tables that should also be considered.

Delta Merge on Partitioned Tables

During the delta merge operation, every partition of a partitioned table is treated internally as a standalone table with its own data and delta store. Only the affected partitions are subject to the merge operation. As described above, the whole table has to be duplicated during the merge operation, so for partitioned tables, the amount of needed main memory during the merge operation is reduced, depending on the size of the partition.

A table can be repartitioned independently of whether it has data in main or delta storage. However, repartitioning can be faster if the delta is small.

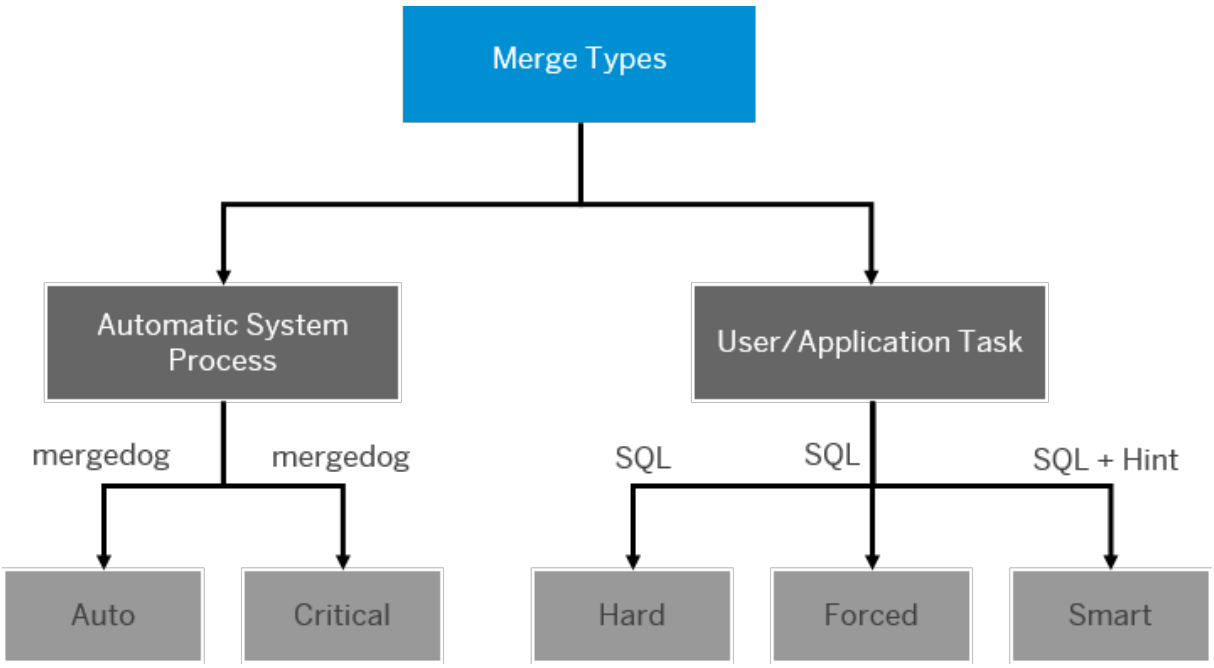
Related Information

Table Partitioning [page 317]

8.11.13.1 Merge Motivations

The request to merge the delta storage of a table into its main storage can be triggered in several ways. These are called merge motivations.

The following figure illustrates the different merge motivations and how they are triggered.



Merge Motivations

Auto Merge

The standard method for initiating a merge in SAP HANA is the auto merge. A system process called mergedog periodically checks the column store tables that are loaded locally and determines for each individual table (or single partition of a split table) whether or not a merge is necessary based on configurable criteria (for example, size of delta storage, available memory, time since last merge, and others).

Auto merge is active if the `active` parameter in the mergedog section of the `indexserver.ini` file is set to `yes`.

> ### Note
>
> Auto merge can be activated and deactivated for an individual table in the system view TABLES (SYS). The value in the column AUTO_MERGE_ON can be changed to TRUE or FALSE.

> ### Note
>
> Deactivating auto merge for a table implicitly disables automatic compression optimization as well (column AUTO_OPTIMIZE_COMPRESSION_ON set to FALSE). This is the case even if the AUTO_OPTIMIZE_COMPRESSION_ON column is set to TRUE in the system view TABLES. For tables with auto merge disabled, compression optimization must be triggered manually.

Smart Merge

If an application powered by SAP HANA requires more direct control over the merge process, SAP HANA supports a function that enables the application to request the system to check whether or not a delta merge

makes sense now. This function is called smart merge. For example, if an application starts loading relatively large data volumes, a delta merge during the load may have a negative impact both on the load performance and on other system users. Therefore, the application can disable the auto merge for those tables being loaded and send a “hint” to the database to do a merge once the load has completed.

When the application issues a smart merge hint to the database to trigger a merge, the database evaluates the criteria that determine whether or not a merge is necessary. If the criteria are met, the merge is executed. If the criteria are not met, the database takes no further action and only a subsequent hint from the application will trigger another evaluation of the criteria.

Smart merge is active if the `smart_merge_enabled` parameter in the mergedog section of the `indexserver.ini` file is set to `yes`.

Hard Merges

You can trigger the delta merge operation for a table manually by executing the SQL statement `MERGE DELTA OF "<table_name>"`. This is called a hard merge and results in the database executing the delta merge for the table either immediately if sufficient system resources are available, or as soon as sufficient system resources become available. The hard merge is therefore subject to the merge token control mechanism.

> **Note**
>
> Unlike system-triggered delta merge operations, all of the manually-executed delta merge operations listed here do not subsequently trigger an optimization of the compression of the table's new main storage. If the table was compressed before the delta merge operation, it remains compressed with the same compression strategy afterward. If it was not compressed before the delta merge operation, it remains uncompressed afterward. After a manual delta merge, you must therefore trigger compression optimization manually.

Critical Merge

The database can trigger a critical merge in order to keep the system stable. For example, in a situation where auto merge has been disabled and no smart merge hints are sent to the system, the size of the delta storage could grow too large for a successful delta merge to be possible. The system initiates a critical merge automatically when a certain threshold is passed.

Related Information

Perform a Manual Delta Merge Operation [page 311]
Compress a Column Table Manually [page 316]
TABLES System View
SAP Note 2057046

8.11.13.2 The Merge Monitor

The delta merge operation for column tables is a potentially expensive operation and must be managed according to available resources and priority. This is the responsibility of the merge monitor.

The system uses cost functions to decide which table to merge, when, and in which order. There are also cost functions that control how many tables are merged at the same time and how many threads are used to merge a single table.

The merge monitor is responsible for controlling all merge requests for all column tables on a single host. In a distributed system, every index server has its own merge monitor.

All merge requests must acquire a merge token from the merge monitor. A merge token represents an allocation of system resources and "entitles" the merge to actually start. The merge monitor blocks merge requests if there are not enough system resources available or if the same table is already being merged by another thread. This avoids long waits and delays for other threads for inserting or just reading data.

Depending on current system resource consumption, merge motivation, and the evaluation of the various cost functions, the merge monitor lets single requesting merge threads pass and releases waiting threads.

> **Note**
>
> There is no option or need to disable, stop, or even kill the merge monitor. The merge monitor is not a thread.

8.11.13.3 Cost Functions

The SAP HANA database decides whether or not to execute a requested delta merge and the order in which to execute multiple requests based on configurable merge criteria or cost functions.

Cost functions can be configured depending on the merge motivation, that is whether the merge is being requested by the automatic system process mergedog (auto merge), by a hint from the application (smart merge), by SQL statement (hard merge), and so on.

Cost functions are evaluated in runtime and configured in the mergedog section of the `indexserver.ini` file. The following cost functions are available:

- `auto_merge_decision_func` and `smart_merge_decision_func`
  These cost functions determine whether or not a requested delta merge is executed.
- `auto_merge_priority_func` and `smart_merge_priority_func`
  These cost functions determine the priority that is assigned to the delta merge request.
- `critical_merge_decision_func`
  This cost function determines whether or not a delta merge is executed. It will run a delta merge to avoid situations that could lead to an out of memory or system crash even if other cost functions have been turned off or fail to run.
- `hard_merge_priority_func`
  This cost function determines the priority of hard merges.
- `load_balancing_func`
  This cost function determines the allocation of system resources to merge processing.

> **Note**
>
> The decision cost function is evaluated only once for each merge request. In the case of a merge request triggered by a smart merge hint, if the cost function returns a result of false (that is, the system decides that a delta merge is not required), the request is logged but no further evaluation takes place. Only a new hint can potentially initiate a new delta merge.

The following parameters are available for configuring the cost functions. You can use them to build cost functions for all delta merge configurations.

> **Caution**
>
> It is not recommended that you change the default settings for delta merge unless instructed to do so by SAP Support.

Parameter	Meaning
DMS	Delta memory size [MB] This refers to the size of the table's delta storage.
TMD	Table merge delay [sec] This refers to the time since the last merge of table
MRC	Main row count [million] This refers to the current number of rows in the main storage of the table.
DMR	Deleted main rows [million] This refers to the number of deleted records not in delta storage, but marked as deleted in main storage. Merging makes sense if there are many deleted rows.
DLS	Delta log size [MB]
DCC	Delta cell count [million] This refers to the current number of cells in the delta storage of the table. For example, if the delta storage contains 3 records, each with 4 columns, then the delta cell count is 12.
DRC	Delta row count [million] This refers to the current number of rows in the delta storage of the table.
QDW	Queuing delay wait [sec] This refers to the time that a merge thread has been waiting for the merge monitor to allocate it merge tokens. This parameter can be useful if you want to implement a first come first served scheduling strategy.
NAME	Table name [string]
SCHEMA	Schema name [string]
CLA	CPU load average [percentage]
LCC	Logical CPU count
THM	Total heap memory [MB]

Parameter	Meaning
AHM	Available heap memory, including memory that could be freed [MB]
DUC	Delta uncommitted row count [million] This refers to the number of uncommitted rows in the delta storage of the table.
MMS	Main memory size [MB]
UPT	Index server uptime [sec]
MMU	Main max udiv [million]
OCRC	(Last) optimize compression row count [million]
CRCSOC	Change row count since (last) optimize compression [million]
RP	Table is range partitioned [boolean]
PAL	Process allocation limit [MB]

Cost Functions Examples

Cost Function Configuration	Meaning
`auto_merge_decision_func = DMS>1000 or TMD>3601 or DCC>800 or DMR>0.2*MRC or DLS>5000`	An automatic delta merge of a table is executed if : • The size of its delta storage exceeds 1000 MB, or • It has not been merged in over 60 minutes, or • Its delta cell count exceeds 800 million, or • More than 20% of the records in its main storage were deleted, or • The size of its delta log is greater than 5000 MB
`auto_merge_decision_func = DMS > 1000 or DMS > 42 and weekday(now())=6 and secondtime(now())>secondtime('01:00') and secondtime(now())<secondtime('02:00')`	An automatic delta merge of a table is executed if: • The size of its delta storage exceeds 1000 MB, unless • It is Saturday between 1.00 and 2.00, in which case it will be merged if delta storage exceeds 42MB Note the week starts with Monday as day 0.
• `smart_merge_decision_func = DMS>1000 or DCC>800 or DLS>5000` • `smart_merge_priority_func = DMS/1000`	A delta merge request of a table triggered by a smart merge hint is executed if: • The delta storage size exceed 1000 MB, or • The delta cell count in the delta storage is greater than 800 million, or • The size of the delta log is greater than 5000 MB The system prioritizes smart merge requests based on the size of the delta storage, that is, tables with the bigger deltas are merged first.

Cost Function Configuration	Meaning
`hard_merge_priority_func = QDW`	Delta merges triggered by hard merge are prioritized only by queuing delay wait, in other words, on a first in first out basis.
`hard_merge_priority_function = 1/(7+MMS)`	Delta merges triggered by hard merge are prioritized by table size. Smaller tables are merged first, the idea being to free some memory first before bigger tables start merging.

Related Information

Change a System Property in SAP HANA Studio
Data Compression in the Column Store [page 313]
SAP Note 2057046

8.11.13.4 Merge Tokens

The delta merge operation can create a heavy load on the system. Therefore, controls need to be applied to ensure that merge operations do not consume all system resources. The control mechanism is based on the allocation of merge tokens to each merge operation.

With the exception of the forced merge, a merge operation cannot start unless it has been allocated tokens. If all merge tokens are taken, merge requests have to wait either until the system releases new merge tokens because more resources are available, or until merge tokens have been released by completed merge requests. Each merge token represents a single CPU

The number of merge tokens available for allocation is adjusted based on current system resource availability. This number is recalculated periodically by the system based on a cost function configured in the `load_balancing_func` parameter in the mergedog section of the indexserver.ini file. The default configuration is `load_balancing_func = 1 + LCC * (AHM/GAL) * max(100-CLA,0)/100`. If a hard maximum is required for the number of tokens available, you can enter just a constant as a parameter value (for example, LCC).

For every merge request, the number of tokens required to perform the merge is calculated by the system. If the system is not able to determine a value, a default value is returned. This default value can be configured in the `token_per_table` parameter in the mergedog section of the indexserver.ini file. However, it is not recommended that you change this value.

> **Note**
>
> It is not possible to check the number of merge tokens available for allocation at any given time, but it is logged in the indexserver trace file if you activate the indexserver component `mergemonitor` with trace level INFO.

Related Information

Database Trace (Basic, User-Specific, and End-to-End) [page 464]
SAP Note 2057046

8.11.13.5 Monitoring Delta Merge History

Information about all delta merge operations since the last system start are logged in the monitoring view M_DELTA_MERGE_STATISTICS. In addition to completed merge operations, information is available on merge hints received by applications and post-merge compression optimization.

You can access a predefined view of these merge statistics in the statement library of the SAP HANA database explorer or in the SAP HANA studio on the *System Information* tab of the Administration editor.

The following columns contain potentially useful information:

Column	Description
TYPE	Here you can see the type of merge history entry. The following values are possible: • MERGE for an actual delta merge operation • HINT for a merge hint sent to SAP HANA by an application • SPARSE for the post-merge optimization of main storage compression
MOTIVATION	This column identifies the underlying merge motivation: AUTO, SMART, HARD, or FORCE
SUCCESS	This column depends on the entry in the TYPE column. • For MERGE or SPARSE entries, it indicates whether or not the merge or compression optimization operation was successful. • For HINT entries, it indicates whether or not the hint from the application to merge was accepted. If the hint was accepted (SUCCESS=TRUE), then there is an associated entry of type MERGE. If the hint was rejected (SUCCESS=FALSE), then no merge is triggered, so there is no associated MERGE entry. **Note** Even if the hint was accepted (SUCCESS=TRUE), this does not necessarily mean that the subsequent merge was successful. You must check the SUCCESS column of the merge entry.
LAST_ERROR	This column provides information about error codes of the last errors that occurred (most often 2048). Details are provided in ERROR_DESCRIPTION.

Column	Description
ERROR_DESCRIPTION	The following error codes are possible: • Error 2480: The table in question is already being merged. • Error 2481: There are already other smart merge requests for this table in the queue. • Error 2482: The delta storage is empty or the evaluation of the smart merge cost function indicated that a merge is not necessary. • Error 2483: Smart merge is not active (parameter `smart_merge_enabled=no`). • Error 2484: Memory required to optimize table exceeds heap limit (for failed compression optimization operations (TYPE=SPARSE, SUCCESS=FALSE)).
PASSPORT	For entries with the merge motivation SMART, this column identifies the application that sent the hint to merge (for example, SAP BW powered by SAP HANA)

Note

If the index server is restarted, the delta merge history will initially be empty. The system also collects delta merge statistics in the table HOST_DELTA_MERGE_STATISTICS (_SYS_STATISTICS) independent of system restarts. However, as the system only collects statistical data periodically, this table may not have the most recent delta merge operations.

Example

The following is an example of how to use the merge history to find a merge you were expecting to happen based on the settings for triggering smart merge hints in your application.

1. Look for merges triggered by smart merge in the merge history by executing the following SQL statement:
   ```
   SELECT * FROM M_DELTA_MERGE_STATISTICS WHERE table_name = '<your_table>' AND motivation = 'SMART'
   ```
2. If no results are returned, check to see if the application actually sent any hints by executing the following statement:
   ```
   SELECT * FROM M_DELTA_MERGE_STATISTICS WHERE type = 'HINT' AND table_name = '<your_table>'
   ```
   If the application did not send a hint, then the system will not initiate a delta merge. However, if the application did send a hint, the system only executes the merge if the criteria for smart merge are fulfilled. The information is available in the SUCCESS column. The system decides whether or not to accept the hint and execute the merge by evaluating the smart merge decision cost function.
3. If you still have not found the smart merge, check the long-term history by executing the following statement:
   ```
   SELECT * FROM _SYS_STATISTICS.HOST_DELTA_MERGE_STATISTICS WHERE table_name = '<your_table>'
   ```

Tracing

You can activate the logging of merge-related information in the database trace for the indexserver component. The relevant trace components are `mergemonitor` and `mergedog`. We recommend the trace level INFO.

Related Information

M_DELTA_MERGE_STATISTICS System View
View Diagnostic Files in the SAP HANA Database Explorer
View Diagnosis Files in SAP HANA Studio
Configure Traces in SAP HANA Studio
Use the Statement Library to Administer Your Database (SAP HANA Database Explorer)

8.11.13.6 Perform a Manual Delta Merge Operation

You can trigger the delta merge operation for a column table manually, for example, if you need to free up memory.

Prerequisites

You have one of the following privileges:

- System privilege TABLE ADMIN
- SQL object privilege UPDATE for the table or the schema in which the table is located

Context

It may be necessary or useful to trigger a merge operation manually in some situations, for example:

- An alert has been issued because a table is exceeding the threshold for the maximum size of delta storage.
- You need to free up memory. Executing a delta merge operation on tables with large delta storages is one strategy for freeing up memory. The delta storage does not compress data well and it may hold old versions of records that are no longer required for consistent reads. For example, you can use the following SQL statement to retrieve the top 100 largest delta storages in memory:

```
SELECT TOP 100 * from M_CS_TABLES ORDER BY MEMORY_SIZE_IN_DELTA DESC
```

You can trigger the delta merge operation for a column table manually in the SAP HANA studio by menu command or SQL statement. A manually-executed delta merge operation corresponds to a hard merge. However, if you use SQL, you can also pass additional parameters that trigger forced merges and memory-only merges.

Procedure

1. Execute the required merge in one of the following ways:

Option	Description
SAP HANA studio	1. In the *Systems* view, navigate to the table. 2. In the context menu of the table, choose *Perform Delta Merge*. 3. Choose *OK*.
SQL	Open the SQL console and execute the required statement. The first command is for a hard merge, the FORCE REBUILD option executes a hard merge even if no rows from old delta can be merged and no rows from old main can be deleted: • `MERGE DELTA OF '<table_name>'` • `MERGE DELTA OF '<table_name>' FORCE REBUILD`

2. Optional: Confirm the delta merge operation in one of the following ways (SAP HANA studio):
   - Open the table definition from the *Systems* view and on the *Runtime Information* tab, check the relevant values.

     > ⓘ Note
     >
     > Even though the delta merge operation moves data from the delta storage to the main storage, the size of the delta storage will not be zero. This could be because while the delta merge operation was taking place, records written by open transactions were moved to the new delta storage. Furthermore, even if the data containers of the delta storage are empty, they still need some space in memory.

   - Check the merge history by opening the *Merge Statistics* table on the *System Information* tab. The SUCCESS column indicates whether or not the merge operation was executed.

> → Tip
>
> The delta merge operation can take a long time. You can see the progress of delta merge operations currently running in the Administration editor on the *Performance* > *Job Progress* tab.

Results

The delta merge operation is executed.

> ⓘ Note
>
> Unlike system-triggered delta merge operations, manually-executed delta merge operations do not subsequently trigger an optimization of the compression of the table's new main storage. If the table was compressed before the delta merge operation, it remains compressed with the same compression strategy afterward. If it was not compressed before the delta merge operation, it remains uncompressed afterward. After a manual delta merge, you must therefore trigger compression optimization manually.

Related Information

MERGE DELTA Statement (Data Manipulation)

8.11.14 Data Compression in the Column Store

The column store allows for the efficient compression of data. This makes it less costly for the SAP HANA database to keep data in main memory. It also speeds up searches and calculations.

Data in column tables can have a two-fold compression:

- Dictionary compression
  This default method of compression is applied to all columns. It involves the mapping of distinct column values to consecutive numbers, so that instead of the actual value being stored, the typically much smaller consecutive number is stored.
- Advanced compression
  Each column can be further compressed using different compression methods, namely prefix encoding, run length encoding (RLE), cluster encoding, sparse encoding, and indirect encoding. The SAP HANA database uses compression algorithms to determine which type of compression is most appropriate for a column.

> ⓘ Note
>
> Advanced compression is applied only to the main storage of column tables. As the delta storage is optimized for write operations, it has only dictionary compression applied.

Compression is automatically calculated and optimized as part of the delta merge operation. If you create an empty column table, no compression is applied initially as the database cannot know which method is most appropriate. As you start to insert data into the table and the delta merge operation starts being executed at regular intervals, data compression is automatically (re)evaluated and optimized.

Automatic compression optimization is ensured by the parameter `active` in the `optimize_compression` section of the `indexserver.ini` configuration file. This parameter must have the value `yes`.

> ⓘ Note
>
> If the standard method for initiating a delta merge of the table is disabled (AUTO_MERGE_ON column in the system view TABLES is set to FALSE), automatic compression optimization is implicitly disabled as well. This is the case even if the AUTO_OPTIMIZE_COMPRESSION_ON column is set to TRUE in the system view TABLES. It is necessary to disable auto merge if the delta merge operation of the table is being controlled by a smart merge triggered by the application. For more information, see the section on merge motivations.

Compression Factor

The compression factor refers to the ratio of the uncompressed data size to the compressed data size in SAP HANA.

The uncompressed data volume is a database-independent value that is defined as follows: the nominal record size multiplied by the number of records in the table. The nominal record size is the sum of the sizes of the data types of all columns.

The compressed data volume in SAP HANA is the total size that the table occupies in the main memory of SAP HANA.

> ### Example
>
> You can retrieve this information for a fully-loaded column table from the monitoring view M_CS_TABLES by executing the statement: `select SCHEMA_NAME, TABLE_NAME, MEMORY_SIZE_IN_TOTAL from PUBLIC.M_CS_TABLES where SCHEMA_NAME='<schema>' and TABLE_NAME='<table>'`

The compression factor achieved by the database depends on your SAP HANA implementation and the data involved.

For more information see *Cost Functions*

Cost Functions for Optimize Compression

The cost functions for optimize compression are in the optimize_compression section of the service configuration (e.g. indexserver.ini)

- **auto_decision_func** - if triggered by MergeDog
- **smart_decision_func** - if triggered by SmartMerge

Default Cost Function Configuration	Meaning
`MMU > 0.010240 and if(OCRC, max(MRC, OCRC) / min(MRC, OCRC) >= 1.75, 1) and (not RP or (RP and TMD > 86400))`	Optimize compression runs if • The table contains more than 10240 rows AND • (Optimize compression was never run before OR • The number of rows increase or decrease by factor of 1.75) • AND - if range partitioned, the last delta merge happened more than 24 hours ago.

Related Information

SAP Note 1514966
SAP Note 1637145
Merge Motivations [page 302]
M_CS_TABLES System View
TABLES System View
Cost Functions [page 305]
Compress a Column Table Manually [page 316]
Merge Motivations [page 302]

8.11.14.1 Check the Compression of a Column Table

For column-store tables, you can check the type of compression applied to table columns, as well as the compression ratio.

Prerequisites

To check the compression status of a table accurately, ensure that it is first fully loaded into main memory.

Procedure

1. To check the type of compression applied to table columns, execute the following SQL statement:

   ```
   SELECT SCHEMA_NAME, TABLE_NAME, COLUMN_NAME, COMPRESSION_TYPE, LOADED
   from PUBLIC.M_CS_COLUMNS where SCHEMA_NAME='<your_schema>' and
   TABLE_NAME='<your_table>'
   ```

   The columns of the selected table are listed with the type of compression applied. The following values are possible:

   - DEFAULT
   - SPARSE
   - PREFIXED
   - CLUSTERED
   - INDIRECT
   - RLE

   > ### Note
   >
   > Even if the column is not loaded into memory, the compression type is indicated as DEFAULT. This is because there will always be some level of dictionary compression. However, unless the column is loaded, the database cannot determine the type of compression actually applied. The LOADED column indicates whether or not the column is loaded into memory.

2. Check the compression ratio of table columns, that is, the ratio of the column's uncompressed data size to its compressed data size in memory.

   You can do this in the SAP HANA studio:

   a. In the Administration editor, open the table definition in the table editor.
   b. Choose the *Runtime Information* tab.
   c. In the *Details for Table* area, choose the *Columns* tab.

   The compression ratio is specified in the *Main Size Compression Ratio [%]* column.

Related Information

Load/Unload a Column Table into/from Memory [page 294]
M_CS_ALL_COLUMNS System View

8.11.14.2 Compress a Column Table Manually

The SAP HANA database decides which columns in a column table to compress and which compression algorithm to apply for each column. It does this as part of the delta merge operation. It is normally not necessary that you interfere with this process. However, you can trigger compression manually.

Prerequisites

You have the UPDATE privilege for the table.

Context

We do not recommend that you interfere with the way in which the SAP HANA database applies compression. However, if a table is not compressed and you think it should be, you can request the database to reevaluate the situation.

Before you do this, consider the reasons why the table may not be compressed, for example:

- The table is very small.
- The table's delta storage has never been merged with its main storage.
- The table was created and filled using an old version of the SAP HANA database that did not compress data automatically. No further data loads, and consequently no delta merge operations, have taken place.
- The auto merge function has been disabled for the table (AUTO_MERGE_ON column in the system view TABLES is set to FALSE). Deactivating auto merge for a columnstore table implicitly disables the automatic compression optimization as well. This is the case even if the AUTO_OPTIMIZE_COMPRESSION_ON column is set to TRUE in the system view TABLES.

Procedure

1. Request the database to reevaluate compression by executing the SQL statement:

   ```
   UPDATE "<your_table>" WITH PARAMETERS ('OPTIMIZE_COMPRESSION'='YES')
   ```

   The database checks all of the table's columns and determines whether or not they need to be compressed, or whether or not existing compression can be optimized. If this is the case, it compresses the data using the most appropriate compression algorithm. However, note the following:

- The database will only reevaluate compression if the contents of the table have changed significantly since the last time compression was evaluated.
- Even if the database does reevaluate the situation, it may determine that compression is not necessary or cannot be optimized and so changes nothing.

2. Check the compression status of the table.
3. Optional: If compression has not changed, force the database to reevaluate compression by executing the following SQL statement `UPDATE "<your_table>" WITH PARAMETERS ('OPTIMIZE_COMPRESSION'='FORCE')`.
   The database checks all of the table's columns and determines whether or not they need to be compressed, or whether or not existing compression can be optimized. If this is the case, it compresses the data using the most appropriate compression algorithm. Note that the database may still determine that compression is not necessary or cannot be optimized and so changes nothing.
4. Check the compression status of the table.

Related Information

Check the Compression of a Column Table [page 315]
The Delta Merge Operation [page 301]
UPDATE Statement (Data Manipulation)
TABLES System View

8.11.15 Table Partitioning

The partitioning feature of the SAP HANA database splits column-store tables horizontally into disjunctive sub-tables or partitions. In this way, large tables can be broken down into smaller, more manageable parts. Partitioning is typically used in multiple-host systems, but it may also be beneficial in single-host systems.

Partitioning can be done either in SAP HANA cockpit (refer to *SAP HANA Administration with SAP HANA Cockpit*) or at the SQL command line. Partitioning is transparent for SQL queries and data manipulation language (DML) statements. There are additional data definition statements (DDL) for partitioning itself:

- Create table partitions
- Re-partition tables
- Merge partitions to one table
- Add/delete partitions
- Move partitions to other hosts
- Perform the delta merge operation on certain partitions

Privileges

In general, to modify the partitions of a given table, a specific ALTER object privilege for that table is required. However, the PARTITION ADMIN system privilege is available, and a user with this privilege can perform

some partitioning operations with the ALTER TABLE statement; for example: partition a table, move or merge partitions. The complete list of partitioning clauses that can be used with this privilege is given in the corresponding 'Non-Heterogeneous Alter Partition Clauses' and 'Heterogeneous Alter Partition Clauses' of the ALTER TABLE section of the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

About Partitioning

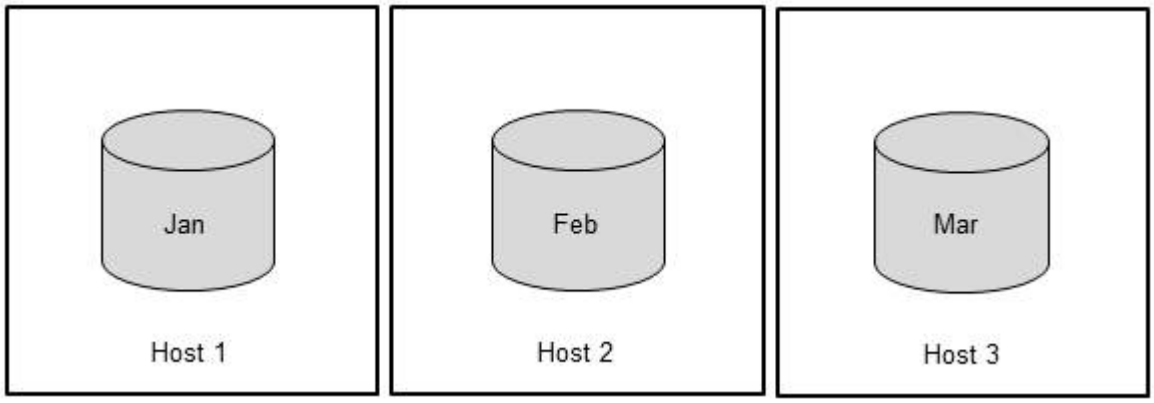
When a table is partitioned, the split is done in such a way that each partition contains a different set of rows of the table. There are several alternatives available for specifying how the rows are assigned to the partitions of a table, for example, hash partitioning or partitioning by range.

The following are the typical advantages of partitioning:

- Load balancing in a distributed system
  Individual partitions can be distributed across multiple hosts. This means that a query on a table is not processed by a single server but by all the servers that host partitions.
- Overcoming the size limitation of column-store tables
  A non-partitioned table cannot store more than 2 billion rows. It is possible to overcome this limit by distributing the rows across several partitions. Each partition must not contain more than 2 billion rows.
- Parallelization
  Partitioning allows operations to be parallelized by using several execution threads for each table.
- Partition pruning
  Queries are analyzed to determine whether or not they match the given partitioning specification of a table (static partition pruning) or match the content of specific columns in aging tables (dynamic partition pruning). If a match is found, it is possible to determine the specific partitions that hold the data being queried and avoid accessing and loading into memory partitions which are not required. See *Static and Dynamic Partition Pruning* for details.
- Improved performance of the delta merge operation
  The performance of the delta merge operation depends on the size of the main index. If data is only being modified on some partitions, fewer partitions will need to be delta merged and therefore performance will be better.
- Explicit partition handling
  Applications may actively control partitions, for example, by adding partitions to store the data for an upcoming month.

The following figure illustrates how a table can be distributed over three hosts with dedicated partitions for individual months.

Example of Table Partitioning



> **Note**
>
> After adding or removing hosts, it is recommended that you execute a redistribution operation. Based on its configuration, the redistribution operation will suggest a new placement for tables and partitions in the system. If you confirm the redistribution plan, the redistribution operation will re-distribute the tables and partitions accordingly.

For more detailed information about the SQL syntax for partitioning, see *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Related Information

The Delta Merge Operation [page 301]
Static and Dynamic Partition Pruning [page 363]
Redistributing Tables in a Scaleout SAP HANA System [page 405]
Partition a Table (in SAP HANA Administration with SAP HANA Cockpit)

Links to SAP HANA SQL Reference Guide for SAP HANA Platform

ALTER TABLE Statement (Data Definition)
Heterogeneous Alter Partition Clauses
Non-heterogeneous Alter Partition Clauses

8.11.15.1 Single-Level Partitioning

When a table is partitioned, its rows are distributed to partitions according to different criteria known as partitioning specifications.

The SAP HANA database supports the following single-level partitioning specifications:

- Round robin
- Hash
- Range

For advanced use cases, these specifications can be nested using multi-level partitioning.

Related Information

Multilevel Partitioning [page 325]

8.11.15.1.1 Hash Partitioning

Hash partitioning is used to distribute rows to partitions equally for load balancing and to overcome the 2 billion row limitation. The number of the assigned partition is computed by applying a hash function to the value of a specified column. Hash partitioning does not require an in-depth knowledge of the actual content of the table.

For each hash partitioning specification, columns must be specified as partitioning columns. The actual values of these columns are used when the hash value is determined. If the table has a primary key, these partitioning columns must be part of the key (note, however, that a hash partitioning that creates only 1 partition does not impose any primary key restriction on the partitioning key columns). The advantage of this restriction is that a uniqueness check of the key can be performed on the local server. You can use as many partitioning columns as required to achieve a good variety of values for an equal distribution.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Hash-Partitioned Table Using SQL

SQL Command	Result
`CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT, PRIMARY KEY (a,b)) PARTITION BY HASH (a, b) PARTITIONS 4`	• Four partitions are created on columns a and b. • The target partition is determined based on the actual values in columns a and b. • At least one column has to be specified. • If a table has a primary key, all partitioning columns must be part of that key.
`CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT, PRIMARY KEY (a,b)) PARTITION BY HASH (a, b) PARTITIONS GET_NUM_SERVERS()`	The number of partitions is determined by the database at runtime according to its configuration. It is recommended to use this function in scripts, and so on.

SQL Command	Result

`CREATE COLUMN TABLE T_HASH_1 (A INT, B VARCHAR(64), C INT) PARTITION BY HASH(A) PARTITIONS 1;`	In this example a single first-level HASH partition is created. Although the data in the table is therefore unpartitioned, the partition exists and can be referenced by its logical partition ID. This ID can be used with other SQL statements which then apply to the whole table. For example, the table can be truncated by referring to the partition ID:
	`TRUNCATE TABLE T_HASH_1 PARTITION(1);`
	The partition ID is also used, for example, in this MOVE PARTITION statement:
	`ALTER TABLE T_HASH1 MOVE PARTITION 1 TO 'host1:port2';`

Data Types

The following data types are allowed for the partitioning column:

Date time Types	DATE, TIME, SECONDDATE, TIMESTAMP
Numeric Types	TINYINT, SMALLINT, INTEGER, BIGINT, DECIMAL, DECIMAL(p,s)
Boolean Type	BOOLEAN
Character String Types	VARCHAR, NVARCHAR
Binary Types	VARBINARY
Text Types	SHORTTEXT

> **Note**
>
> For balanced partitioning TIMESTAMP values using hours are supported, but for heterogeneous partitioning the smallest time unit possible is the whole day - see example in Range Partitioning.

Related Information

CREATE TABLE Statement (Data Definition)
Range Partitioning [page 323]

8.11.15.1.2 Round-Robin Partitioning

Round-robin partitioning is used to achieve an equal distribution of rows to partitions. However, unlike hash partitioning, you do not have to specify partitioning columns. With round-robin partitioning, new rows are assigned to partitions on a rotation basis. The table must not have primary keys.

Hash partitioning is usually more beneficial than round-robin partitioning for the following reasons:

- The partitioning columns cannot be evaluated in a pruning step. Therefore, all partitions are considered in searches and other database operations.
- Depending on the scenario, it is possible that the data within semantically-related tables resides on the same server. Some internal operations may then operate locally instead of retrieving data from a different server.

For Round-Robin and Round-Robin-Range partitioning schemas there is validation of the following DML statements to ensure that the data remains consistent: INSERT, UPDATE, DELETE and UPSERT. For these statements the partition id passed is checked to ensure that the values are consistent with the original round-robin partitioning specification.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Round-Robin Partitioned Table Using SQL

SQL Command	Result
`CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT) PARTITION BY ROUNDROBIN PARTITIONS 4`	Four partitions are created. Note: The table must not have primary keys.
`CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT) PARTITION BY ROUNDROBIN PARTITIONS GET_NUM_SERVERS()`	The number of partitions is determined by the database at runtime according to its configuration. It is recommended to use this function in scripts or clients that may operate in various landscapes.

`CREATE COLUMN TABLE T_ROUNDROBIN_1 (A INT, B VARCHAR(64), C INT) PARTITION BY ROUNDROBIN PARTITIONS 1;`	In this example a single first-level ROUNDROBIN partition is created. Although the data in the table is therefore unpartitioned, the partition exists and can be referenced by its logical partition ID. This ID can be used with other SQL statements which then apply to the whole table. For example, the table can be truncated by referring to the partition ID:
	`TRUNCATE TABLE T_ROUNDROBIN_1 PARTITION(1);`
	The partition ID is also used, for example, in this MOVE PARTITION statement:
	`ALTER TABLE T_ROUNDROBIN_1 MOVE PARTITION 1 TO 'host1:port2';`

Related Information

CREATE TABLE Statement (Data Definition)

8.11.15.1.3 Range Partitioning

Range partitioning creates dedicated partitions for certain values or value ranges in a table. For example, a range partitioning scheme can be chosen to create a partition for each calendar month. Partitioning requires an in-depth knowledge of the values that are used or are valid for the chosen partitioning column.

Partitions may be created or dropped as needed and applications may choose to use range partitioning to manage data at a fine level of detail, for example, an application may create a partition for an upcoming month so that new data is inserted into that new partition.

> ⓘ Note
>
> Range partitioning is not well suited for load distribution. Multi-level partitioning specifications address this issue.

The range partitioning specification usually takes ranges of values to determine one partition (the integers 1 to 10 for example) but it is also possible to define a partition for a single value. In this way, a list partitioning known in other database systems can be emulated and combined with range partitioning.

When rows are inserted or modified, the target partition is determined by the defined ranges. If a value does not fit into one of these ranges, an error is raised. To prevent this you can also define an 'others' partition for any values that do not match any of the defined ranges. 'Others' partitions can be created or dropped on-the-fly as required.

Range partitioning is similar to hash partitioning in that if the table has a primary key on it, the partitioning columns must be part of the key. Many data types are supported for range partitioning, see the list below.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Range-Partitioned Table Using SQL

The following example creates three columns for integers and divides the first column into four partitions. Ranges are defined using this semantic: `<= VALUES <`, ranges for single values use `=`.

- 1 partition for values greater than or equal to 1 and less than 5
- 1 partition for values greater than or equal to 5 and less than 20
- 1 partition for values of 44
- 1 others partition for all other values which do not match the specified ranges

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT, PRIMARY KEY (a,b))
PARTITION BY RANGE (a)
(PARTITION 1 <= VALUES < 5,
PARTITION 5 <= VALUES < 20,
PARTITION VALUE = 44, PARTITION OTHERS)
```

The partitioning column (a in this example) has to be part of the primary key.

Data Types

The following data types are allowed for the partitioning column:

Datetime Types	DATE, SECONDDATE, TIMESTAMP
Numeric Types	TINYINT, SMALLINT, INTEGER, BIGINT, DECIMAL(p,s)
Boolean Type	BOOLEAN
Character String Types	VARCHAR, NVARCHAR
Binary Types	VARBINARY
Text Types	SHORTTEXT

For heterogeneous range partitioning the numeric data types support negative and positive numbers, so that the use case shown in the following example (where the column TESTRESULT is defined as an integer) is possible:

```
PARTITION BY RANGE("TESTRESULT") (
(PARTITION VALUES = -11),
(PARTITION -10 <= VALUES < 10),
(PARTITION OTHERS)
);
```

This option is supported for the following multi-level range partitioning types: RANGE-RANGE, RANGE-HASH and also HASH-RANGE. It is not supported for multi-store (dynamic tiering) tables.

For the VARBINARY type the partition definition entered in the SQL code must specify the boundaries of the conditions using the HEX representation of the binary value. This is shown in the following example which creates a 2-column table, the first column is for integers and the second column for 16 byte varbinary data. In this case, partition boundaries for the varbinary data must be specified with 32 hexadecimal digits (2 hex characters required for each binary byte):

```
create column table DEMO_RAW (a int, b varbinary(16))
partition by range (b)
(
partition '00000000000000000000000000000000' <= values < '01000000000000000000000000000000',
partition '01000000000000000000000000000000' <= values < '02000000000000000000000000000000',
partition '02000000000000000000000000000000' <= values < '03000000000000000000000000000000',
partition '03000000000000000000000000000000' <= values < 'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
)
```

When using a balanced partitioning schema and defining ranges for partitioned tables on TIMESTAMP columns, the precision of TIMESTAMP values can be specified in units up to the level of hour (does not apply to heterogeneous partitioning). The hour value can be either appended to a date separated by a space or included in a single string of digits as shown in the following examples:

```
PARTITION '2010-01-01 00' <= VALUES < '2010-02-01 00', PARTITION OTHERS
```

```
PARTITION 2016031400 <= VALUES < 2016031500, PARTITION OTHERS
```

Related Information

Partitioning Limits [page 367]
CREATE TABLE Statement (Data Definition)

8.11.15.2 Multilevel Partitioning

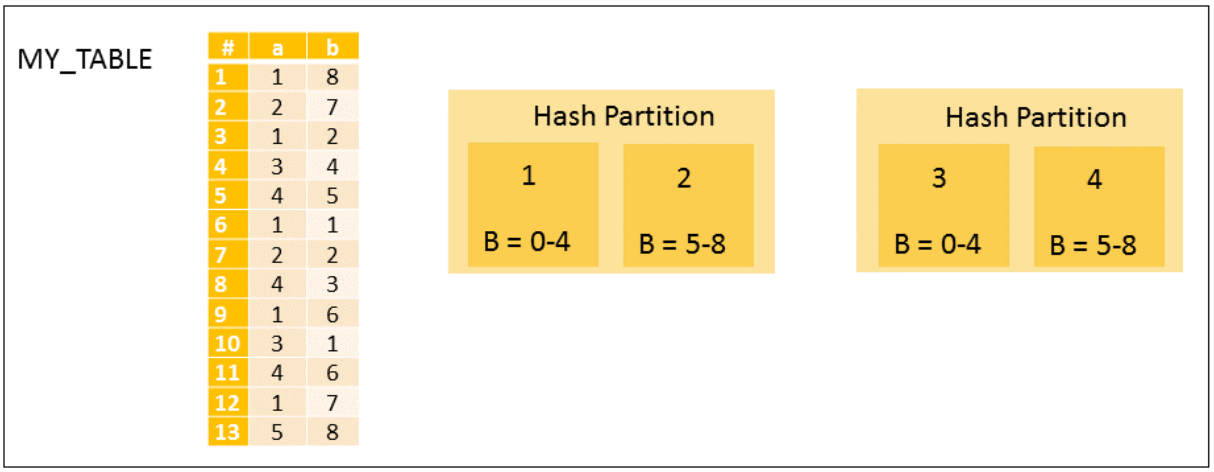
Multilevel partitioning can be used to overcome the limitation of single-level hash partitioning and range partitioning, that is, if the table has a primary key this column must be used as a partitioning column. Multilevel partitioning makes it possible to partition by a column that is not part of the primary key.

For balanced partitioning the following multilevel partitioning types are available:

- HASH-HASH
- HASH-RANGE
- RANGE-RANGE
- ROUNDROBIN-RANGE

Different multilevel partitioning types are available using *heterogeneous* partitioning refer to the comparison table in the topic 'Heterogeneous Partitioning' to see which partitioning type is required. Additionally, using heterogeneous range partitioning, three partition levels are supported as described in the topic 'Partitioning Over Three Levels'.

The following code example and illustration show how multilevel partitioning can be applied using hash partitioning at the first level and range partitioning at the second level. Data in the second level partitions is grouped on the basis of the value of a selected column 'b': rows where the value is below 5 and rows where the value is 5 or greater but less than 9.



Multilevel Partitioning

The syntax of the SQL code to create these partitions is as follows:

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, PRIMARY KEY (a,b))
PARTITION BY
HASH (a,b) PARTITIONS 2,
RANGE (b) (PARTITION 0 <= VALUES < 5, PARTITION 5 <= VALUES < 9)
```

The primary key column restriction only applies at the first partitioning level. When a row is inserted or updated, the unique constraint of the key must be checked. If the primary key has to be checked on all partitions across the landscape, this would involve expensive remote calls. Second-level partition groups, however, allow inserts to occur whilst only requiring primary key checks on local partitions.

Related second level partitions form a partition group; the figure above shows two groups (partitions 1 and 2 are a group and partitions 3 and 4). When a row is inserted into partition 1, it is only required to check for uniqueness on partitions 1 and 2. All partitions of a partition group must reside on the same host. SQL commands are available to move partitions but it is not possible to move individual partitions of a group, only partition groups as a whole.

Using Date Functions to Partition

You can use multilevel partitioning to implement time-based partitioning to leverage a date column and build partitions according to month or year. This could be used for example to minimize the run-time of the delta merge operation. The performance of the delta merge operation depends on the size of the main index of a table. If data is inserted into a table over time and it also contains a date in its structure, then multilevel partitioning on the date could be very effective. Partitions containing old data are typically only modified infrequently, there is therefore no need for a delta merge on these partitions; the merge is only required on partitions where new data is inserted.

If a table needs to be partitioned by month or by year and it contains only a date column or a timestamp column, you can use the date functions shown below to restrict your query results by year or by year and month.

Example: Partitioning Using Date Functions

This example partitions by hash using the year() function:

```
CREATE COLUMN TABLE MY_TABLE (a DATE, b INT, PRIMARY KEY (a,b)) PARTITION BY HASH
(year(a)) PARTITIONS 4
```

If a value takes the format "2018-12-08", the hash function is only applied to "2018". This function can also be used for pruning.

This example partitions by range using the year() function:

```
CREATE COLUMN TABLE MY_TABLE (a DATE, b INT, PRIMARY KEY (a,b)) PARTITION BY
RANGE (year(a)) (PARTITION '2010' <= values < '2013', PARTITION '2013' <= values
< '2016')
```

This example partitions by range using the year and month value using the month() function:

```
CREATE COLUMN TABLE MY_TABLE (a DATE, b INT, PRIMARY KEY (a,b)) PARTITION BY
RANGE (month(a)) (PARTITION '2005-01' <= values < '2005-07', PARTITION '2005-07' <=
values < '2006-01')
```

Related Information

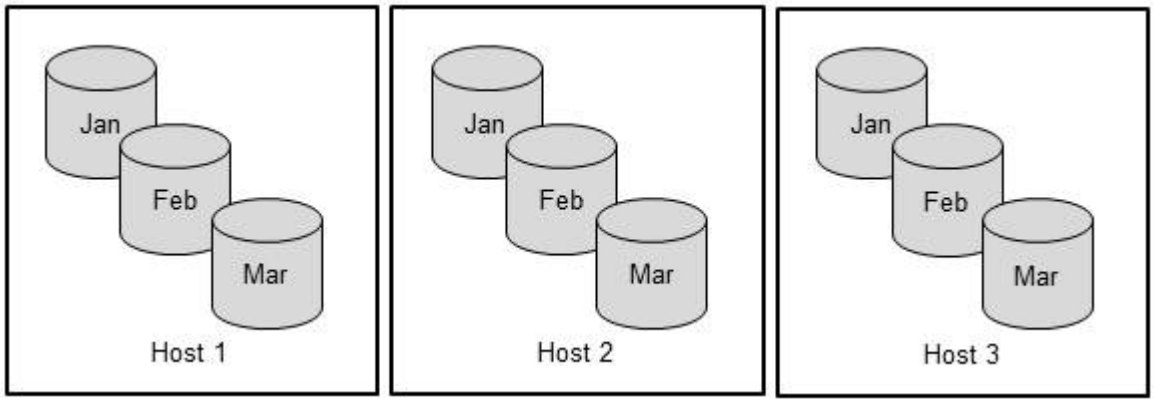
Partitioning Over Three Levels [page 332]
Heterogeneous Partitioning [page 329]

8.11.15.2.1 Hash-Range Partitioning

Hash-range partitioning is the most common type of multi-level partitioning. Hash partitioning is implemented at the first level for load balancing and range partitioning at the second level for time-based partitioning.

The following figure shows a typical usage scenario. The load is distributed to three hosts using hash partitioning. Range partitioning is used at the second level to distribute the data to individual partitions according to month.

Hash-Range Partitioning



For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Table with Hash-Range Multi-Level Partitioning Using SQL

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT, PRIMARY KEY (a,b))
    PARTITION BY
        HASH (a, b) PARTITIONS 4,
        RANGE (c)
            (PARTITION 1 <= VALUES < 5,
```

```
             PARTITION 5 <= VALUES < 20)
```

Related Information

CREATE TABLE Statement (Data Definition)

8.11.15.2.2 RoundRobin-Range Partitioning

RoundRobin-range multi-level partitioning is the same as hash-range multi-level partitioning but with RoundRobin partitioning at the first level.

For RoundRobin and RoundRobin-Range partitioning schemas there is validation of the following DML statements to ensure that the data remains consistent: INSERT, UPDATE, DELETE and UPSERT. For these statements the partition id passed is checked to ensure that the values are consistent with the original roundrobin partitioning specification.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Table with RoundRobin-Range Partitioning Using SQL

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT)
    PARTITION BY
        ROUNDROBIN PARTITIONS 4,
        RANGE (c)
            (PARTITION 1 <= VALUES < 5,
             PARTITION 5 <= VALUES < 20)
```

Related Information

CREATE TABLE Statement (Data Definition)

8.11.15.2.3 Hash-Hash Partitioning

Hash-hash multi-level partitioning is implemented with hash partitioning at both levels. The advantage of this is that the hash partitioning at the second level may be defined on a non-key column.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Table with Hash-Hash Partitioning Using SQL

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT, PRIMARY KEY (a,b))
    PARTITION BY
        HASH (a, b) PARTITIONS 4,
        HASH (c) PARTITIONS 7
```

Related Information

CREATE TABLE Statement (Data Definition)

8.11.15.2.4 Range-Range Partitioning

Range-range multi-level partitioning is implemented with range partitioning at both levels. The advantage of this is that the range partitioning at the second level may be defined on a non-key column.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

Example: Creating a Table with Range-Range Partitioning Using SQL

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT, PRIMARY KEY (a,b))
    PARTITION BY
        RANGE (a)
            (PARTITION 1 <= VALUES < 5,
             PARTITION 5 <= VALUES < 20),
        RANGE (c)
            (PARTITION 1 <= VALUES < 5,
             PARTITION 5 <= VALUES < 20)
```

Related Information

CREATE TABLE Statement (Data Definition)

8.11.15.3 Heterogeneous Partitioning

Heterogeneous partitioning offers more flexibility for multilevel partitioning schemas.

The normal *balanced* (*homogeneous*) partitioning schema applies the same second-level specification to all first-level partitions. But for some scenarios a flexible *heterogeneous* second-level range partitioning schema

may be required with different second-level or third-level range specifications. This is possible for both column-store tables and Dynamic Tiering multistore tables using the heterogeneous partitioning clauses for the CREATE TABLE and ALTER TABLE PARTITION BY statements.

The partitioning schema types supported depend on which set of partitioning clauses you use. The following table compares homogenous and heterogenous partitioning schemas:

Single / Multilevel Schema	HOMOGENEOUS	HETEROGENEOUS
RANGE	Supported	Supported
HASH	Supported	
ROUNDROBIN	Supported	
RANGE-RANGE	Supported	Supported
RANGE-HASH		Supported
HASH-RANGE	Supported	Supported
HASH-HASH	Supported	
ROUNDROBIN-RANGE	Supported	
RANGE-RANGE-RANGE		Supported
RANGE-RANGE-HASH		Supported

For range-range and range-hash partitioning all partitions at the second level must be based on the same columns.

Heterogeneous partitioning supports the following features:

- Properties (metadata) for partitions; these are defined simply as a value with a boolean state (see section *Partition Properties*)
- Placement of first-level and second-level partitions so that partitions can be placed (or moved) directly on a specified host supporting HANA scale-out or Dynamic Tiering solution (placement is also supported for normal balanced partitioning).

Primary Keys

Depending on your requirements, you can apply a condition for heterogeneous partitioning to enforce partitioning on a primary key column. When you create a table with partitions or when you repartition a table using the ALTER TABLE ... PARTITION BY syntax, a <PRIMARY KEY CHECK CLAUSE> is available which you can use for this purpose. If the check is applied then the partitioning columns must be part of the key. If the check is not applied then any column may be used for partitioning. See also the topic *Partitioning Limits* for further information on partitioning columns.

An additional option which is available with the create table statement is the PRIMARY KEY UPDATE property, this determines if UPDATE statements are allowed on primary key columns. SQL statements are also available to drop heterogeneous subpartitions, redefine heterogeneous first-level partitions, and move heterogeneous first-level partitions (and their subpartitions) to a new location. Full details of these options are given in the *SAP HANA SQL and System Views Reference*.

Examples

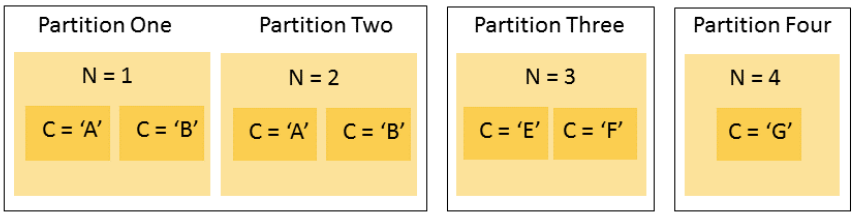
The following set of range-range examples illustrate this feature.

Example 1: Heterogeneous 2-level range-range partitions. The SUBPARTITION BY keyword is required to specify the second-level heterogeneous partitions. The syntax in this example shows partitions one and two bracketed together; this is possible if the same subpartition definition applies to both first level partitions.

```
CREATE COLUMN TABLE T(N INT, C CHAR) PARTITION BY RANGE(N) (
  (PARTITION VALUE = 1, PARTITION VALUE = 2) SUBPARTITION BY RANGE(C) (PARTITION VALUE = 'A' , PARTITION VALUE = 'B'),
  (PARTITION VALUE = 3)                      SUBPARTITION BY RANGE(C) (PARTITION VALUE = 'E', PARTITION VALUE = 'F'),
  (PARTITION VALUE = 4)                      SUBPARTITION BY RANGE(C) (PARTITION VALUE = 'G')
  )
```

Table 'T'

#	N	C
1	1	A
2	2	A
3	1	B
4	3	E
5	4	G
6	1	A
7	2	B
8	4	G
9	1	B
10	3	F
11	4	A
12	1	C
13	5	A

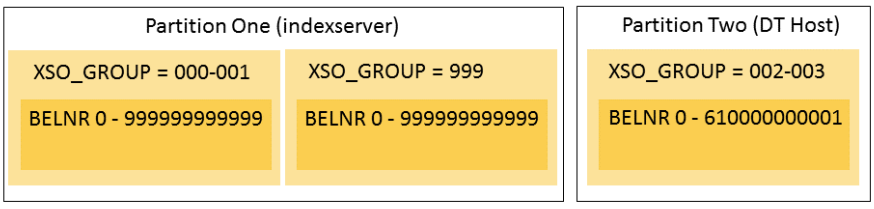


Example 2: AT LOCATION can be used at first or second level to specify where the partition(s) should be created, this can be either on an indexserver or the extended store server for Dynamic Tiering. The INSERT switch is available to create read-only partitions and can be used to prevent insert statements from being executed on the partition.

```
CREATE COLUMN TABLE T3(XSO_GROUP INT, BELNR BIGINT) PARTITION BY RANGE(XSO_GROUP) (
    (PARTITION 000 <= VALUES < 002 AT LOCATION 'indexserver_host:port',
     PARTITION VALUE = 999 AT LOCATION 'indexserver_host:port')
            SUBPARTITION BY RANGE(BELNR) (PARTITION 000000000000 <= VALUES < 999999999999 INSERT OFF ),
     (PARTITION 002 <= VALUES < 004 AT LOCATION 'DT_host:port')
            SUBPARTITION BY RANGE(BELNR) (PARTITION 000000000000 <= VALUES < 610000000001 INSERT OFF));
```

Table 'T3'

#	XSO_GROUP	BELNR
1	001	610000000001
2	000	999999999999
3	002	610000000001
4	999	610000000001
5	001	610000000001
6	999	610000000001
7	002	610000000001
8	001	610000000001
9	999	610000000001

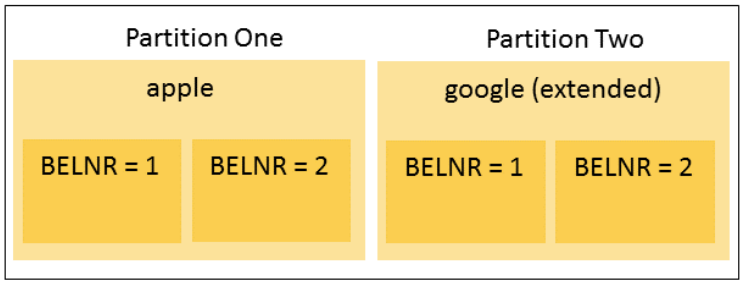


Example 3: This example shows the storage type option which is available: either **`USING DEFAULT STORAGE`** or **`USING EXTENDED STORAGE`** for Dynamic Tiering.

```
CREATE COLUMN TABLE T_Unbalanced ( BELNR INT, COLGRP NVARCHAR(20) ) PARTITION BY RANGE(COLGRP) (
    ( PARTITION VALUE = 'apple'  USING DEFAULT STORAGE,
      PARTITION VALUE = 'google' USING EXTENDED STORAGE )
SUBPARTITION BY RANGE(BELNR) ( PARTITION VALUE=1, PARTITION VALUE=2 ) )
```

Table 'T_Unbalanced'

#	BELNR	COLGRP
1	1	apple
2	2	google
3	2	google
4	2	apple
5	1	google
6	1	apple
7	2	google
8	2	google
9	1	apple
10	1	google



Related Information

Partitioning Limits [page 367]
Partition Properties [page 334]
Partitioning Over Three Levels [page 332]
Links to SAP HANA Database SQL Reference Guide
Heterogeneous Create Partition Clauses

8.11.15.3.1 Partitioning Over Three Levels

SAP HANA supports three-level partitioning using heterogeneous partitioning. This approach allows different branches of the partition hierarchy to use different partitioning types or depths, making it possible to define flexible and data-specific partitioning structures.

The following three-level partitioning types are supported:

- RANGE-RANGE-HASH
- RANGE-RANGE-RANGE

You can find details about third-level partitions in the following system views:

RANGE-RANGE-HASH

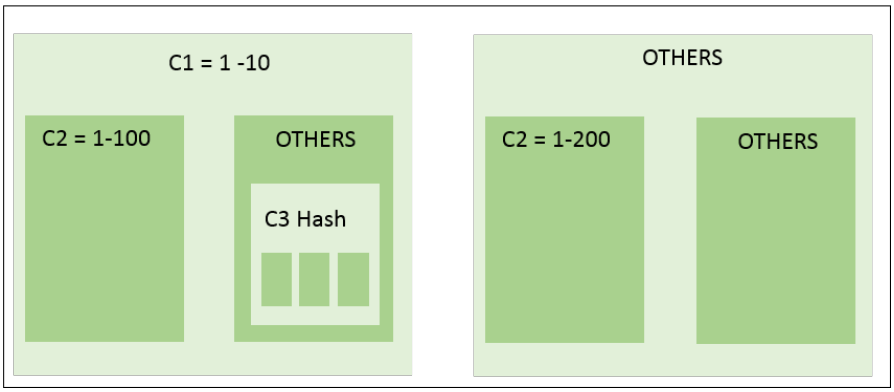
This layout uses two levels of range partitioning to group related values, with a hash at the third level to distribute data more evenly.

The following code example and illustration show how multilevel partitioning can be applied using RANGE-RANGE-HASH partitioning.

Heterogeneous 3-level Partitions: RANGE-RANGE-HASH

Table ‘T’

#	C1	C2	C3
1	1	41	411
2	2	27	272
3	17	170	17017
4	3	35	353
5	4	41	414
6	1	15	151
7	2	29	292
8	9	97	979
9	15	352	1515
10	3	3	33
11	4	246	464
12	11	121	1211
13	5	50	505



The SQL code to create these partitions is as follows:

```
CREATE COLUMN TABLE T (C1 INT, C2 INT, C3 INT) PARTITION BY RANGE (C1) (
            (PARTITION 1 <= VALUES < 10) SUBPARTITION BY RANGE (C2) (
                (PARTITION 1 <= VALUES < 100),
                (PARTITION OTHERS) SUBPARTITION BY HASH (C3) PARTITIONS 3),
            (PARTITION OTHERS) SUBPARTITION BY RANGE (C2) (
                (PARTITION 1 <= VALUES < 200),
                (PARTITION OTHERS)));
```

RANGE-RANGE-RANGE

This layout applies range partitioning at each level and is suitable for hierarchically structured data, for example time-based data organized by year, month, and day, or product categories.

The following code example shows how to define a table using range partitioning at all three levels.

```
CREATE COLUMN TABLE T1 (C1 INT, C2 INT, C3 INT) PARTITION BY RANGE (C1) (
            (PARTITION 1 <= VALUES < 10) SUBPARTITION BY RANGE (C2) (
                (PARTITION 1 <= VALUES < 100),
                (PARTITION OTHERS) SUBPARTITION BY RANGE (C3) (
                    PARTITION 1 <= VALUES < 10,
                    PARTITION 10 <= VALUES < 20,
                    PARTITION OTHERS)),
            (PARTITION OTHERS) SUBPARTITION BY RANGE (C2) (
                (PARTITION 1 <= VALUES < 200),
                (PARTITION OTHERS)));
```

Related Information

Links to SAP HANA Database SQL Reference Guide
TABLE_PARTITIONS System View
PARTITIONED_TABLES System View

8.11.15.4 Partition Properties

A set of configurable properties is available for partitions, which can be applied using the ALTER PARTITION command.

The following properties are supported for partitions, values for these are saved in the TABLE_PARTITIONS system view:

Property	Value	Scope
Load Unit	DEFAULT, COLUMN, PAGE	Heterogeneous partitioning only
Persistent memory (NVM)	ON / OFF (see note below)	Balanced and heterogeneous partitioning
NUMA Node	List of preferred NUMA Nodes (see note below)	Balanced and heterogeneous partitioning
Insert	TRUE / FALSE	Heterogeneous partitioning only
Storage type	DEFAULT / EXTENDED (for multistore tables - see note below)	Balanced and heterogeneous partitioning
GROUP_NAME, GROUP_TYPE, SUBTYPE	Group properties from table placement	Heterogeneous partitioning only

These properties can be modified for a partition using the ALTER PARTITION key words or using PARTITION BY. Using the PARTITION BY syntax multiple properties can be applied simultaneously, see the third example below. Refer to the sections of the *SAP HANA SQL Reference Guide* for details of the heterogeneous and non-heterogeneous (balanced) alter partition clauses

> **Note**
>
> Partition properties are lost if a table is repartitioned using ALTER TABLE ... PARTITION BY ... and the partition specification is switched from Balanced to Heterogeneous (or vice-versa). This is true even if the same partition layout is used. In this case partition properties must be redefined using the ALTER TABLE statement.

```
ALTER TABLE T1 ALTER PARTITION 1 INSERT ON;
```

```
ALTER TABLE T1 ALTER PARTITION 2 INSERT OFF;
```

```
ALTER TABLE T1 PARTITION BY RANGE(x) (PARTITION VALUE =1 INSERT ON, PARTITION
VALUE =2 INSERT OFF);
```

The following example applies properties in a multilevel partitioning scenario:

```
ALTER TABLE T1 PARTITION BY RANGE(c1) (
       (PARTITION 0 <= VALUES < 5 GROUP NAME 'test5')
                    SUBPARTITION BY RANGE(c2) (PARTITION VALUE = 'a' GROUP NAME
'test6', PARTITION VALUE = 'b', PARTITION VALUE = 'c' ),
       (PARTITION  5 <= VALUES < 10 NUMA NODE ('5'))
                    SUBPARTITION BY RANGE(c2) (PARTITION VALUE = 'a' NUMA NODE
('6'), PARTITION VALUE = 'b', PARTITION VALUE = 'c'),
       (PARTITION 10 <= VALUES < 15 INSERT OFF)
```

```
                    SUBPARTITION BY RANGE(c2) (PARTITION VALUE = 'a' INSERT ON,
PARTITION VALUE = 'b', PARTITION VALUE = 'c'));
```

> **Note**
>
> Explicit partition properties can only be set for RANGE partitions. For HASH and ROUNDROBIN partitions, these properties can be inherited from their parent or ancestor partitions, but they cannot be explicitly set.

Storage Type Property

The storage type property only applies at the first level location for balanced partitioning and can only be modified by using the MOVE PARTITION command:

```
MOVE PARTITION <partition_number> TO <move_location>
```

Persistent Memory Property

Persistent memory can be enabled or disabled at the level of partitions using the CREATE and ALTER table commands with the alter_persistent_memory_spec clause. Refer to the section Persistent Memory for more details.

Persistent memory is consumed only when the partitioning property is set to TRUE. Consumption is shown as PERSISTENT_MEMORY_SIZE_IN_TOTAL in the monitoring view M_TABLE_PARTITIONS, which shows current pure persistent memory consumption (in bytes) for every column store table. The columns MEMORY_SIZE_IN_MAIN and MEMORY_SIZE_IN_TOTAL capture only the paged memory consumption and not the persistent memory consumption. See also 'Persistent Size of Page Loadable Data' in the Native Storage Extension section.

NUMA Node property

NUMA node preferences can be applied to range partitions using the `numa_node_preference_clause`. Values for this property can be applied to hash partitions using ALTER TABLE ALTER PARTITION syntax.

Related Information

Persistent Memory [page 186]
Heterogeneous Alter Partition Clauses
Non-heterogeneous Alter Partition Clauses
Persistent Size of Page Loadable Data [page 217]

8.11.15.5 Setting the Load Unit

The load unit determines the loading behavior for column-store tables, partitions or columns. When setting the load unit to `PAGE LOADABLE`, the data is managed by SAP HANA native storage extension (NSE).

There are several options that support the `PAGE LOADABLE` load unit to be compatible with NSE. The `PAGE LOADABLE` attribute is only supported for balanced `HASH-RANGE` and heterogeneous partitions.

Convert Partitions of Multistore Tables

You can convert partitions of multistore tables in extended storage, which have been created for Dynamic Tiering and partitioned by `HASH-RANGE`, to default-store partitions for use with the native storage extension. The `ALTER PARTITION` statement supports a value to specify the partition load unit `<partition_loadunit_spec>`. This can be set to `COLUMN LOADABLE`, `PAGE LOADABLE` (for NSE) or `DEFAULT LOADABLE`. The load unit for `DEFAULT LOADABLE` is column loadable.

The following example shows how multi-level `HASH-RANGE` partitioning is applied to a table and specific storage types are assigned. Three `RANGE` partitions based on column Y are in extended storage:

```
CREATE COLUMN TABLE TAB (X INT NOT NULL, Y INT NOT NULL)
   PARTITION BY HASH(X) PARTITIONS 3, RANGE(Y) (
   USING DEFAULT STORAGE (PARTITION 0 <= VALUES < 10)
   USING EXTENDED STORAGE (PARTITION 10 <= VALUES < 20,
       PARTITION 20 <= VALUES < 30, PARTITION 30 <= VALUES < 50));
```

Two of these are then moved to column store with the `PAGE LOADABLE` load unit option:

```
ALTER TABLE TAB ALTER PARTITION (Y) USING DEFAULT STORAGE
                ( 10 <= values < 20, 30 <= VALUES < 50) PAGE LOADABLE
```

Note that the syntaxt above can only be used for `HASH-RANGE` multistore tables.

Set the Load Unit When Creating a Table

You can specify the load unit when creating a table that has a heterogeneous partitioning schema. The load unit of all target `RANGE` partitions can be set individually to either `COLUMN LOADABLE` or `PAGE LOADABLE`.

- If the load unit of a partition is not specified, it will inherit the load unit from the table level. You can specify the default table-level load unit at the end of the DDL.
- If no table level unit value is specified, the load unit is set to `DEFAULT LOADABLE`. The load unit for `DEFAULT LOADABLE` is column loadable.

In the following example, the partition [1, 50) is set to `COLUMN LOADABLE`, the partition [50, 100) inherits the default load unit, that is, the table-level load unit `PAGE LOADABLE`, and the partition [100, 150) is set to `PAGE LOADABLE`:

```
CREATE TABLE T1 (C1 INT) PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 50 COLUMN LOADABLE, PARTITION 50 <= VALUES < 100,
PARTITION 100 <= VALUES < 150 PAGE LOADABLE,
PARTITION 150 <= VALUES < 200 COLUMN LOADABLE, PARTITION OTHERS COLUMN
LOADABLE)) PAGE LOADABLE;
```

Set the Load Unit When Repartitioning a Table

You can specify the load unit when repartitioning a table that has a heterogeneous partitioning schema with (`ALTER TABLE ... PARTITION BY`). The load unit property of all target `RANGE` partitions can be set individually to either `COLUMN LOADABLE` or `PAGE LOADABLE`:

```
CREATE TABLE T1 (C1 INT) PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 40, PARTITION 40 <= VALUES < 70, PARTITION 70 <=
VALUES < 90,
PARTITION 90 <= VALUES < 160, PARTITION 160 <= VALUES < 190,  PARTITION OTHERS
COLUMN LOADABLE));
ALTER TABLE T1 PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 30 PAGE LOADABLE, PARTITION 30 <= VALUES < 70 COLUMN
LOADABLE, PARTITION 70 <= VALUES < 90 COLUMN LOADABLE,
```

```
PARTITION 90 <= VALUES < 160 PAGE LOADABLE, PARTITION 160 <= VALUES < 190 COLUMN
LOADABLE, PARTITION OTHERS PAGE LOADABLE));
```

When repartitioning a table, new partitions are created with whatever load unit value is assigned in the `ALTER TABLE` statement. However, if you have not specified a load unit for a partition, and you attempt to create a new partition from sources which have a combination of different load unit values (`COLUMN LOADABLE` and `PAGE LOADABLE`), this conflict generates an error.

The following example shows source partitions with different property values: the load unit is not explicitly provided for the new partition [30, 70), and there is a conflict between `COLUMN LOADABLE` and `PAGE LOADABLE` in the source partitions. Thus, the new partition cannot be successfully created and generates an error:

```
CREATE TABLE T1 (C1 INT) PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 50 COLUMN LOADABLE, PARTITION 50 <= VALUES < 100 PAGE
LOADABLE, PARTITION 100 <= VALUES < 150 PAGE LOADABLE,
PARTITION 150 <= VALUES < 200 COLUMN LOADABLE, PARTITION OTHERS COLUMN
LOADABLE)) PAGE LOADABLE;
ALTER TABLE T1 PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 30 PAGE LOADABLE, PARTITION 30 <= VALUES < 70,
PARTITION 70 <= VALUES < 90 COLUMN LOADABLE,
PARTITION 90 <= VALUES < 160 PAGE LOADABLE, PARTITION 160 <= VALUES < 190 COLUMN
LOADABLE, PARTITION OTHERS PAGE LOADABLE));
```

In the following repartitioning statement for a table, the load unit property is not explicitly provided for the new partition [30, 70), but because there is no conflict between `COLUMN LOADABLE` and `PAGE LOADABLE` in the source partitions the new partition can be successfully created:

```
CREATE TABLE T1 (C1 INT) PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 50 COLUMN LOADABLE, PARTITION 50 <= VALUES < 100,
PARTITION 100 <= VALUES < 150 PAGE LOADABLE,
PARTITION 150 <= VALUES < 200 COLUMN LOADABLE, PARTITION OTHERS COLUMN
LOADABLE)) PAGE LOADABLE;
ALTER TABLE T1 PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 30 PAGE LOADABLE, PARTITION 30 <= VALUES < 70,
PARTITION 70 <= VALUES < 90 COLUMN LOADABLE,
PARTITION 90 <= VALUES < 160 PAGE LOADABLE, PARTITION 160 <= VALUES < 190 COLUMN
LOADABLE, PARTITION OTHERS PAGE LOADABLE));
```

> **Note**
>
> Changing the load unit can also be done for a table, partition, or column using `ALTER TABLE`. A 'cascade' option is available, which applies the load unit to all lower objects in the object hierarchy.

Set the Load Unit for Partitions With `ALTER PARTITION`

You can specify the load unit for a table that has a heterogeneous partitioning schema with (`ALTER TABLE ... ALTER PARTITION`) in two ways:

1. You can set the load unit for a partition in the partition specification.
   This example shows the change to `PAGE LOADABLE` of the partition [1, 50):

   ```
   CREATE TABLE T1 (C1 INT) PARTITION BY RANGE (C1)
   ((PARTITION 1 <= VALUES < 50 COLUMN LOADABLE, PARTITION 50 <= VALUES < 100,
   PARTITION 100 <= VALUES < 150 PAGE LOADABLE,
   PARTITION 150 <= VALUES < 200 COLUMN LOADABLE, PARTITION OTHERS COLUMN
   LOADABLE)) PAGE LOADABLE;
   ALTER TABLE T1 ALTER PARTITION RANGE (C1) ((PARTITION 1 <= VALUES < 50)) PAGE
   LOADABLE;
   ```

2. In most cases, you can set the load unit for a partition by specifying the partition ID. For exceptions, see the next section.
   This example shows the change to `COLUMN LOADABLE` of the partition 2, [50, 100):

```
CREATE TABLE T1 (C1 INT) PARTITION BY RANGE (C1)
((PARTITION 1 <= VALUES < 50 COLUMN LOADABLE, PARTITION 50 <= VALUES < 100,
PARTITION 100 <= VALUES < 150 PAGE LOADABLE,
PARTITION 150 <= VALUES < 200 COLUMN LOADABLE, PARTITION OTHERS COLUMN
LOADABLE)) PAGE LOADABLE;
ALTER TABLE T1 ALTER PARTITION 2 COLUMN LOADABLE;
```

Change Load Unit for RANGE-HASH and RANGE-RANGE-HASH Partitioned Tables

You can change the load unit for `RANGE-HASH` and `RANGE-RANGE-HASH` partitioned tables by specifying the load unit for the upper layer.

This example shows how to change the load unit for a `RANGE-HASH` partitioned table:

```
CREATE COLUMN TABLE T1 (A INT, B INT NOT NULL, C INT) PARTITION BY RANGE (A)
    ( (PARTITION 1 <= VALUES < 10) SUBPARTITION BY HASH (C, B) PARTITIONS 3 ,
      (PARTITION OTHERS) );
ALTER TABLE T1 ALTER PARTITION RANGE (A) ((PARTITION 1 <= VALUES < 10)) COLUMN
LOADABLE;
```

This example shows how to change the load unit for a `RANGE-RANGE-HASH` partitioned table:

```
CREATE COLUMN TABLE T2 (A INT, B INT NOT NULL, C INT) PARTITION BY RANGE (A)
    ( (PARTITION 1 <= VALUES < 10) SUBPARTITION BY RANGE (B) ( (PARTITION 1 <=
VALUES < 100), (PARTITION OTHERS) SUBPARTITION BY HASH (C) PARTITIONS 3 ),
      (PARTITION OTHERS)           SUBPARTITION BY RANGE (B) ( (PARTITION 1 <=
VALUES < 200), (PARTITION OTHERS DYNAMIC THRESHOLD 1) ) );
ALTER TABLE T2 ALTER PARTITION RANGE (A) ((PARTITION 1 <= VALUES < 10)
SUBPARTITION BY RANGE (B) ((PARTITION OTHERS))) COLUMN LOADABLE;
```

> **Note**
>
> You cannot change the load unit of `HASH` partitions in `RANGE-HASH` and `RANGE-RANGE-HASH` tables by specifying the partition ID.

The following example shows a `RANGE-HASH` table with 4 partitions. Partitions 1 to 3 are `HASH` partitioned, and partition 4 is `RANGE` partitioned. Hence, changing the load unit of partition 1 fails, and changing the load unit of partition 4 succeeds:

```
CREATE COLUMN TABLE T1 (A INT, B INT NOT NULL, C INT) PARTITION BY RANGE (A)
    ( (PARTITION 1 <= VALUES < 10) SUBPARTITION BY HASH (C, B) PARTITIONS 3 ,
      (PARTITION OTHERS) );
ALTER TABLE T1 ALTER PARTITION 1 COLUMN LOADABLE; -- fail;
ALTER TABLE T1 ALTER PARTITION 4 COLUMN LOADABLE; -- succeed;
```

The following example shows a `RANGE-RANGE-HASH` partitioned table where partition 1 is `RANGE` partitioned, and partition 2 is `HASH` partitioned. Hence, changing the load unit of partition 1 succeeds, and changing the load unit of partition 2 fails:

```
CREATE COLUMN TABLE T2 (A INT, B INT NOT NULL, C INT) PARTITION BY RANGE (A)
    ( (PARTITION 1 <= VALUES < 10) SUBPARTITION BY RANGE (B) ( (PARTITION 1 <=
VALUES < 100), (PARTITION OTHERS) SUBPARTITION BY HASH (C) PARTITIONS 3 ),
      (PARTITION OTHERS)           SUBPARTITION BY RANGE (B) ( (PARTITION 1 <=
VALUES < 200), (PARTITION OTHERS DYNAMIC THRESHOLD 1) ) );
ALTER TABLE T2 ALTER PARTITION 1 COLUMN LOADABLE; -- succeed;
ALTER TABLE T2 ALTER PARTITION 2 COLUMN LOADABLE; -- fail;
```

Related Information

SAP HANA Native Storage Extension [page 196]
Changing the Load Unit Using ALTER TABLE [page 205]
CREATE TABLE Statement (Data Definition)
ALTER TABLE Statement (Data Definition)

8.11.15.6 Truncating Partitions

You can use the SQL TRUNCATE statement to delete the content of specific partitions of a table.

The SQL statement TRUNCATE can be used to truncate whole tables or specific partitions of a table. This could be required, for example, if data in an individual partition needs to be refreshed, such as data for a single year, but other partitions preserved.

Partitions can be truncated only in column store tables (not row store); partitions in remote tables (those from a different tenant) and replicated tables (with either an OSTR/ATR replica) cannot be truncated.

To truncate a partition, identify the partition by its logical partition id, this can be found in the PART_ID column in the TABLE_PARTITIONS view. The following example illustrates this:

Example

Create table T1 with partitions and subpartitions:

```
CREATE COLUMN TABLE T1 (A INT,B INT)
     PARTITION BY RANGE (A) ((PARTITION 1 <= VALUES < 5, PARTITION 5 <= VALUES
<    20, PARTITION OTHERS)
     SUBPARTITION BY RANGE (B) (PARTITION 1 <= VALUES < 100, PARTITION 100 <=
VALUES < 200));
```

Look up the logical partition id:

```
select * from TABLE_PARTITIONS where table_name='T1';
```

You can then truncate a selected partition:

```
TRUNCATE TABLE T1 PARTITION (4);
```

In this example partition 4 corresponds to range(A) [5..20) range(B)[100..200).

Related Information

TRUNCATE TABLE Statement (Data Manipulation)

8.11.15.7 Range Partitioning: More Options

Some special features are available for range partitioning: adding additional ranges, deleting ranges, and using dynamic 'others' partitions.

8.11.15.7.1 Explicit Partition Handling for Range Partitioning

For all partitioning specifications involving range, it is possible to have additional ranges added and removed as necessary. This means that partitions are created and dropped as required by the ranges in use. In the case of multi-level partitioning, the desired operation is applied to all relevant nodes.

> ⓘ Note
>
> If a partition is created and an others partition exists, the rows in the others partition that match the newly-added range are moved to the new partition. If the others partition is large, this operation may take a long time. If an others partition does not exist, this operation is fast as only a new partition is added to the catalog.

Range partitioning requires at least one range to be specified regardless of whether or not there is an others partition. When partitions are dropped, the last partition created cannot be dropped even if an others partition exists.

For range-range partitioning you have to specify whether a partition has to be added or dropped on the first or second level by specifying the partitioning column.

> ⚠ Caution
>
> The DROP PARTITION command deletes data. It does not move data to the others partition.

For more information about the SQL syntax for partitioning, see *SAP HANA SQL and System Views Reference*.

> ✤ Example
>
> Changing a Table to Add/Drop Partitions
>
> ```
> ALTER TABLE MY_TABLE ADD PARTITION 100 <= VALUES < 200
> ```
>
> ```
> ALTER TABLE MY_TABLE DROP PARTITION 100 <= VALUES < 200
> ```

> ✤ Example
>
> Changing a Table to Add/Drop an Others Partition
>
> ```
> ALTER TABLE MY_TABLE ADD PARTITION OTHERS
> ```

```
ALTER TABLE MY_TABLE DROP PARTITION OTHERS
```

You can add a range to the first and second level respectively as follows;

Example

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT)
PARTITION BY
    RANGE (a) (PARTITION 1 <= VALUES < 5),
    RANGE (b) (PARTITION 100 <= VALUES < 500)
```

Example

```
ALTER TABLE MY_TABLE ADD PARTITION (a) 5 <= VALUES < 10

ALTER TABLE MY_TABLE ADD PARTITION (b) 500 <= VALUES < 1000
```

Example

```
ALTER TABLE MY_TABLE DROP PARTITION (a) 5 <= VALUES < 10

ALTER TABLE MY_TABLE DROP PARTITION (b) 500 <= VALUES < 1000
```

Related Information

ALTER TABLE Statement (Data Definition)

8.11.15.7.2 Dynamic Range Partitioning

For heterogeneous partitioning schemas Dynamic Range Partitioning is available to support the automatic maintenance of the 'others' partition.

When you create an *OTHERS* partition there is a risk that over time it could overflow and require further maintenance. Using the dynamic range feature the others partition is monitored by a background job and will be automatically split into an additional range partition when it reaches a predefined size threshold. The background job also checks for empty partitions and if a range partition is found to be empty it is automatically merged to neighboring empty partitions (the others partition is never automatically deleted).

There are three possible ways of setting the row count threshold for dynamic range partitions:

- Using the PARTITION OTHERS syntax with the CREATE and ALTER SQL statements; in this case the value is saved in the table meta data. Some example statements are given below, refer to the *SAP HANA SQL Reference Guide* for full details.
- Using the DYNAMIC_RANGE_THRESHOLD field of the TABLE_PLACEMENT table which also triggers dynamic partitioning (see *Table Placement Rules*).
- A third option is to define a threshold value as a system configuration parameter (see *Configuration* below).

You can create partitions with a dynamic others partition by including the DYNAMIC keyword in the command when you create the partition, this can be used with either a THRESHOLD value to define a maximum row

count number or an INTERVAL value which can be used to define a maximum time or other numeric 'distance' value. The partition can be either a single level or a second level RANGE partition and dynamic ranges can be used with both balanced and heterogeneous partitioning scenarios.

The INTERVAL option can be used to either:

- Define a time value using data types such as DATE, TIMESTAMP, SECONDDATE (to support existing tables NVARCHAR is also accepted as a date/time value), in this case an interval type value (YEAR, MONTH, HOUR) is also required - see the second example below.
- Define a numeric interval using data types such as INT, BIGINT, NVARCHAR - see the third and fourth examples below.

SQL Examples

Some basic examples are given here, refer to the *SAP HANA SQL Reference Guide for SAP HANA Platform* for full details of the partitioning clauses. These examples create a table T and partition it with a dynamic others partition:

Example

1) This example uses the keyword DYNAMIC and a <threshold_count> value to specify a maximum size for the partition of 3 million rows. When this threshold is reached a new partition will be created:

```
CREATE COLUMN TABLE T (A NVARCHAR(5) NOT NULL, NUM INTEGER NOT NULL)
PARTITION BY RANGE (A AS INT) (PARTITION OTHERS DYNAMIC THRESHOLD 3000000);
```

2) In the following example table T has a column for time values. It uses the keyword DYNAMIC and values for <interval_value> (3) and <interval_type> (MONTH) so that when this threshold of three months is reached new partitions will be generated for every 3 months starting from '2020-01' up to the partition which holds the latest record:

```
CREATE COLUMN TABLE T (A SECONDDATE NOT NULL)
PARTITION BY RANGE(MONTH(A)) ((PARTITION VALUE = '2020-01', PARTITION OTHERS
DYNAMIC INTERVAL 3 MONTH));
```

3) The following example partitions table T and applies a numeric dynamic interval for the OTHERS partition; in this case you use the INTERVAL keyword without a type value:

```
CREATE COLUMN TABLE T (A INT NOT NULL, B INT)
PARTITION BY RANGE(A) ((PARTITION 0 <= VALUES < 1000, PARTITION OTHERS
DYNAMIC INTERVAL 100));
```

4) The following example uses ALTER PARTITION to set a numeric interval for partition 2:

```
ALTER TABLE T ALTER PARTITION 2 DYNAMIC INTERVAL 100;
```

Further examples of using ALTER TABLE are given here:

Example

The first example redefines the threshold for the others partition (balanced partitioning):

```
ALTER TABLE T PARTITION OTHERS DYNAMIC THRESHOLD 1000000;
```

In a heterogeneous partitioning scenario use the ALTER PARTITION syntax. You can refer to OTHERS or to a specific partition ID value:

```
ALTER TABLE T ALTER PARTITION 2 DYNAMIC THRESHOLD 1000000;
ALTER TABLE T ALTER PARTITION 4 DYNAMIC INTERVAL 2 YEAR;
```

This command turns dynamic range partitioning off on table T:

```
ALTER TABLE T PARTITION OTHERS NO DYNAMIC;
```

To turn dynamic range partitioning off in a heterogeneous partitioning scenario ALTER PARTITION is required:

```
ALTER TABLE T ALTER PARTITION OTHERS NO DYNAMIC;
```

As an alternative to the name OTHERS you can also enter a partition id value. This may be helpful if you want to disable dynamic range partitioning on a subpartition group rather than the whole table.

Data Types and Validation

Not all data types are suitable for dynamic partitioning, the range partitioning column must be a not-nullable column and must be a consistently incrementing numerical sequence such as a timestamp or an integer. The following types are supported:

Datetime Types	DATE, SECONDDATE, TIMESTAMP
Numeric Types	TINYINT, SMALLINT, INTEGER, BIGINT
Character String Types	VARCHAR, NVARCHAR
Text Types	SHORTTEXT

For the INTERVAL property the following column types are supported:

Datetime Types	DATE, SECONDDATE, TIMESTAMP For these types the interval unit (MONTH, YEAR) must be specified as shown in the examples above.
Numeric Types	TINYINT, SMALLINT, INTEGER, BIGINT
Character String Types	NVARCHAR Here, only the digits 0-9 should be used. In this case string data representing up to 19 digits is interpreted as BIGINT, it is not necessary to cast the data as an integer using AS INT.

Negative Numbers

With range partitioning in general and also dynamic range partitioning, the numeric data types support positive and negative numbers. A limitation which applies in this case is that when a new dynamic partition is created only values greater than the lower bound of the OTHERS partition can be added (that is, new partitions

can only be created for values which are greater in the positive direction). The following example partition specifications and comments show this:

Partition Specification	Handling of OTHERS partition
partition -200 <= values < -100	This specification defines a bounded partition which accepts values between -200 to -100. The number -99 is outside the partition range but can be successfully added to the OTHERS partition. A new dynamic partition can be created for numbers greater-than in a positive direction. The number -201 is also outside the partition range. It can be successfully added to the OTHERS partition, however, a new dynamic partition cannot be created for numbers which are lower in a negative direction. In this case, the ALTER TABLE command ADD PARTITION FROM OTHERS will generate an error: 'Minimum value in the OTHERS partition is smaller than the least possible lower bound for a new range'
partition values >= -100, partition values < -500	This specification defines two unbound partitions for negative numbers, but is not supported: • The first partition is for all values greater than -100. • The second partition is for all values less than -500. An OTHERS partition in this case is not possible and the PARTITION BY statement with PARTITION OTHERS DYANMIC would generate an error: 'Feature not supported: Dynamic Range OTHERS partition should contain positive infinite.'

Thresholds and Processing

Tables with a dynamic others partition are monitored by a configurable background job which checks the current size of the partition in comparison to the defined limit and makes adjustments as required. If the OTHERS partition has exceeded the defined limit then all records in the OTHERS partition at the time the job runs are added to the new partition, the corresponding SQL statement is ADD PARTITION FROM OTHERS.

Using the THRESHOLD property empty partitions are also removed during this process. Using the INTERVAL property empty partitions must be removed manually using the DROP EMPTY PARTITIONS statement. See the scenario below for details of actions taken and refer to the sections on `Alter Partition Clauses` in the *SAP HANA Database SQL Reference Guide* for more details.

The background job runs at a predefined interval and searches for a threshold value in the sequence of options given above starting with the table meta data. Similarly an ALTER TABLE instruction for dynamic partitioning where a threshold is not specified in the SQL command will apply whichever threshold value it finds first.

Configuration

In the [partitioning] section of indexserver.ini file the following configuration parameters are available for dynamic partitioning:

- `dynamic_range_default_threshold` - enter the value you require, the default value is 10,000,000 rows. If no other value has been specified then this parameter value is used.
- `dynamic_range_check_time_interval_sec` - the background job runs at a predefined interval defined as a number of seconds. By default this is 900 but this can be changed here if required. You can set a value of zero to deactivate the background job (a value of minus 1 has the same effect).

Scenario to Illustrate Dynamic Range Partitioning with Interval

In the following scenario, table T is partitioned on the basis of years with a first partition defined for 2020. A dynamic OTHERS partition is defined with a time interval of 2 years:

```
CREATE COLUMN TABLE T (A SECONDDATE NOT NULL)  PARTITION BY RANGE(YEAR(A)) ((
  PARTITION VALUE = '2020',
  PARTITION OTHERS DYNAMIC INTERVAL 2 YEAR));
```

The table is updated with data records for January 2023 which are initially added to the table's OTHERS partition. In normal use the table is checked by the background job but this can also be done manually on demand:

```
ALTER TABLE T ADD PARTITION FROM OTHERS;
```

When dynamic repartitioning takes place new partitions are created where necessary on the basis of the defined interval to accommodate the data. The partitions may be empty, but are created to complete the sequence. In this case an empty partition is created '2021 <=2023' and the new data for 2023 is added to a partition '2023 <=2025'. Any existing data from the OTHERS partition is moved to the appropriate partitions and a new OTHERS partition is created.

Dynamic repartitioning also adjusts the partitioning of data when partitions are dropped. However, in the case of dynamic partitioning with an INTERVAL value the background job does not do this and it has to be done manually using the following statement:

```
ALTER TABLE T DROP EMPTY PARTITIONS;
```

In this scenario (assuming these partitions still have no data) the partitions 1 and 2 for ranges '2020' and '2021 <=2023' are merged with range '2023 <=2025'. The empty partitions are then dropped and the remaining partition is then renamed to '2020 <=2025'.

Related Information

SAP HANA SQL Reference Guide for SAP HANA Platform
Table Placement Rules [page 373]
Non-heterogeneous Alter Partition Clauses

8.11.15.7.2.1 Dynamic Aging

Dynamic Aging makes it possible to automatically manage at which point in time older partitions can be moved to the 'warm' data store.

The data in a new OTHERS partition is 'hot' data, that is, stored in memory with the load-unit attribute implicitly set to COLUMN LOADABLE. As an extension of the Dynamic Range Partitioning feature Dynamic Aging makes it possible to automatically manage when older partitions can be moved to the 'warm' data store (Native Storage Extension) with the load-unit attribute for the partition set to PAGE LOADABLE. Warm data is then stored on disk and only loaded to memory when required.

Dynamic Aging can be used with both THRESHOLD mode (defining a maximum row count number in partition OTHERS) and INTERVAL mode (defining a maximum time or other numeric interval between each new partition). For example, for a partitioned table which is managed by dynamic partitioning and containing date/time information, you can specify an age limit (for example six months) so that when data in an old partition reaches this age it is moved to NSE storage. The interval is referred to as a 'distance threshold' value. Dynamic Aging can be applied to partition groups and can be individually enabled on each group of a partitioned table with different settings in each partition group (see example 3 below).

The table and its partitions are monitored by a background job and when the distance of the smallest data record in a partition and the greatest data record in the partition group is greater than the distance threshold then the data in the old partition is automatically moved.

Dynamic Aging supports INTEGER columns and DATE columns using the following data types (the NVARCHAR data type is not supported):

- INTEGER data types: INT / SMALLINT / TINYINT / BIGINT
- DATE data types: DATE / LONGDATE / SECONDDATE / TIMESTAMP.

Syntax

Dynamic Aging can be applied to a table by setting a value for the additional DISTANCE property for the OTHERS partition with either CREATE TABLE or ALTER TABLE. This can be either a THRESHOLD value or INTERVAL value:

```
... PARTITION OTHERS DYNAMIC [THRESHOLD <int> [DISTANCE <int> PAGE LOADABLE]]
... PARTITION OTHERS DYNAMIC INTERVAL <int + type> [DISTANCE <int + type> PAGE
LOADABLE]
```

The feature is designed to change hot COLUMN LOADABLE partitions to warm PAGE LOADABLE partitions and the distance value must therefore always be qualified as PAGE LOADABLE as shown here. Where an interval or distance value relates to a date/time column then an interval-type value is also required - see examples which follow.

If a partition has been explicitly set to COLUMN LOADABLE, or if the column loadable attribute is inherited from its parent, then the partition's load unit will not be changed by Dynamic Aging. By default, the load unit setting of a partition is DEFAULT LOADABLE, which means there is no explicit preference. In practice, this means that you must not explicitly apply the COLUMN LOADABLE load unit property to an OTHERS partition where dynamic aging is active as this would prevent any automatic conversion of data.

The interval value (DYNAMIC_DISTANCE_INTERVAL) for an existing partition can be retrieved from the view SYS.TABLE_PARTITIONS.

Examples

Apply Dynamic Aging

1) In the following example, table T has a column for time values; it has an interval value of three months and a distance value of two. This means new partitions generated by Dynamic Range will have range interval set as 3 months, and the partitions containing data older than 2 months (comparing with the latest data) will be moved to warm storage:

```
CREATE COLUMN TABLE T (A SECONDDATE NOT NULL) PARTITION BY RANGE(MONTH(A)) ((
PARTITION VALUE = '2020-01',
```

```
PARTITION OTHERS DYNAMIC INTERVAL 3 MONTH DISTANCE 2 MONTH PAGE LOADABLE));
```

2) The following partitioning statement applies a dynamic range THRESHOLD and a Dynamic Aging distance value of 200:

```
PARTITION OTHERS DYNAMIC THRESHOLD 3000000 DISTANCE 200 PAGE LOADABLE
```

3) The following example illustrates how different interval and distance values can be applied in multilevel partitioning with subpartition groups. Table T1 is partitioned by RANGE-RANGE with a dynamic range INTERVAL. For group one the Dynamic Aging distance is 2 and the unit is 'YEAR'. For group two the distance is 20 and the unit is 'MONTH'. For group three the distance is 600 and the unit is 'HOUR'.

```
CREATE TABLE T1 (C1 date, C2 seconddate NOT NULL) PARTITION BY RANGE (C1) (
  (PARTITION '2010' <= VALUES < '2015') subpartition BY RANGE (MONTH(c2)) (
    PARTITION '2010-01' <= VALUES < '2013-02',
    PARTITION '2013-02' <= VALUES < '2017-06',
    PARTITION OTHERS DYNAMIC INTERVAL 10 MONTH DISTANCE 2 YEAR PAGE LOADABLE),
  (PARTITION '2015' <= VALUES < '2020') subpartition BY RANGE (MONTH(c2)) (
    PARTITION '2010-01' <= VALUES < '2010-02',
    PARTITION '2010-02' <= VALUES < '2010-06',
    PARTITION OTHERS DYNAMIC INTERVAL 20 MONTH DISTANCE 20 MONTH PAGE LOADABLE),
  (PARTITION OTHERS) subpartition BY RANGE (MONTH(c2)) (
    PARTITION '2010-01' <= VALUES < '2010-02',
    PARTITION '2010-02' <= VALUES < '2010-06',
    PARTITION OTHERS DYNAMIC INTERVAL 30 MONTH DISTANCE 600 HOUR PAGE
LOADABLE))
```

Modify Existing Tables

1) You can use the ALTER TABLE statement to apply Dynamic Aging to an existing table with a dynamic range. The table T1 has been created and partitioned with a dynamic OTHERS partition (if no value is specified, as in this example, a DYNAMIC THRESHOLD of 10,000,000 is applied):

```
CREATE TABLE T1 (c1 int NOT NULL, c2 int) PARTITION BY RANGE (c1) ((
    PARTITION 1 <= VALUES < 100,
    PARTITION 100 <= VALUES < 200,
    PARTITION 200 <= VALUES < 300,
    PARTITION OTHERS DYNAMIC
));
```

You can then use ALTER TABLE to modify the OTHERS partition by adding a distance value:

```
ALTER TABLE T1 ALTER PARTITION OTHERS DYNAMIC DISTANCE 200 PAGE LOADABLE;
```

2) Similarly, You can use the ALTER TABLE statement to modify an existing Dynamic Aging distance value. Note that you can also identify the OTHERS partition by its id number, in this example it is partition four:

```
ALTER TABLE T1 ALTER PARTITION 4 DYNAMIC DISTANCE 500 PAGE LOADABLE;
```

Disable Dynamic Aging

The following examples show how you can disable Dynamic Aging.

Disable Dynamic Aging on all of the partition OTHERS in a table.

```
ALTER TABLE T1 ALTER PARTITION OTHERS NO DYNAMIC DISTANCE
```

Disable Dynamic Aging on partition OTHERS with partition id 2.

```
ALTER TABLE T1 ALTER PARTITION 2 NO DYNAMIC DISTANCE
```

Disable Dynamic Aging on all of the partition OTHERS with partition id 1 and 2.

```
ALTER TABLE T1 ALTER PARTITION (1,2) NO DYNAMIC DISTANCE
```

Manual Refresh

The background job to apply Dynamic Aging is timed to run automatically every 20 minutes but an update can be triggered manually for a single table with a REFRESH instruction:

```
ALTER TABLE T1 REFRESH DYNAMIC DISTANCE
```

Related Information

Reduce the Memory Footprint Using Page-Loadable Columns [page 201]

8.11.15.8 Partitioning Operations

How a table is partitioned can be determined on creation or at a later point in time. You can change how a table is partitioned in several ways.

You can change partitioning in the following ways:

- Partition a non-partitioned table
- Change a partitioned table into a non-partitioned table by merging all of its partitions
- Re-partition an already-partitioned table:
  - Change the partitioning specification, for example, from hash to round-robin
  - Change the partitioning columns
  - Increase or decrease the number of partitions.

> ⓘ Note
>
> When you change the partitioning of tables the table creation time of the affected tables will be updated to the time you performed the action.

> ⓘ Note
>
> When you are modifying the properties of a partitioned table using the ALTER TABLE command you can specify a list of partition ID values which will then all be modified with a single statement. For example, this command sets the page loadable attribute for partitions 17 and 18:
>
> ```
> ALTER TABLE T1 ALTER PARTITION (17 18) PAGE LOADABLE;
> ```
>
> This option is available for use with the INSERT ON/OFF, COLUMN/PAGE LOADABLE and NUMA NODE properties. The modifications are only applied if the statement completes without error. If an error occurs

then no partitions are modified. For full syntax details refer to Heterogeneous Alter Partition Clauses in the *SAP HANA SQL Reference Guide* .

Optimization

Performing a partitioning operation on a table can be costly for the following reasons:

- It takes a long time to run, up to several hours for huge tables.
- It has relatively high memory consumption.
- It writes everything to the log (required for backup and recovery).

The re-partitioning operation is a non-blocking process. Apart from a short period at the start as the partitioning process is prepared it does not require an exclusive lock on the database. This means that you can execute SQL DML commands while repartitioning is running, however, DDL operations (table manipulations such as create or drop) are still blocked. At the end of the partitioning process an exclusive lock is again required to apply repartitioning to delta contents.

→ Recommendation

(Re-)partition tables before inserting mass data or while they are still small. If a table is not partitioned and its size reaches configurable absolute thresholds, or if a table grows by a certain percentage per day, the system issues an alert.

Although it is possible to (re-)partition tables and merge partitions manually, in some situations it may be more effective to use the **Table Redistribution** operation available for optimizing table partitioning (for example, if a change of partition specification is **not** required). Redistribution operations use complex algorithms to evaluate the current distribution and determine a better distribution depending on the situation.

Creating Partitions at Specific Locations

When you partition a table you can specify at which location (index server) the partition should be created using the AT LOCATION clause as shown in the following examples. Note that this can be used with all partitioning types: hash, round robin and range.

In this first example, the specific number of partitions to create is given (3) with a corresponding list in the location clause of three hosts (name and port number) so that one partition will be created on each host:

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT) PARTITION BY HASH (a, b)
PARTITIONS 3 AT LOCATION 'myHost01:30001', 'myHost02:30003', 'myHost03:30003';
```

If the number of partitions doesn't match with the number of hosts a best fit is applied automatically: extra locations are ignored if too many are named and locations are reused in round robin fashion if there are more partitions than locations. If no location is specified then the partitions are automatically assigned to hosts at random.

In this second example the number of partitions is returned from the GET_NUM_SERVERS() function:

```
CREATE COLUMN TABLE MY_TABLE (a INT, b INT, c INT) PARTITION BY HASH (a, b)
```

```
PARTITIONS GET_NUM_SERVERS() AT LOCATION 'myHost01:30001';
```

Similarly, in this case, if the number of partitions doesn't match with the number of hosts a best fit is applied automatically. Note that the number of servers returned by GET_NUM_SERVERS() is based on table placement rules and ignores the number of servers specified by the AT LOCATION clause.

Specifying Locations When Repartitioning and Merging Partitions

You can specify locations when repartitioning a table using ALTER TABLE and PARTITION BY with the AT LOCATION clause. If partition locations were defined when the table was originally partitioned it is possible to keep those locations or to overwrite them (see examples in following subsections). This option is supported for range partitioning but cannot be used with large objects (BLOB, NCLOB). The feature is not supported for ONLINE repartitioning.

This section gives examples and commentary on the options available.

Example: a table is initially partitioned with a partition located on host1:

```
CREATE TABLE T(A INTEGER) PARTITION BY RANGE(A) ((PARTITION 1 <= VALUES < 200 AT
LOCATION 'host1:30003'));
```

After working with the table it is then repartitioned with two partitions on specific hosts:

```
ALTER TABLE T PARTITION BY RANGEA(A)
((PARTITION 1 <= VALUES < 100 AT LOCATION 'host2:30003', PARTITION 100 <= VALUES
< 200 AT LOCATION 'host1:30003'));
```

Locations can also be assigned when merging partitions: the following example merges the two partitions of table T at a new location:

```
ALTER TABLE T MERGE PARTITIONS AT LOCATION 'host1:30004';
```

If no location is specified then the new target is created at the location defined by the first source partition - in this example host2:30003.

When repartitioning, the location of the source partition is preserved where possible, but if this is not possible, or if there is any ambiguity about the location then the following rules are applied (see also following example):

- The location of the first source partition is used (ordered by logical partition id)
- For a new partition (where there is no source partition), the first entry of possible table locations for the table is used

In the following example two partitions are created on specific hosts:

```
CREATE TABLE T (A INTEGER) PARTITION BY RANGEA(A)
((PARTITION 1 <= VALUES < 100 AT LOCATION 'host2:30003', PARTITION 100 <= VALUES
< 200 AT LOCATION 'host1:30003'));
```

The two partitions are then combined without specifying a location. The partition is therefore placed on the first available location: host2:

```
ALTER TABLE PARTITION BY RANGE(A) ((PARTITION 1 <= VALUES < 200));
```

Specifying New Locations

In the following scenario a table has been created with a single partition but no locations have been specified:

```
CREATE TABLE T1 (C1 INT, C2 INT) PARTITION BY RANGE (C1) (
(PARTITION 1 <= VALUES < 100));
```

If you repartition this table you can specify a location for individual subpartitions by using AT LOCATION with each partition definition as shown here:

```
ALTER TABLE T1 PARTITION BY RANGE (C1) (
(PARTITION 1 <= VALUES < 50,
PARTITION 50 <= VALUES < 100 AT LOCATION 'host1:30001'));
```

In this case, the second subpartition is created on the specified location - host 1.

You can also specify the same location for all subpartitions by defining the location at the parent level:

```
ALTER TABLE T1 PARTITION BY RANGE (C1) (
(PARTITION 1 <= VALUES < 50 AT LOCATION 'host3:30003')
SUBPARTITION BY RANGE(C2) (PARTITION 10 <= VALUES < 50,
                           PARTITION 50 <= VALUES < 100,
                           PARTITION 300 <= VALUES < 350));
```

In this case all three subpartitions are created at the specified location - host 3.

Geo-spatial Data Types

Repartitioning to locations where the data type of the partitioning column is ST_POINT or ST_GEOMETRY is also supported.

```
CREATE TABLE T(A INT, B st_geometry(NULL)) PARTITION BY HASH(A) PARTITIONS 2 AT
LOCATION 'host1:30103';
 -- ... insert some data etc.
ALTER TABLE T PARTITION BY RANGE(A) (
( PARTITION 0 <= VALUES < 3 AT LOCATION 'host1:30103'),
( PARTITION 3 <= VALUES < 5 AT LOCATION 'host2:30140')
 );
```

In this example, after repartitioning the complete first target partition stays on host1, the second target partition is now located on host2.

Keeping or Overwriting Existing Locations

If the table already has partitions at specific locations, then it is possible to keep those partitions, move the partitions to new locations, or create new partitions at new locations. This is shown in the following example.

A table has been created with a subpartition at host1 and a first level partition at host 2:

```
CREATE TABLE T1 (C1 INT, C2 INT) PARTITION BY RANGE (C1) (
(PARTITION 1 <= VALUES < 50)
SUBPARTITION BY RANGE(C2) (PARTITION 10 <= VALUES < 90 AT LOCATION
'host1:30001'),
(partition value = 50 AT LOCATION 'host2:30002'));
```

The following command repartitions the table specifying new locations at both the parent level and the subpartition level:

```
ALTER TABLE T1 PARTITION BY PARTITION BY RANGE (C1) (
(PARTITION 1 <= VALUES < 50 AT LOCATION 'host3:30003’)
```

```
SUBPARTITION BY RANGE(C2) (PARTITION 10 <= VALUES < 50 AT LOCATION
'host4:30004’,
                           PARTITION 50 <= VALUES < 100,
                           PARTITION 300 <= VALUES < 350),
(partition value = 50));
```

The effects of this are:

- Subpartition one is created at location host 4, as defined specifically for this partition.
- Subpartition two is created at location host 1, as specified in the original table definition. This is because no specific new location has been defined for this partition but its data intersects with the existing range definition. In this case the existing location is kept.
- Subpartition three is created at location host 3. The data range for this partition does not intersect with the existing partitioning schema and therefore the definition given at the parent level can be applied.
- The level 1 partition at host 2 is unaffected.

Related Information

ALTER TABLE Statement (Data Definition)
CREATE TABLE Statement (Data Definition)
Heterogeneous Alter Partition Clauses
Links to Partitioning topics in the SAP HANA Studio guide:
Partition a Non-Partitioned Table
Change a Partitioned Table into a Non-Partitioned Table
Optimize Table Partitioning

8.11.15.8.1 Tracing Partitioning Operations

A history of partitioning operations is maintained which you can review.

Details of all partitioning operations are stored in the database and can be queried from the monitoring view SYS.M_TABLE_PARTITION_OPERATIONS. You can use this to see which tables have been partitioned, when the operation took place and by whom it was performed. You may be able to use this, for example, to review the effectiveness of your data modeling strategy or to review operations which were performed automatically by the system.

Trace data is initially stored in memory in a ring buffer. The size of the buffer is fixed and once it is full data is overwritten starting with the oldest records. However, the buffer periodically flushes the data to disk and saves the data permanently in trace files. Trace files are saved on the indexserver where the partitioning operation was performed with a name in the following format (where the ID number is the file counter value): `indexserver_<host>.<port>.table_partition_operation.<id>.trc`. The frequency of flushing the buffer, the number of trace files retained and the size of the files can be determined by configuration parameters described below.

When you query the system view, the tool uses in-memory mode by default, that is, only the in-memory buffer is searched, this gives the best performance. If you need to query older data you can set a configuration parameter (see `use_in_memory_tracing`) to also query the trace files.

A user with the SYSTEM privilege can query all records in the view, but other users can only retrieve details of their own operation records, that is, where they invoked the operation themselves, or where they are owner of the target table.

An SQL statement is also available to completely clear the system of all trace information. Execute the following command to remove all partition operation trace files and clear the in memory buffer:

```
ALTER SYSTEM CLEAR TRACES ('TABLE PARTITION OPERATIONS');
```

Configuration Options

Configuration parameters to manage tracing of partitioning operations are available in the `table_partition_operation_trace` section of the indexserver.ini file:

Parameter	Description	Default value
enable	You can use this parameter to enable or disable this feature. It is enabled by default.	TRUE
use_in_memory_tracing	If TRUE, then when querying the system view only the memory buffer is searched. Set this parameter to FALSE to search the trace files.	TRUE
maxfiles	The number of files saved.	5
maxfilesize	The maximum file size in MB.	10
trace_flush_interval	The interval is based on the number of operation items which are written to the buffer. This is set to 10 by default so that whenever the tenth item is written to the buffer, the buffer is flushed and all ten items are written to disk.	10

Querying the View M_TABLE_PARTITION_OPERATIONS

The columns available in the view M_TABLE_PARTITION_OPERATIONS are shown in the following table.

```
SELECT * FROM SYS.M_TABLE_PARTITION_OPERATIONS;
```

Column name	Description
START_TIME	Statement start time
DURATION	Statement duration in microseconds
SCHEMA_NAME	Schema name
TABLE_NAME	Table name
USER_NAME	User name
STATEMENT_HASH	Unique identifier for a statement string
STATEMENT_STRING	Statement string. All partition operation statements on column store tables are monitored.

Column name	Description
PARTITION_DEFINITION	Partition definition after executing the statement
OPERATION_TYPE	Operation type, for example: • ADD PARTITION • DROP PARTITION • ALTER TABLE PARTITION BY • MOVE PARTITION • ENABLE DYNAMIC RANGE • DISABLE DYNAMIC RANGE • ALTER PARTITION PROPERTY • MERGE PARTITIONS
IS_ONLINE	TRUE - online operation, FALSE - regular operation.
APPLICATION_SOURCE	Application source information
APPLICATION_NAME	Name of the application
APPLICATION_USER_NAME	Application user name
CLIENT_IP	IP of client machine

8.11.15.8.2 Partitioning Operations in Online Mode

Use the ONLINE mode when moving tables and partitions and repartitioning to reduce the time for which the table is locked.

Moving Partitions Online

The MOVE PARTITION clause is available for use with the SQL ALTER TABLE command and supports an 'online' operation mode. When you move a partition there is an unavoidable period of time when the table is locked and users cannot write new data to the table. If you execute the move operation in online mode (supported for tables partitioned by RANGE or HASH) the amount of time for which a table and its partitions are locked is minimized. This is achieved by using asynchronous table replication to copy the data which only requires an exclusive lock on the table for a very short time at the beginning of the process and again at the end. For times when the table is locked and during the period of the data copy any DML statements submitted (insert, delete, update and so on) are logged. When the copy operation is complete the new replica becomes the new target, the old table or partition is dropped and the logs are replayed to update the table.

Tables which are partitioned by round robin can also be moved but in this case the table requires an exclusive lock for the duration of the move operation and only a single table or partition can be moved in any one statement:

```
ALTER TABLE T_ROUNDROBIN_1 MOVE PARTITION 1 TO 'host1:port2';
```

Scope

The online operation mode can be used to move one or more partitions to one or more selected locations in parallel. This is supported for both heterogeneous (balanced) and non-heterogeneous partitioning.

The online move can only be used for tables which do not already have a replica. A number of replication limitations also apply to this feature (for example, certain table types such as history tables, multistore tables and so on are not supported). The full list of limitations is given in the 'Table Replication Limitations' topic.

Examples

The following examples illustrate usage of the online move operations.

1) Move a table online

Create a table T1 and move the table to a different host using the online option:

```
CREATE COLUMN TABLE T1(A INT PRIMARY KEY, B INT) AT 'MyHost16:30003';
```

```
ALTER TABLE T1 MOVE TO 'MyHost17:30003' ONLINE;
```

2) Move partitions

Create a table T2 and create three partitions using HASH partitioning:

```
CREATE COLUMN TABLE T2(A INT PRIMARY KEY, B INT) PARTITION BY HASH(A) PARTITIONS
3 AT 'MyHost16:30003';
```

Move one partition to a new host:

```
ALTER TABLE T2 MOVE PARTITION 2 TO 'MyHost17:30003' ONLINE;
```

3) Repartition from HASH to RANGE

The table T2 with three HASH partitions can also be repartitioned using ALTER TABLE with the online mode. This example repartitions the table using RANGE partitioning:

```
ALTER TABLE T2 PARTITION BY RANGE(A) (PARTITION 0 <= VALUES < 5, PARTITION 5 <=
VALUES < 20, PARTITION OTHERS) ONLINE;
```

4) This and the following example illustrate moving multiple partitions. Create table T3 and create five partitions using HASH partitioning:

```
CREATE COLUMN TABLE T3(A INT) AT 'MyHost190:34203';
```

```
ALTER TABLE T3 PARTITION BY HASH(A) PARTITIONS 5;
```

5) Move multiple partitions to one or more hosts:

```
ALTER TABLE T3 MOVE PARTITION 1,2,3 TO 'MyHost190:34240’ ONLINE;
```

```
ALTER TABLE T3 MOVE PARTITION 1,2 TO 'MyHost190:34240', PARTITION 3,4 TO
'selibm190:34242’ ONLINE;
```

6) This example illustrates moving partitions in a multi-level scenario:

```
CREATE COLUMN TABLE T4(A INT, B INT) PARTITION BY HASH(A) PARTITIONS 4, HASH(B)
PARTITIONS 2 AT LOCATION 'MyHost41:34403';
```

```
ALTER TABLE T4 MOVE PARTITION 2 TO 'MyHost41:34440’, PARTITION 3,4 TO
'seltera41:34443’ ONLINE;
```

Heterogeneous Repartitioning Online

Repartitioning to apply a heterogeneous two-level partitioning schema can also be done in the 'online' operation mode to reduce the time for which the table is locked. This can be used for range-range and range-hash partitioning (see Heterogeneous Repartitioning for more details). Any property settings (including the load unit value) which have been defined for the table are preserved during this action (properties can be seen in the PARTITIONED_TABLES System View). You can see the progress of this operation in the view M_JOB_PROGRESS, the JOB_NAME value is set to 'Online Repartitioning'.

Heterogeneous repartitioning online can only be used for tables which do not already have a replica. A number of replication limitations also apply to this feature (for example, certain table types such as history tables, multistore tables and so on are not supported). The full list of limitations is given in the 'Table Replication Limitations' topic.

Example

The following example illustrates this, firstly creating a table and then applying a heterogeneous range-range schema. The keyword online is used at the end of the statement.

```
create column table SRC(n int, c char);
```

```
alter table SRC partition by range(n) (
  (partition value = 1, partition value = 2) subpartition by range(c) (partition
value = 'A', partition value = 'B'),
  (partition value = 3)                      subpartition by range(c) (partition
value = 'E', partition value = 'F'),
  (partition value = 4)                      subpartition by range(c) (partition
value = 'G', partition others)) online;
```

Note that an 'online' option for non-heterogeneous repartitioning is also available. SQL syntax details for this are given in the 'Non-heterogeneous Alter Partition Clauses' section of the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Related Information

Table Replication Limitations [page 382]
Heterogeneous Partitioning [page 329]
PARTITIONED_TABLES System View
M_JOB_PROGRESS System View
Non-heterogeneous Alter Partition Clauses

8.11.15.9 Time Selection Partitioning (Aging)

The SAP HANA database offers a special time selection partitioning scheme, also called aging. Time selection or aging allows SAP Business Suite application data to be horizontally partitioned into different temperatures like hot and cold.

SAP Business Suite ABAP applications can use aging, which must not be used for customer or partner applications, to separate hot (current) data from cold (old) data by using time selection partitioning to:

- Create partitions and re-partition
- Add partitions
- Allocate rows to partitions
- Set the scope of Data Manipulation Language (DML) and Data Query Language (DQL) statements.

Because of technology advances, use of SAP HANA Native Storage Extension (NSE) instead of Data Aging is preferable in the case of technical/logging data. Based on customer experience, use of Data Aging for application data is not recommended. In addition to the information given here the following SAP notes may be helpful:

- SAP Note: 2416490 *FAQ: SAP HANA Data Aging in SAP S/4HANA*.
- SAP Note: 2869647 *Guidance for use of Data Aging in SAP S/4HANA*.

Setting the DML and DQL scope is the most important aspect of time selection partitioning. It uses a date to control how many partitions are considered during SELECT, CALL, UPDATE, UPSERT and DELETE. This date may be provided by the application with a syntax clause and it restricts the number of partitions that are considered.

For example a SELECT statement may be issued that retrieves all data having a date greater or equal to May 1st, 2009. It shall also include the current/hot partition. On the other hand, UPDATE operations can also be restricted in the same way. If a date is provided, the current partition is also always included.

> **Note**
>
> Tables with time selection partitioning cannot be converted into any other kind of tables using `ALTER TABLE`.

Unique Constraints for Cold Partitions

By default SAP HANA enforces unique constraints on all partitions. The application may actively overrule this behavior for cold partitions though. This requires that the applications enforce uniqueness for cold partitions by themselves. Duplicate keys are then considered to be application errors.

The reason is that typical OLTP workload in SAP Business Suite for SAP HANA is executed on the current/hot partition and its performance shall not be affected by unique checks for cold partitions that are not relevant for typical OLTP processing.

If the application overrules the uniqueness constraints:

- A row of the current/hot partition may be in conflict with a row of a cold partition,
- A row of a cold partition may be in conflict with a row of another cold partition, and

- A row within a cold partition may be in conflict with another row within the same cold partition.

Partitioning is transparent from an SQL perspective. If a table has a unique index or a primary key and if it has duplicates in cold partitions, a SELECT may return duplicates for a unique index or primary key. This behavior is correct from a database perspective, but this is considered an application error. The database will return an undefined result set. The only kind of statement that will return a correct result set if duplicate primary keys exist is a SELECT statement, which does nothing but select data with a WHERE clause on the full key (no joins, aggregations, aggregate functions or the like and not complex WHERE conditions). There is no guarantee with respect to the result set for further unique constraints if duplicates exist.

Paged Attributes

Cold partitions may optionally be created as paged attributes. This reduces memory consumption. Memory for resident pages are included in the system views M_CS_TABLES, M_CS_COLUMNS and M_CS_ALL_COLUMNS. The field MEMORY_SIZE_IN_MAIN and related statistics include both the paged and non-paged memory for tables or columns.

Global statistics for resident pages can be found in the M_MEMORY_OBJECT_DISPOSITIONS view. The number and size of pages used by paged attributes are tracked in PAGE_LOADABLE_COLUMNS_OBJECT_COUNT and PAGE_LOADABLE_COLUMNS_OBJECT_SIZE, respectively.

Converting Aging tables to Column Store Tables

In some specific cases it is possible to convert time selection partitioned tables to column store tables.

- Single-level aging partitioned table can be converted to non-partitioned table
- Two-level aging partitioned table can be converted to one-level partitioned table with second-level time selection partitions merged
- Single-level or two-level aging multistore tables can be converted to regular column store partitioned tables by firstly moving partitions from extended storage to the column store and then converting. Refer to the topic *Convert a Multistore Table to a Partitioned Column Store Table* in the SAP HANA Dynamic Tiering documentation set for more details.

Examples to illustrate this are given here showing firstly the command to create the table and then alter it:

Example

1) Convert single-level aging table and merge the partitions to an unpartitioned column store table

```
CREATE COLUMN TABLE TAB (A INT, B INT PRIMARY KEY, _DATAAGING NVARCHAR(8))
PARTITION BY RANGE (_DATAAGING)
(USING DEFAULT STORAGE (PARTITION value = '00000000' IS CURRENT,
PARTITION '20100101' <= VALUES < '20110101', PARTITION OTHERS))
WITH PARTITIONING ON ANY COLUMNS ON
FOR NON CURRENT PARTITIONS UNIQUE CONSTRAINTS OFF
FOR DEFAULT STORAGE NON CURRENT PARTITIONS PAGE LOADABLE;
```

```
ALTER TABLE TAB MERGE PARTITIONS;
```

2) Convert two-level aging-partitioned table to a single-level column store partitioned table (HASH-RANGE to HASH)

```
CREATE COLUMN TABLE TAB (A INT, B INT PRIMARY KEY, _DATAAGING NVARCHAR(8))
PARTITION BY HASH(B) PARTITIONS 2, RANGE (_DATAAGING)
(USING DEFAULT STORAGE (PARTITION value = '00000000' IS CURRENT,
PARTITION '20100101' <= VALUES < '20110101', PARTITION OTHERS))
WITH PARTITIONING ON ANY COLUMNS ON
FOR NON CURRENT PARTITIONS UNIQUE CONSTRAINTS OFF
FOR DEFAULT STORAGE NON CURRENT PARTITIONS PAGE LOADABLE;
```

```
ALTER TABLE TAB PARTITION BY HASH (B) PARTITIONS 2;
```

Note that in this case the partition number must be 2.

3) Convert two-level aging-partitioned table to single-level column store partitioned table (RANGE-RANGE to RANGE)

```
CREATE COLUMN TABLE TAB (A INT PRIMARY KEY, B INT, _DATAAGING NVARCHAR(8))
PARTITION BY RANGE(A) (PARTITION VALUE = 1, PARTITION 10 <= VALUES < 20,
PARTITION OTHERS), RANGE (_DATAAGING)
(USING DEFAULT STORAGE (PARTITION value = '00000000' IS CURRENT,
PARTITION '20100101' <= VALUES < '20110101', PARTITION OTHERS))
WITH PARTITIONING ON ANY COLUMNS ON
FOR NON CURRENT PARTITIONS UNIQUE CONSTRAINTS OFF
FOR DEFAULT STORAGE NON CURRENT PARTITIONS PAGE LOADABLE;
```

```
ALTER TABLE TAB PARTITION BY RANGE(A) (PARTITION VALUE = 1, PARTITION 10 <=
VALUES < 20, PARTITION OTHERS);
```

Note that in this case the partition spec must be the same as the first-level spec of the aging table.

Related Information

SAP HANA Native Storage Extension [page 196]
Convert a Multistore Table to a Partitioned Column Store Table
SAP Note 2416490
SAP Note 2869647

8.11.15.10 Partitioning Consistency Check

You can call general and data consistency checks for partitioned tables to check, for example, that the partition specification, metadata and topology are correct.

There are two types of consistency checks available for partitioned tables:

1. General check
   Checks the consistency among partition specification, metadata and topology.
2. Data check
   Performs the general check and additionally checks whether all rows are located in the correct parts.

To perform the general check, execute the following statement:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK_PARTITIONING', '<schema>', '<table>')
```

To perform the extended data check, execute:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK_PARTITIONING_DATA', '<schema>', '<table>')
```

If any of the tests encounter an issue with a table, the statement returns a row with details on the error. If the result set is empty (no rows returned), no issues were detected.

If the extended data check detects, that rows are located in incorrect parts, this may be repaired by executing:

```
CALL CHECK_TABLE_CONSISTENCY('REPAIR_PARTITIONING_DATA', '<schema>', '<table>')
```

> ⓘ Note
>
> The data checks can take a long time to run depending on the data volume.

Related Information

Table Consistency Check [page 283]

8.11.15.11 Designing Partitions

There are a number of factors to consider to optimize the design of your data partitioning strategy including how it will affect select and insert performance and how it will adjust to data changes over time.

Different partitioning strategies need to be tested to determine the best one for your particular scenario. Based on your tests you should choose the partitioning strategy that shows the best performance for your scenario. The design principals listed here are aids to help you decide on the correct partitioning strategy for your scenario.

> ⓘ Note
>
> SAP Business Warehouse on SAP HANA handles partitioning itself. Do not interfere with its partition management unless this has been recommended by SAP.

Query Performance

- For replicated dimension tables the database tries to use replicas that are local to the fact table partitions.
- Partition pruning analyzes the WHERE clauses and seeks to reduce the number of partitions. Try to use partitioning columns that are often used in WHERE clauses. This reduces run time and load.
- Usually hash partitioning is the best partitioning scheme for the first level, especially in scale out scenarios. This is because the client may already use pruning on the client machine and send the query directly to

the host that holds the data, where possible. This is called “client-side statement routing”. This is especially important for single select statements.

- Use as many columns in the hash partitioning as required for good load balancing, but try to use only those columns that are typically used in requests. In the worst case only single select statements may leverage pruning.
- If tables are joined with each other, it is beneficial if the tables are partitioned over the same columns and have the same number of partitions. This way the join may be executed locally in scale out scenarios and the network overhead is reduced.
  - This guarantees that the matching values are in a partition with the same part ID. You have to put all parts with the same ID on the same host.
- Queries do not necessarily become faster when smaller partitions are searched. Often queries make use of indexes and the table or partition size is not significant. If the search criterion is not selective though, partition size does matter.

Data Manipulation Language (DML) Performance

- If insert performance is key to your scenario, a larger number of partitions might show better results. On the other hand, a higher number of partitions may reduce query performance.
- Partition pruning is used during DML operations.
- For replicated column store tables, all DML operations are routed through the host with the master partition (where the replica with Part ID 1 is located).
- If there is a unique constraint on a non-key column, the performance will suffer exponentially with the number of partitions on other servers. This is because the uniqueness on all partitions has to be checked. Therefore, if partitioning is required, consider a low number of partitions and ideally put all partitions on the same host. This way the number of remote calls is reduced.

Data Lifecycle

If time-based partitioning is suitable for the dataset being partitioned, it should always be used as it has a number of advantages:

- The runtime of a delta merge is dependent on the size of the main index. This concept leverages the fact that new data is inserted into new partitions whereas data in old partitions is infrequently updated. Over time, the formerly new partitions become old and new partitions are being created. Delta merges on old partitions are not required anymore. This way the overall runtime of delta merges does not increase with the table size, but remains at a constant level. Using time-based partitioning often involves the use of hash-range partitioning with range on a date column. This requires knowledge of the actual values for range partitioning.
- By using explicit partition management, new partitions can be created, for example, one partition per calendar week and old partitions may be dropped entirely rather than deleting individual rows.
- If you split an index, always use a multiple of the source parts (for example 2 to 4 partitions). This way the split will be executed in parallel mode and also does not require parts to be moved to a single server first.
- Do not split/merge a table unless necessary. These operations write all data into the log which consumes a high amount of disk space. Moreover, the operations take a long time and locks the table exclusively

(only selects are allowed during partitioning operations). ADD PARTITION can be used to add additional partitions. If there is no 'others' partition, this call only creates a new partition which is fast and happens in real time after an exclusive lock of the table was acquired. On the other hand, if the table has an others partition, a call to ADD PARTITION causes the existing others partition to be split into a new others partition and newly requested range. This is a costly operation. Therefore it is recommended, that if ADD PARTITION is used frequently in a scenario, the table shall not have an others partition.

Partition Size

- Due to the high number of factors to be considered when evaluating a partitioning scheme a recommendation for partition sizes cannot be provided. If you do not know if you will partition a table at all or with how many partitions you need to start with measurements. Here are some suggested starting points:
  - If a table has less than 500 million rows, do not partition it at all unless:
    - The corresponding tables in joins are partitioned. If they are try to find mutual partitioning columns.
    - Table growth is expected. Since re-partitioning is time consuming, it is recommended to split a table while it is still small.
  - If your table has more than 500 million rows, choose 300 million per partition.
  - Keep in mind that a partition must not exceed 2 billion rows.
- Be aware that a higher number of partitions might lead to higher memory consumption as each partition has its own exclusive dictionary, which is not shared. If each partition stores disjunctive values, this is not an issue. On the other hand, if each partition has similar or the same values this means that the dictionaries have similar data which is stored redundantly. In this case consider using fewer partitions

Table Design

- If the data is replicated into the SAP HANA database, it might be fair from a data consistency perspective to remove a primary key or to extend the key (since the key constraint is enforced in the source database). This way you might be able to have multiple tables with the same partitioning columns even though the original database design would not have allowed it. Having the same partitioning columns is ideal as related data may reside on the same physical host and therefore join operations may be executed locally with no or hardly any communication costs.
- When designing database schemas for dependent hosts, for example, a database structure for business objects with header and leaf nodes, do not use a single GUID column as the primary key. In such a case it is hardly possible to have all related data (for example, a business object instance) on the same host. One option might be to have a GUID as the primary key in the header table and each host, irrespective of its level, could have that GUID as the first primary key column.
- Do not define a unique constraint on a partitioned table unless absolutely necessary.
- On the second partitioning level, a non-primary key column may be used. Still, the unique constraint has to be enforced on all parts of the respective first-level partition. Since all parts of one first-level partition are moved as a whole, this unique check is always local.

- In case the database table is replicated from another database, an others partition for range is generally recommended. If a proper range is not defined, the insert statement will fail and the data will not get replicated properly.
- Ideally tables have a time criterion in the primary key. This can then be used for time-based partitioning. Number ranges and so on can also be used. The advantage of number ranges is that it is easy to form equally sized partitions, but on the other hand it introduces an administrative burden the amount of data that is loaded needs to be closely monitored and new partitions need to be created in advance. In case of actual dates, you only need to periodically create new partitions, for example, before a new quarter starts.

Other Considerations

- Use GET_NUM_SERVERS() in scripts for hash and round-robin partition specifications. This way Table Placement is used to calculate the number of partitions that will be used in the given landscape.
- If it is likely that a table has to be re-split in future and range partitioning is used, define an others partition. (If it is not defined upon table creation, it can be created afterward and if required dropped after the split operation).
- To check whether a table is partitioned, do not consider the existence of a partition specification in the metadata. Instead check IS_PARTITIONED in M_TABLES or for the existence of parts, for example in M_CS_TABLES. It is allowed that a partition specification is defined which does not immediately lead to a partitioned table.

8.11.15.11.1 Static and Dynamic Partition Pruning

An important partitioning strategy to improve performance is to match partitions wherever possible with the most frequently queried data so that data pruning is possible.

Pruning takes place automatically in the background and tries to eliminate any unnecessary partition from the selection which is not essential for the query result. If your partitions are designed to support this, pruning can avoid accessing and loading into memory partitions which are not required which reduces the overall load on the system. The classic use case for effective pruning is where date-based partitions are used. For example, if a table is partitioned by year, a query restricted to the data of a single year is executed only on the partition with data for the selected year.

Two forms of partition pruning are possible: static and dynamic.

- **Static partition pruning** is based on the partition definition. The query optimizer analyzes the WHERE clause of queries to determine whether or not the filters match the given partitioning specification of a table. If a match is found, it may be possible to target only the specific partitions that hold the data and thus reduce the number of partitions being queried.
- **Dynamic partition pruning** is content-based and can be applied to historical partitions of a HANA aging table. This type of pruning takes place at run time but the decision about which partitions to load is based on pre-calculated column statistics of the data in selected columns. Based on the existence of statistics valid for partition pruning (which must be up to date at the time the query is processed), query evaluation can determine if it is necessary to read the partitioned column of data and load it into memory.

Statistics for Dynamic Partition Pruning

Statistics for dynamic partition pruning are created and maintained as part of the SAP HANA data statistics functionality and are maintained and used only in the indexserver instance where the partition resides. Statistics are explicitly created and refreshed through SQL statements (such as CREATE STATISTICS) and to use dynamic pruning you must first run CREATE STATISTICS for a specified table, partition and column (see following *Enabling* subsection). Any column other than the partition key can be used. The statistics for dynamic partition pruning are persisted with other statistic values in the system catalogs (this takes place asynchronously in the background), however, the pruning statistics values are also cached in memory for the pruning process.

Statistics for pruning have a limited life-span: as new data is added to the column the content of these statistics becomes stale. It is therefore necessary to periodically refresh the pruning statistics. This is done automatically each time delta merge runs and can also be done on demand (see DDL statements below). The statistics are considered to be invalid as soon as new data is added to the column and pruning is then no longer applied to the column until the pruning statistics have been refreshed.

With historically aged data partitions, it is likely that the greatest number of inserts and deletions occurs in the current time period. The older time partitions will be more stable, and for this reason dynamic pruning is only applied to time-based selections on historical data not the current time period.

Enabling Dynamic Partition Pruning

Static and dynamic pruning are designed as integral features of the query optimizer and execution processes.

Statistics for dynamic partition pruning must be explicitly created for a specific table using the CREATE STATISTICS command as shown in the following example (also applies to ALTER STATISTICS). This creates statistics for all partitions:

```
CREATE STATISTICS test_dynPruning_tab1_col1
                    ON tab1 (col1)
                    TYPE SIMPLE VALID FOR DATA DEPENDENCY;
```

For dynamic partition pruning the statistics object type 'SIMPLE' is required with the property set 'VALID FOR DATA DEPENDENCY' as shown here. It is only applicable to partitioned column store tables and multistore tables using time selection partitioning. Only the following datatypes are supported for the statistics columns: VARCHAR (strings with numerical content), INTEGER, DECIMAL, DATE.

Executing the CREATE STATISTICS statement in this way initializes the column for dynamic pruning and populates the pruning statistics. The pruning statistics values are recalculated either by delta merge or, additionally, an explicit refresh statement is available to recalculate the statistics on demand:

```
REFRESH STATISTICS test_dynPruning_tab1_col1
```

Similarly, a DROP STATISTICS command is available and can be used in the same way to drop pruning statistics for a table.

Refer to the *SAP HANA SQL and System Views Reference* for full details.

Monitoring Views and Tracing

Information about statistics for dynamic partition pruning is available in the M_DATA_STATISTICS view.

Diagnostics information is also available. You can see debug details of the pruning process by setting the trace level of the `part_pruning` configuration parameter (in the trace section of the indexserver.ini file) to **debug**.

See also *Configure Traces* for more information.

Related Information

Configure Traces in SAP HANA Studio
Time Selection Partitioning (Aging) [page 357]

8.11.15.11.2 Creating an Effective Partitioning Scheme

This checklist demonstrates how to choose a good partitioning scheme for given tables.

- Tables of above 500 million rows are good candidates for partitioning. This also applies to small tables that are often joined with tables of above 500 million rows.
- If the table has a unique index (other than the primary key), the table may be partitioned, but the additional unique checks introduce a performance penalty.
- Check the primary key.
  - If none exists, any columns may be used for Hash partitioning.
  - If one is present, identify the minimal set of columns that are required to have equally balanced partitions; a sufficiently high number of distinct values is required. Keep in mind that if these columns are all in the WHERE clause of a query, partition pruning may be leveraged.
  - In the case of tables that are replicated into SAP HANA, it may be legitimate to drop the primary key since it is checked in the source database.
- Take other tables into consideration that are often used in joins with the current table. Ideally they have the same number of partitions and partitioning columns.
- Identify time-based attributes; this may be a date, year or at least a sequence. Use them for time-based partitioning. Ideally this column is part of the primary key.
- If you define range partitioning, decide whether or not you require an others partition. Ideally, no others partition is required.
- Decide on the number of partitions. Use Table Placement rules, if applicable.
- In case of a scale out system, move all corresponding partitions to the respective hosts.
- Run extensive performance tests with the most-prominent queries and/or DML load. Try to use analytical views. Vary partitioning columns, partitioning schemes and the number of partitions.

8.11.15.11.2.1 Partitioning Example

This example describes an initial and subsequently improved partitioning schema for a database storing test results.

Assume that for each make various tests run. So many rows have to be stored for a TEST_CASE and MAKE_ID.

Original Table Design

- The original table design suggested having a sequence number as the only primary key column. There are time columns marking the start and end of the test run.
- The table was partitioned by Hash over the sequence number and by range over the start date.

Access Pattern

There are two prominent ways how the data is accessed:

1. Select all test results for a make (“Did my make pass the tests?”)
2. Select results of a single test for all makes within the last month (“How is the history of my test? Has an error happened before?”)

So typically either the TEST_CASE or the MAKE_ID is in the WHERE clause, sometimes both when investigating details of a test run.

Problems with this Original Table Design and Partitioning

- The sequence number is not often part of the WHERE clause and hence all partitions are considered in the query. This is especially an issue in scale-out landscapes where OLTP-like queries are ideally only executed on a single node.
- There is a hash-range partitioning with range on a date column. This allows time-based partitioning to be used. But the date column is not part of the primary key. Therefore the unique constraint on the primary key has to be ensured by checks with the parts of the first-level partition.

Suggested Table Design

- Have TEST_CASE, MAKE_ID and SEQ_NUMBER as the primary key. The actual requirement that there are uniquely identifiable rows for a combination of TEST_CASE and MAKE_ID is met.
- Partition by hash-range with hash over TEST_CASE and range over MAKE_ID. The MAKE_ID increases over time and therefore is also a good column to use for time-based partitioning.

Reasoning

- No primary key checks with other partitions (using range on primary key column).
- Good pruning since partitioning columns match the typical access pattern.
  - If the query has the MAKE_ID in the WHERE clause (compare query type 1), all servers that hold partitions have to be considered, but only a single partition per server.
  - If the query has the TEST_CASE in the WHERE clause (compare type 2), only one server has to be considered (compare Client-Side Statement Routing), but all partitions on that server.
  - If MAKE_ID and TEST_CASE are in the WHERE clause, only a single partition on one server has to be considered.

In this scenario there is one type of query which has the effect that all servers that hold partitions are considered. This is not ideal, but cannot always be prevented depending on the nature of the data and access patterns.

8.11.15.11.3 Partitioning Limits

General restrictions that apply to the use of partitioning are explained here.

General Limitations

- The maximum number of partitions for one table is 16000. A table may be re-partitioned as often as required. The limit of 16000 partitions is independent from the location of the partitions in a distributed (scale-out) landscape.
- When using an equidistant series and table partitioning, for efficient compression ROUND ROBIN partitioning should not be used. HASH or RANGE partitioning should be used so that records with the same series key are in the same partition.

History Tables

Tables with history tables can also be partitioned. A history table is always partitioned with the same partitioning type as the main table.

- The 2 billion rows barrier is also valid for the history tables and in case of partitioned history tables this will be also hold true on a per partition basis.
- If a table uses multi-level partitioning, it is possible to use partitioning columns which are not part of the primary key. This feature cannot be used in conjunction with history tables.
- Tables with history cannot be replicated.

Partitioning Columns

The supported data types for partitioning columns for hash and range partitioning are listed in the corresponding subsections.

- The data type of columns used as partitioning columns must not be changed.
- If a table has a primary key, a column must not be removed from the primary key if it is being used as partitioning column. In the case of a multi-level partitioning, this applies to the first level. It is always possible to remove the entire primary key.
- For all kinds of hash partitioning (hash, hash-hash and hash-range) partitioning column names must not contain commas (“,”), dollar signs (“$”) or round opening parentheses (“(“).
- For homogenous (balanced) range partitioning, partitioning column names must not contain spaces, commas (“,”), dollar signs (“$”) or round opening parentheses (“(“).
- Also for homogenous range partitioning, range partition values must not contain commas (“,”), minus signs (“-“), asterisks (“*”) or the space character.

Related Information

SAP Note 2044468
Range Partitioning [page 323]
Hash Partitioning [page 320]

8.11.15.12 System Views for Monitoring Partitions

A number of system views allow you to monitor your partitions.

System View	Description
TABLE_PARTITIONS	Contains partition-specific information for partitioned tables including details of level 1 and level 2 ranges. This also includes table placement grouping properties GROUP NAME, GROUP TYPE, SUBTYPE. For heterogeneous partitioning the properties NODE_ID and PARENT_NODE_ID are shown so that the hierarchical relationships between partitions can be seen. In this view only values entered by the user are shown (not inherited values from the parent node).
M_TABLE_PARTITIONS	Contains column table information per partition. For heterogeneous partitioning the properties NODE_ID and PARENT_NODE_ID are shown so that the hierarchical relationships between partitions can be seen. In this view implicit inherited values from the parent node are shown.
PARTITIONED_TABLES	General partitioning information for all partitions of a table
TABLE_COLUMNS	Contains table column information.
M_CS_TABLES	Shows run time data per partition (the PART_ID value is the sequential number of the partition). Be aware that after a split/merge operation the memory size is not estimated and therefore the values show zero. A delta merge is required to update the values.

System View	Description
M_TABLES	Shows row counts and memory usage in an aggregated way for partitioned tables (IS_PARTITIONED set to TRUE). Information is based on M_CS_TABLES.
M_CS_PARTITIONS	Shows which partitions or sub-partitions in column store tables form a partition group. This information is for example required when partitions or groups of sub-partitions are to be moved to another host.
M_TABLE_PARTITION_STATISTICS	Shows selection statistics for each partition. In particular, it shows the number of times the partition is selected and includes the timestamp of the last time it was selected. This view is populated if the configuration parameter `partition_statistics_select_enabled` in the `partitioning` section of the indexserver.ini file is enabled.
M_TABLE_PRUNING_STATISTICS	Provides more information about which partitions are used and how pruning is being applied. This monitoring view is not populated by default because collecting pruning statistics may have an impact on performance when executing statements. You can enable this feature by setting the `enable_pruning_statistics` configuration parameter to **`True`**: `alter system alter configuration ('global.ini', 'system') set ('partitioning', 'enable_pruning_statistics' ) = 'true' with reconfigure;`

Related Information

TABLES System View
M_CS_TABLES System View
M_TABLES System View
M_CS_PARTITIONS System View - Deprecated
M_TABLE_PARTITION_STATISTICS System View

8.11.15.13 Multistore Tables

A multistore table is a range partitioned SAP HANA column table that has at least one partition in SAP HANA and other partitions in different physical stores associated with SAP HANA.

Multistore capability is currently implemented through SAP HANA dynamic tiering, which provides extended storage as an alternate physical store for SAP HANA table partitions. Multistore tables are managed and monitored just like other columnar tables, but offer the advantages of disk-based storage for data aging through extended storage.

Multistore data management allows flexible administration, such as:

- Moving data between extended or default storage.
- Creating or dropping partitions directly in either extended or default storage.
- Repartitioning a table, if new partitioning does not move data between default and extended storage.

Data modification operations let you easily change the storage type for data. For example, you can change the storage type for a partition when you import data into a table or modify the data in a multistore table during an insert operation.

> **Note**
>
> Multistore functionality does not support all features of SAP HANA. For details of features supported, see *Multistore Tables* in the *SAP HANA Dynamic Tiering Administration Guide*.

Related Information

Manage a Multistore Table (Dynamic Tiering Administration Guide)

8.11.16 Table Placement

Table classification and table placement configuration, enhanced by partitioning, build the foundation for controlling the data distribution in a SAP HANA scale-out environment.

Table Classification and Placement

Application data is usually stored in a multitude of database tables, and data from several of these tables is combined via SQL operations like join or union when it is queried. As these relations between different tables are defined in the application code, this information is not available within SAP HANA. The table classification feature provides a possibility to push down this semantic information in the database by allowing administrators to define groups of tables. This information can be used, for example, when determining the number of partitions to be created, or, in the case of a scale-out landscape, the node where to locate the tables or partitions.

The classification is performed by providing each table with a group name, group type and subtype. Based on combinations of these elements as well as the table names and schema names, a set of configuration values can be defined as table placement rules, the rules are used to control, for example, the placement of partitions or the number of partitions during operations like table creation or redistribution. By doing this, associated or strongly related tables are placed in such a way that the required cross-node communication is minimized for SQL operations on tables within the group.

Table placement rules are applied during system migration or table creation, but it may also be necessary to adjust the location or the number of partitions on an ongoing basis for handling data growth. Table redistribution can therefore also be run on demand to optimize the landscape as the system evolves. Repartitioning is always necessary, for example, for any table or partition in the database which reaches the maximum count of 2 billion rows.

The following tools are available to perform table repartitioning and redistribution; these tools evaluate the current landscape and determine an optimized distribution:

- SAP HANA Table Redistribution (available in SAP HANA cockpit, SAP HANA studio, or executed from the command line)
- Data Distribution Optimizer (part of SAP HANA Data Warehousing Foundation)

Balancing a SAP HANA scale-out landscape with these tools is done in two stages:

1. Generation of a plan based on table placement rules (described in detail below). After generating the plan, you can review it and adjust the definition of the rules if required.
2. Execution of the plan which implements the partitioning and distribution changes.

Because split table and move table are operations that require table locks, the execution of the plan should not be performed during a period where there is heavy load on the database.

> **Note**
>
> Please refer to the corresponding documentation for details. In particular, the *SAP HANA SQL and System Views Reference* gives details of all SQL syntax for the operations described here.

Table Classification and Table Placement Rules

Table placement rules are defined in the table SYS.TABLE_PLACEMENT; system privilege TABLE ADMIN is required to maintain these settings. Placement rules basically address the following areas:

- Classification: related tables which must be located together are organized in groups
- Configuration settings to manage partitioning (number of initial partitions, split threshold and so on)
- Physical distribution or location of tables or partitions in the server landscape.

When creating a table, if the defined rules match with a table or a table group, SAP HANA will consider them while creating the table.

Please keep in mind that partition specifications still need to be defined by the application.

Related Information

Scaling SAP HANA [page 1079]
CREATE TABLE Statement (Data Definition)
ALTER TABLE Statement (Data Definition)
Extension Node [page 243]
Redistributing Tables in a Scaleout SAP HANA System [page 405]

8.11.16.1 Table Classification (Groups)

Associated tables can be classified by a common table group.

The SQL interface of SAP HANA provides three possible kinds of classifications: group name, group type and subtype. Tables which have been classified with group information are included in the SYS.TABLE_GROUPS

table and you can review classification details in the monitoring view SYS.TABLE_GROUPS (details below). Tables with the same group name are kept on the same host, or, in the case of partitioned tables that are distributed over several hosts, corresponding first-level partitions are distributed for all tables the same way.

One table in the group is defined as the leading table and its table placement settings are applied to all other tables in the group. This can be, for example, the location, or in the case of partitioned tables (if SAME_PARTITION_COUNT is set in SYS.TABLE_PLACEMENT, see below), the number of first-level partitions.

Note that specific applications, like SAP BW, classify objects automatically as they are created and must not be changed manually.

SYS.TABLE_GROUPS

Column	Description
SCHEMA_NAME	The schema name.
TABLE_NAME	The table name.
GROUP_NAME	The group name
GROUP_TYPE	The group type. Example: in SAP BW there are predefined group types that classify the tables associated with a BW object, such as: sap.bw.cube (InfoCubes), sap.bw.dso (DataStore Objects), sap.bw.psa (PSA tables) and so on.
SUBTYPE	The subtype. This is required for some group types, for example in SAP BW: • a table belonging to an InfoCube (group type sap.bw.cube) can be a fact table (subtype FACT_IMO) or a dimension table (subtype DIM) • a table belonging to a DataStore Object (group type sap.bw.dso) can be an active table (subtype ACTIVE), an activation queue (subtype QUEUE), or a Changelog (subtype CHANGE_LOG)
IS_GROUP_LEAD	Determines the leading table within a group. If none is set, the largest, partitioned, non-replicated column store table is used as leading table.

Example

In this example the following three tables that belong to SAP Business Warehouse (BW) DataStore Object with technical name ZFIGL, all have the same group name value and form the group 'ZFIGL'.

Example Table Group

Table Name	Group Type (GROUP_TYPE)	Subtype (SUBTYPE)	Group Name (GROUP_NAME)
/BIC/AZFIGL00	sap.bw.dso	ACTIVE	ZFIGL
/BIC/AZFIGL40	sap.bw.dso	QUEUE	ZFIGL
/BIC/B0000197000	sap.bw.dso	CHANGE_LOG	ZFIGL

For native applications, the application developer can define a grouping manually, for example, by grouping tables together that are often joined. The following statements show examples of setting table group attributes using CREATE and ALTER table:

```
CREATE COLUMN TABLE "TEST".A1 (A INT) GROUP TYPE ABC GROUP SUBTYPE T
```

```
ALTER TABLE "SAPLG1"."/BIC/MUCSTR000000" SET GROUP NAME "ABC"
```

Creating Groups Dynamically

There are a number of options for creating table groups dynamically based on the information available in the SQL Plan Cache.

- The Join Path Analysis tool is available within the Data Distribution Optimizer, or the ABAP grouping report for S/4HANA scale-out.
- Table Redistribution also includes an optional preparation step to integrate the Group Advisor tool into the plan generation process to create table groups dynamically.

Related Information

Configuration of Table Redistribution [page 412]

8.11.16.2 Table Placement Rules

The TABLE_PLACEMENT table provides a customizing interface which can be used for the dynamic management of partitions and locations.

The TABLE_PLACEMENT Table

The content of the table is shown here. Parameters 6-10 define how a table or a group of tables is partitioned; further information about usage is provided in the subsections and topics which follow.

SYS.TABLE_PLACEMENT

#	Column	Description
1	SCHEMA_NAME	The schema name (as for table groups).
2	TABLE_NAME	The table name.
3	GROUP_NAME	The group name.
4	GROUP_TYPE	The group type.
5	SUBTYPE	The subtype.
6	MIN_ROWS_FOR_PARTITIONING	Partitioning rule: the number of records that must exist in the table before the number of first-level partitions is increased above 1.
7	INITIAL_PARTITIONS	Partitioning rule: determines the number of initial partitions to create, for example, HASH 1, HASH 3,
8	REPARTITIONING_THRESHOLD	Partitioning rule: if the row count exceeds this value then further split iterations are considered.

#	Column	Description
9	DYNAMIC_RANGE_THRESHOLD	Applies to tables that use the dynamic range partitioning feature. Overwrites the system default value defined in *indexserver.ini* > *[partitioning]* > *dynamic_range_default_threshold* (10,000,000) for that specific combination of schema / table / group characteristics.
10	SAME_PARTITION_COUNT	Specifies that all partitions of the tables in a group will contain the same number of partitions. Globally maintained in *global.ini* > *[table_placement]* > *same_num_partitions* but in case of several applications with deviating settings, it can be maintained on a more granular level.
11	LOCATION	Location rule: master, slave, all (see following topic on Locations).
12	PERSISTENT MEMORY	Whether or not the data is loaded in persistent memory.
13	PAGE LOADABLE	Whether or not the data is page loadable.
14	NUMA_NODE_INDEXES	The allowed NUMA nodes.
15	REPLICA COUNT	The required number of replicas.
16	REPLICA TYPE	The replication type. The default type is ASYNCHRONOUS.

Partitioning Rules

The partitioning parameters are used to define how a table or a group of tables is partitioned if the table has a first-level partitioning specification of HASH or ROUNDROBIN. RANGE partitioning is not handled in this way.

If the number of rows is lower than MIN_ROWS_FOR_PARTITIONING, the table consists only of one partition. If this minimum row limit is exceeded, the table will be partitioned in as many partitions as are required to fulfill the following constraints:

- The number of partitions is larger or equal to the value of (row count of the table) / REPARTITIONING_THRESHOLD
- The number of partitions is a multiple of INITIAL_PARTITIONS
- The number of partitions is smaller or equal to the number of hosts in case parameter *max_partitions_limited_by_locations* is not set to false and the number of partitions is less than the value of parameter *max_partitions* (see Repartitioning below).

Therefore, if the table has more than one partition, there are at least INITIAL_PARTITIONS partitions, and each partition has less than REPARTITIONING_THRESHOLD records. Here partitions refer to first-level partitions (of type HASH or ROUNDROBIN).

Please note, when a partitioned table is created without an estimated row count (default behavior) a partitioned table is created with INITIAL_PARTITIONS first-level partitions, whereas in a redistribution it is targeted to have a single first-level partition (assuming MIN_ROWS_FOR_PARTITIONING > 0). In specific

applications, creation is performed with an estimated row count, for example, BW with 1 million and therefore it will be created with only one first level-partition (assuming MIN_ROWS_FOR_PARTITIONING > 1,000,000).

Repartitioning

There is no automatic repartitioning when threshold values are exceeded. Instead, this is proposed the next time the redistribution process is executed.

The values entered for partitioning must be consistent with the physical landscape, especially the number of server nodes available:

- If repartitioning is necessary, tables are only repartitioned by doubling the number of existing (initial) partitions. This is done for performance reasons. The maximum number of (first-level) partitions reached by that process is defined by parameter *global.ini* > *[table_placement]* > *max_partitions* (default 12).
- By default, the system does not create more partitions than the number of available hosts (or more specifically possible locations). For example, if INITIAL_PARTITIONS is set to 3 but the distributed SAP HANA database has five possible locations, repartitioning from three to six partitions would not take place. A table can have more than one partition per host if the parameter *global.ini* > *[table_placement]* > *max_partitions_limited_by_locations* is set to false (default true). This rule is disregarded if a higher number of first level partitions is required to partition groups with more than 2 billion records (*global.ini* > *[table_placement]* > *max_rows_per_partition*, default = 2,000,000,000).

How Rules Are Applied

The TABLE_PLACEMENT table is read in such a way that a more specific rule supersedes a more generic one. A complete matrix of priorities is available in SAP Note 1908082 Table Placement Priorities.

For example, an entry with only schema applies to all tables of that schema; additional entries for that schema and specific group types overrule the more general rule.

Monitoring View

You can see the actual table placement settings per table by querying the system view M_EFFECTIVE_TABLE_PLACEMENT. You can see the valid location(s) according to the configuration and for each partitioning parameter the actual values and in the corresponding _MATCH columns the reason (matching rule) for those.

The information how a table is classified, can be reviewed in the monitoring view SYS.TABLE_GROUPS.

Related Information

Table Placement Locations [page 376]
Dynamic Range Partitioning [page 341]
SAP Note 1908082

8.11.16.3 Table Placement Locations

A location is a text value that uses table placement rules at runtime to map to one or more specific volume IDs.

Locations

Locations use table placement rules (specifically table group types) to map to a list of volume IDs; volume IDs are used in preference to a host:port combination because port numbers may change if a takeover occurs. Some predefined locations are available which are linked to hosts by the worker group value, these have a pre-defined relationship to specific volume IDs. You can also define your own locations in the TABLE_PLACEMENT_LOCATIONS table and for each location make an association to one or more volume IDs; example SQL commands for doing this are given below.

The following predefined locations are available:

Location	Description
default	'default' is the name of a worker group and it used as the default location for hosts if no other locations are defined. This value can be seen in the WORKER_CONFIG_GROUPS field in system view M_LANDSCAPE_HOST_CONFIGURATION.
master	This is the master location node. The value 'MASTER' is set as `coordinator_type` for the indexserver in system view M_SERVICES.
slave	Other pre-defined locations are slaves, that is, any `coordinator_type` value for the indexserver in system view M_SERVICES which is not MASTER.
all	The 'all' location includes both masters and slaves.

Worker Groups as Locations

You can also add your own worker group location values for a host. To add custom locations as worker groups use the following 'SET' procedure and specify the host name; multiple worker group values must be separated by a space as shown:

```
call SYS.UPDATE_LANDSCAPE_CONFIGURATION( 'SET
WORKERGROUPS','<hostname>','<name1> <name2> <name3>' )
```

The worker group assignment for a host can be found in entry WORKER_ACTUAL_GROUPS of view M_LANDSCAPE_HOST_CONFIGURATION and accessed by executing the following 'GET' procedure:

```
call SYS.UPDATE_LANDSCAPE_CONFIGURATION( 'GET WORKERGROUPS','<hostname>')
```

Note that the worker group `worker_dt` is a special worker group required for handling warm data in the extension node (see section on Extension Node and following example).

Other User-defined Locations

Location definitions are stored in table TABLE_PLACEMENT_LOCATION where the location is associated with specific volumes. You can add locations to this table and use the keywords INCLUDE and EXCLUDE to specify which volumes the location can use. Locations are visible in system view TABLE_PLACEMENT_LOCATIONS:

LOCATION_NAME	INCLUDE	EXCLUDE
ALL		WORKER_DT
SLAVE		WORKER_DT
MY_GROUP	WORKER_DT, MASTER	
OLTP_PROCESSOR	2,3	
myLocation	5	

New locations and volume assignments are only effective after the next reconfiguration of the indexserver. The currently effective volume IDs for a location are visible in the view M_TABLE_PLACEMENT_LOCATIONS, this is shown in the example which follows.

> ⓘ Note
>
> - In TABLE_PLACEMENT, the value '#1' can be used as an affix to the location group name. This indicates that the partitions of all tables for which that rule apply will be located on a single host of that location group, that is, no distribution of several hosts takes place.
> - If a location group (worker group) is not found in M_LANDSCAPE_HOST_CONFIGURATION, the system checks whether the location group has been defined by the HANA 1.0 configuration. This means whether an entry with the custom location name exists in section [table_placement] in the global.ini file, assigning a comma separated list of volume IDs as value to it.

Example

This scenario creates a separate location for warm data so that tables intended for warm data can be created and correctly located on the warm data volume ID. It shows the three steps: creating new locations, creating corresponding table placement rules, and positioning of newly-created tables on specified volumes. Full details of these statements with further examples are given in the *SAP HANA SQL Reference Guide: ALTER SYSTEM ALTER TABLE PLACEMENT Statement*.

1) Create Two User-Defined Locations

Two user-defined locations are created and the INCLUDE/EXCLUDE keywords are used to manage the related volume IDs of each location:

```
ALTER SYSTEM ALTER TABLE PLACEMENT LOCATION worker_dt SET (INCLUDE => '5') ;
```

```
ALTER SYSTEM ALTER TABLE PLACEMENT LOCATION all_without_worker_dt SET (INCLUDE
=> 'all', exclude=>'worker_dt') ;
```

The relationship between the location and the volume IDs can be seen in the system view M_TABLE_PLACEMENT_LOCATIONS. Here, the new user-defined locations have been added after the predefined locations with the result that one single volume ID (#5) is reserved for location `worker_dt`:

LOCATION_NAME	SYSTEM_DEFINED_VOLUME_IDS	INCLUDE	EXCLUDE	EFFECTIVE_VOLUME_IDS
all	2,3,4			2,3,4
default	2,3,4			2,3,4
master	2			2
slave		All	master	3,4
worker_dt		5		5
all_without_worker_dt		All	worker_dt	2,3,4

2) Create Table Placement Rules

Corresponding rules for these locations are created in the TABLE PLACEMENT table. The locations are added and the `worker_dt` location is associated with a group type for warm data:

```
ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME => 'MY_APPLICATION') SET
(LOCATION=>'all_without_worker_dt');
```

```
ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME =>
'MY_APPLICATION',GROUP_TYPE=>'WARM_DATA') SET (LOCATION=>'worker_dt');
```

3) Create and Locate Tables

When new tables are created they are positioned according to the placement rules and group type settings.

```
CREATE TABLE "MY_APPLICATION"."TABLE_A" (INT A);
```

```
CREATE TABLE "MY_APPLICATION"."TABLE_B" (INT A);
```

```
CREATE TABLE "MY_APPLICATION"."TABLE_C" (INT A) GROUP TYPE "WARM_DATA";
```

The result is that only TABLE_C is created on the indexserver that is connected to volume ID 5. TABLE_A and TABLE_B are created on one of the other indexservers in the landscape.

Related Information

Table Placement Rules [page 373]
Extension Node [page 243]
ALTER SYSTEM ALTER TABLE PLACEMENT Statement (System Management)
(Ext)ALTER SYSTEM ALTER TABLE PLACEMENT Statement (System Management)

8.11.16.4 Pre-defined Table Placement Scenarios

For specific applications, SAP provides recommendations regarding partitioning and table distribution configurations.

For the following scenarios SAP provides recommendations regarding table distribution configurations, for these scenarios SQL implementation scripts and detailed documentation is provided in SAP Notes.

SAP BW powered by SAP HANA

All required steps and recommended settings for SAP BW on HANA 2 are described in SAP Note 1908075 *BW on SAP HANA: Table placement and landscape redistribution*. This includes a zip file with documentation and SQL code to configure various scenarios covering a range of TABLE_PLACEMENT settings depending on the node size (TB per node) and the number of master and slave nodes.

SAP BW/4 HANA

All required steps and recommended settings for SAP BW/4 on HANA 2 are described in SAP Note 2334091 *BW/4HANA: Table Placement and Landscape Redistribution*. This includes a zip file with documentation and SQL code to configure various scenarios covering a range of TABLE_PLACEMENT settings depending on the node size (TB per node) and the number of master and slave nodes.

SAP S/4HANA

For details of scale-out options for SAP S/4HANA refer to SAP Note 2408419 *SAP S/4HANA - Multi-Node Support*. This note includes scripts and configuration settings as well as detailed documentation about table groups and migration.

SAP Business Suite Powered by SAP HANA and S/4HANA

SAP Note 1899817 *SAP Business Suite on SAP HANA database: table placement* includes configuration scripts to set up partitioning and distribution for Suite and S/4 HANA for various support package stack releases.

Further Resources

An introductory Frequently-Asked-Questions document is available on the SCN Community portal: *SAP BW/4HANA and SAP BW-on-HANA with SAP HANA Extension Nodes*

A scenario for SAP BPC (HANA 1 release) is described in SAP Note 2003863 *Enable BPC HANA table distribution*, this note includes table placement settings for BPC systems.

Related Information

SAP BW/4HANA and SAP BW-on-HANA with SAP HANA Extension Nodes
SAP Note 1908075
SAP Note 2334091
SAP Note 1899817
SAP Note 2408419

SAP Note 2003863

8.11.17 Table Replication

In a scale-out system, tables (or selected columns of column store tables) may be replicated to multiple hosts. This can help to reduce network traffic when, for example, slowly changing master data often has to be joined with tables, or partitions of tables, that are located on other hosts.

Synchronous and Asynchronous Replication

Both asynchronous table replication and synchronous table replication are supported. The following table compares these two approaches.

With asynchronous table replication, there can be differences in the data contained in the source table and the replica table. The application developer must take this into account and decide which queries can be routed to the replica tables, which may have an out-dated snapshot.

Synchronous table replication offers similar performance benefits to asynchronous, but it is more transparent. Using synchronous replication, you do not have to modify queries to access replica tables. Either the source table or the replica table is automatically selected and the query is routed to the selected table. The read-only query workload can be load-balanced without any SQL string or application change.

Comparing the table replication types

Optimistic Synchronous Table Replication (OSTR)	Asynchronous Table Replication (ATR)
Advantages: Symmetrical and easy to use because source and replica always have the same state. Application developers therefore do not need to be aware of the existence of replicas. Queries can be routed to the source and replicas evenly and implicitly by the database without having to modify SQL queries. Replicas are activated when the table modification is committed, there is no need for additional activation.	Advantages: Little overhead at the source node; replicating updates with less overhead at the source transactions.
Disadvantages: There is a performance penalty but only to the write transactions commit operations (DML and read transactions are not affected).	Disadvantages: Asymmetry between source and replica, therefore not easy to use. Replicas have a different (possibly outdated) state compared to their source tables. This incurs difficulty in its usage model. That is, the source and its replica are not symmetrical or equivalent to each other and the application developers should explicitly hint which queries can accept such staleness.

> **Note**
>
> In SAP HANA 2.0 a new version of synchronous table replication was implemented: Optimistic Synchronous Table Replication (OSTR). Any replica tables that were created in earlier versions are no longer valid and must be recreated for the new SAP HANA release. Refer to the *Synchronous Table Replication* topic for details.

Things to Consider

Before using table replication, consider the following aspects, refer also to the topic 'Table Replication Limitations' which follows:

- Replicated tables consume main memory when they are loaded, as well as disk space, as each replica has its own independent persistence. Therefore an increased amount of main memory is needed. This needs to be considered for sizing.
- For replica creation, the configuration for table placement is taken into consideration. See topics on Table Placement and Table Distribution.

Table replication must only be used if these considerations are acceptable for your use case.

Status-Aware Routing

Status-aware routing ensures that in a situation where replica tables exist, if replication is disabled, then queries are automatically routed to the source table. It applies to both synchronous and asynchronous replication. For asynchronous replication it may be necessary to query a specific replica table using hints or routing; to do this, replication must be enabled. See topic *Operations for Asynchronous Table Replication* for details.

Views for Table Replication

In the system view TABLE_REPLICAS, the column HAS_DIFFERENT_COLUMNS is set to TRUE for sub-tables and the column HAS_DIFFERENT_PARTITIONS is set to TRUE for asymmetric partitioned replicas.

The view M_CS_COLUMNS provides more details about each column.

The monitoring view M_TABLE_REPLICAS contains detailed information on asynchronous/synchronous table replicas. The view M_TABLE_REPLICAS_RESET contains values accumulated since the last reset of the main view M_TABLE_REPLICAS.

M_TABLE_REPLICATION_VOLUME_STATISTICS and M_TABLE_REPLICATION_VOLUME_STATISTICS_RESET provide statistical information about replication. The columns SOURCE_REMOTE_SOURCE_NAME and REPLICA_REMOTE_SOURCE_NAME are used by remote table replication replicas.

The view M_TABLE_REPLICATION_VOLUME_HISTORY provides history information.

Related Information

Asynchronous Table Replication [page 394]
Synchronous Table Replication [page 403]
Table Replication Limitations [page 382]
Scaling SAP HANA [page 1079]
ALTER TABLE Statement (Data Definition)

8.11.17.1 Table Replication Limitations

General restrictions that apply to the use of table replication.

Scope

- Table replication works only within a single SAP HANA scale-out landscape. It cannot be used for replicating a table across two different SAP HANA landscapes or between SAP HANA and other DBMS instances.
- Source tables can be distributed to multiple nodes but (except for the case of sub-table replication) a source table and its replica cannot be located at the same node.
- Only one identical replica table (or table partition) can exist in a replica node (except for the case of sub-table replication).
- Replica tables used in synchronous or asynchronous replication cannot be used as smart data integration (SDI) replication source tables.

HANA System Replication Scenarios

Table replication is not supported in a HANA System Replication secondary. Replica tables may exist in a secondary (replicated by HSR and persist redo-logs) but data is not replicated to these tables by table replication. Replica tables are only accessible in the primary system; the replicas on the secondary only become active when a system replication takeover occurs.

Supported Table Types

The following table types cannot be set as a replication table:

- History table
- Flexible table
- Temporary table
- Virtual table
- Extended Storage table
- Multistore table
- System-versioned table
- Tables with masked columns, tables with associations, tables with series data.
- Tables with record commit timestamp column

Write Operations

- Write operations cannot be executed at the replica table. Such access will return an error.
- DDL operations cannot be executed directly at the replica table. Such access will return an error.
- DDL operations with online option are not allowed for a replication table.
- DDL operations on a replication source table cannot be executed with a 'DDL auto-commit off' transaction. Such access will return an error.
- It is not allowed for a 'DDL auto-commit off' transaction to execute a write operation on a replication source table after a DDL operation on any table in the same transaction boundary.
- It is not allowed for a 'DDL auto-commit off' transaction to execute a DDL operation on replicated table when its replication status is ENABLED.
- Updates to source tables with asymmetric partitions are possible, but not all DDL statements are supported (add partition, for example, is not supported). The DDLs in the following table are supported, other operations may also work but you should verify the results:

Objects	Supported Actions
Replica	add/drop/move/enable/disable
Source table	drop/move/truncate
Column	add drop/modify possible if the target column is not a partitioning key of replica tables
Primary key	drop add is possible if the target column is already a partitioning key of replica tables
Foreign key	add / drop
Unique constraint	add / drop
Constraint	drop
Check	add

Import and Export

- The binary import on the replicated table (source or replica table) is not allowed. You need to drop the replica tables and then import and re-create the replica tables again.
- Export from the replica table is not allowed. Export from the source table is allowed but the exported data contains only the source table's data. It does not contain any data of the replica.

Related Information

SET TRANSACTION AUTOCOMMIT DDL Statement (Transaction Management)

8.11.17.2 Table Replication Consistency Checks

The following consistency checks are available for replicated tables in the column store. Note that they cannot be run with sub-table replication and asymmetric partition replication.

To perform the general check, execute:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK_REPLICATION', '<schema>', '<table>')
```

In OSTR, source and replica always have the same state. Therefore application developers, to perform a lightweight data check which ensures that all rows are replicated, execute:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK_REPLICATION_DATA_LIGHTWEIGHT', '<schema>',
'<table>')
```

To perform a full data check which ensures all replicas hold the same data in all columns, execute:

```
CALL CHECK_TABLE_CONSISTENCY('CHECK_REPLICATION_DATA_FULL', '<schema>',
'<table>')
```

The data checks can take a long time to run, depending on the data volume.

Related Information

Table and Catalog Consistency Checks [page 282]

8.11.17.3 Sub-Table Replication

Use sub-table replication to replicate a defined subset of table columns to a replica, in situations where it is not necessary to replicate the entire table.

With sub-table replication, you can choose to replicate only the specific columns required by users querying the table; this consumes less memory and can improve performance.

Sub-table replication can be used for any column store table using either synchronous or asynchronous replication. Using sub-table replication, a source table can be replicated to one or more nodes and it can also be replicated to the same node as the source. Each replica can have a different column specification from the source; all replicas are treated like a normal table for query purposes.

Setting Up Sub-Table Replication

To create a sub-table replica, add a replica to an existing source table. You can do this at the time a table is created, or later. A value for the `column list` clause is required to create sub-table replicas.

- Create a replica from an existing table using CREATE TABLE - LIKE.

- Create a replica from an existing table using ALTER TABLE - ADD.

Generally, data definition statements are not permitted on replica tables, but in this case MOVE and DROP statements are supported.

- Move a sub-table replication using ALTER TABLE - MOVE TO.
- Drop a sub-table replication using ALTER TABLE - DROP TABLE.

Example

The following examples are based on a source table SRC.TBL which has columns A, B, C, D, E.

SQL Command	Result
`CREATE COLUMN TABLE REP.TBL LIKE SRC.TBL SYNCHRONOUS REPLICA (A, C, D) AT '<host:port>';`	Create a replica with a new name (REP.TBL) from an existing table (explicit schema name mapping). This example creates a replica with columns A, C, D.
`ALTER TABLE SRC.TBL ADD SYNCHRONOUS REPLICA (B, C, E) AT '<host:port>';`	Create a replica from an existing table keeping the name of the source table (implicit schema name mapping). This example creates a replica containing only columns B, C, E.
`ALTER TABLE REP.TBL MOVE TO '<host:port>';`	Move an explicitly named replica sub-table.
`ALTER TABLE REP._SYS_REP_TBL#1 MOVE TO '<host:port>';`	Move an explicitly named replica sub-table.
`DROP TABLE REP.TBL;`	Drop an explicitly named replica sub-table.

Refer to the monitoring view M_CS_COLUMNS to verify the column information of a column-wise replicated replica.

Related Information

ALTER TABLE Statement (Data Definition)

8.11.17.4 Replicating Asymmetric-Partitioned Tables

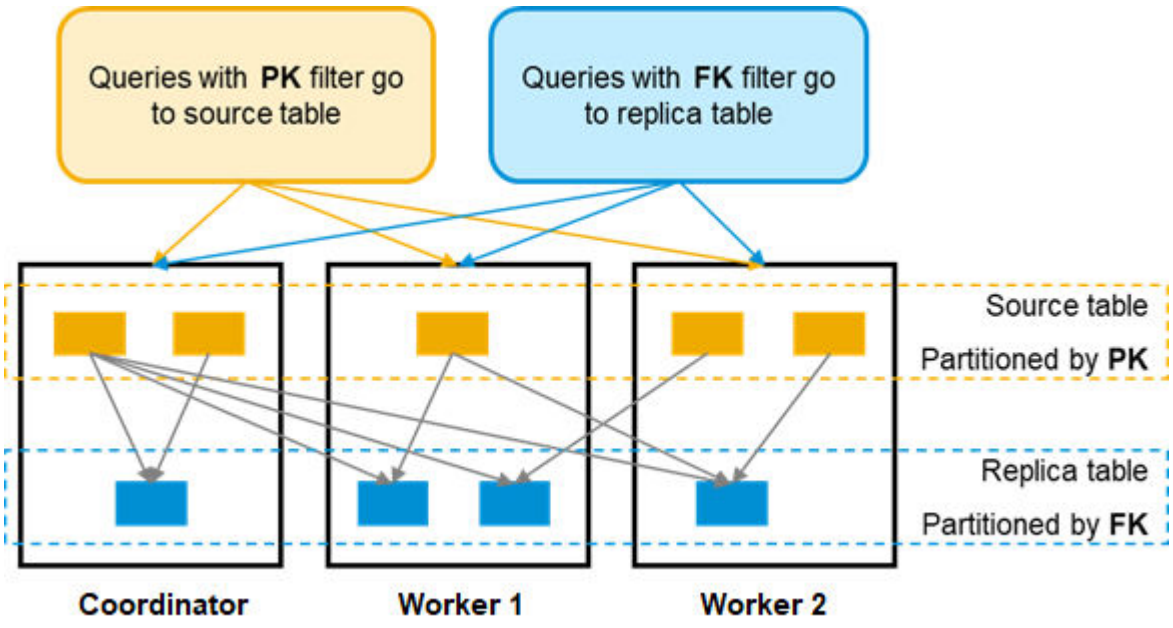
You can create a replica of a partitioned table that has a different partitioning specification to the source table.

To provide flexibility in how replica tables are created, you can replicate partitioned tables in such a way that the replica has a different partitioning specification to the source table.

This is shown in the following illustration based on a scenario using three host servers. A table is partitioned at the first level by ranges on the basis of its primary key (PK) value - giving a result of five partitions which are located on the three servers.

The partitions are replicated to new replica tables by using hash partitioning on the basis of a foreign key (FK) value - which results in four 'replica' tables.

In a scenario like this you can then optimize your database queries to route them directly to the correct replica. There is no automatic routing of queries, but since the replica tables all have specific names, queries can be directed to the appropriate replica by explicitly using the correct table name in the select statement:



This feature can be used with both synchronous and asynchronous replication modes.

In this scenario, the source table and replicas can be located in the same node. You must give asymmetric replica tables and partitions a unique name, and queries to a replica must explicitly name the target replica, for example:

```
SELECT C1, C2, C3, CN FROM MyReplica_Partition3
```

Partitioning to a replica can only be done using HASH and RANGE partitioning. Updates to source tables with asymmetric partitions are possible, but not all DDL statements are supported, see Table Replication Limitations for details.

Note that replica tables with an asymmetric partition have two additional columns:

- $reference_rowid$ stores the corresponding row ID in the source table without partition information
- $reference_partid$ stores the logical partition ID of the corresponding record in the source table.

These are required because the row id of a record in the replica may be different to the row id in the source.

Example: Creating and Replicating Partitioned Tables

Refer to the *SAP HANA SQL and System Views Reference* for full details of syntax and all options supported.

SQL Command	Result
`CREATE COLUMN TABLE SRC_TABLE (C1, C2, C3, … CN, primary key (C1, C2, C3)) PARTITION BY HASH (C1) PARTITIONS 32;`	Create a source table (SRC_TABLE) with a number of columns. The table has 32 hash-partitioned partitions based on the values of column 1 (C1).
`CREATE COLUMN TABLE REP_TABLE1 LIKE SRC_TABLE REPLICA PARTITION BY HASH (C2) PARTITIONS 8;`	Create a replica table (REP_TABLE1) which has an asymmetric partitioning scheme compared to its source table. All columns of the source table are replicated but the replica is partitioned on column 2 (C2) to create 8 partitions.
`CREATE COLUMN TABLE REP_TABLE2 LIKE SRC_TABLE REPLICA (C1, C3) PARTITION BY RANGE (C3) (PARTITION '1' <= VALUES < '100', PARTITION OTHERS);`	Create another replica table (REP_TABLE2) which also has an asymmetric partitioning scheme compared to its source table. Only columns 1 and 3 of the source table are replicated and the replica is partitioned on column 3 (C3) by ranges (specifically one partition with all values between 1 and 100, and a partition for all other values.

Related Information

Table Replication Limitations [page 382]

8.11.17.5 Using Table Placement Rules to Manage Replicas

This topic illustrates how to create table placement rules which can be used to create or add replica tables.

You can manage the physical placement of tables, partitions and replicas in a number of ways by defining location values in the table placement rule. Locations are defined in table TABLE PLACEMENT LOCATIONS and then added to the rule in TABLE PLACEMENT (see topics 'Table Placement Rules' and 'Table Placement Locations'). Table placement rules are effective if no location is explicitly defined in the 'AT LOCATION' clause of the SQL statement to create or alter the table. In these cases the database engine determines the location based on the defined table placement rules. In general, two basic rules apply:

- Replicas cannot be created on the same node as the source table.
- It is not possible to create multiple replicas of the same table on a single replica host.

There are three approaches when using table placement rules and examples for each are given in this topic:

- Predefined location values: three predefined locations are available: all (effective by default), coordinator, worker.
- Custom location values which identify nodes by numerical volume ID.
- Table groups which can be used so that all tables of the same group are located together.

The examples given here use the CREATE TABLE ... LIKE syntax to create a replica table. The alternative syntax ALTER TABLE ... ADD ... REPLICA is also supported and is shown in the last example in the 'Predefined location values' section.

Default behavior (location ‘all’)

If no location is specified when a table is created then the table placement value LOCATION ‘all’ is effective and the table is created on one of the available indexserver(s) determined by the system:

```
CREATE TABLE SRC_SCHEMA.TAB1 (A INT);
```

In this case, the location of a replica of this table is also determined by the system:

```
CREATE TABLE REP_SCHEMA.TAB1 LIKE SRC_SCHEMA.TAB1 ASYNCHRONOUS REPLICA;
```

The replica is created on one of the available indexserver(s) other than the node where the source table is located. You can create further replica tables in the same way. These will be created on one of the replica hosts which does not have the same replica table. If there are no more replica hosts where a replica table can be added an error will be returned.

Predefined location values (worker, coordinator)

1) To use the specific predefined location values, firstly, modify the table placement rule by adding schema-specifc locations to the TABLE PLACEMENT table as shown in the following two example statements which create table placement rules to locate all tables/partitions/replicas in schema SRC_SCHEMA on the worker and all tables/partitions/replicas in schema REP_SCHEMA on the coordinator indexserver:

```
ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME => 'SRC_SCHEMA') SET (LOCATION
=> 'worker');
ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME => 'REP_SCHEMA') SET (LOCATION
=> 'coordinator');
```

2) When a new table is created in the SRC_SCHEMA it will be located on one of the available worker indexservers:

```
CREATE TABLE SRC_SCHEMA.TAB1 (A INT);
```

3) A new asynchronous replica of TAB1 created in schema REP_SCHEMA will be created on the coordinator indexserver:

```
CREATE TABLE REP_SCHEMA.TAB1 LIKE SRC_SCHEMA.TAB1 ASYNCHRONOUS REPLICA;
```

> ⓘ Note
>
> There can only be one coordinator node, it is therefore not possible to add more replicas **for the same table** on the coordinator. In the example given here, because REP_SCHEMA.TAB1 is already located at the coordinator, attempting to create replica table TAB2 with the following statement will fail:
>
> ```
> CREATE TABLE REP_SCHEMA.TAB2 LIKE SRC_SCHEMA.TAB1 ASYNCHRONOUS REPLICA;
> ```
>
> The following example statements illustrate this point using the ALTER TABLE <...> ADD REPLICA syntax. A table placement rule is defined to locate tables created in MY_SCHEMA on the coordinator node. If TAB1 already exists it is not then possible to create a replica of TAB1 in the same schema on this same node:

```
ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME => 'MY_SCHEMA') SET (LOCATION
=> 'coordinator');
CREATE TABLE MY_SCHEMA.TAB1 (A INT);
ALTER TABLE MY_SCHEMA.TAB1 ADD ASYNCHRONOUS REPLICA;
```

Custom location values

1) To use custom location values, firstly, modify the table placement rule by adding schema-specific custom locations to the TABLE PLACEMENT LOCATIONS table as shown in the following example statement which adds a location 'repl_volumes' indicating that volume IDs 6, 7, 8 are preferred:

```
ALTER SYSTEM ALTER TABLE PLACEMENT LOCATION repl_volumes SET (INCLUDE=>'6,7,8')
WITH RECONFIGURE;
```

→ Tip

you can also use the EXCLUDE keyword to prevent specific volumes from being used - see examples in topic Table Placement Locations.

2) You can then use this custom location as a LOCATION value in the table placement rule to locate all tables/partitions/replicas in schema REP_SCHEMA on the corresponding indexserver(s) listed in the custom defined location:

```
ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME => 'REP_SCHEMA') SET (LOCATION
=> 'repl_volumes');
```

ⓘ Note

Replicas cannot be created on the same node as the source table - in this example scenario we assume that the coordinator indexserver is located on a different volume such as number 5.

3) A new asynchronous replica of TAB1 created in schema REP_SCHEMA will be created on one of the available indexserver(s) as defined in location 'repl_volumes':

```
CREATE TABLE REP_SCHEMA.TAB1 LIKE SRC_SCHEMA.TAB1 ASYNCHRONOUS REPLICA;
```

Table Groups

The following examples show how group information is handled to co-locate source tables and replicas. Assume the following table placement rules are in place (defined as shown in the previous examples):

- SRC_SCHEMA - location 'coordinator'.
- REP_SCHEMA - location 'worker'.

1) Create a table TAB1 and assign it to a group:

```
CREATE TABLE SRC_SCHEMA.TAB1 (A INT) GROUP NAME MY_SOURCE_GROUP;
```

Table TAB1 in schema SRC_SCHEMA will be created on the coordinator indexserver with the group name 'MY_SOURCE_GROUP'.

2) Create a second table TAB2 in schema SRC_SCHEMA also created on the coordinator indexserver with the same group details:

```
CREATE TABLE SRC_SCHEMA.TAB2 (A INT) GROUP NAME MY_SOURCE_GROUP;
```

3) To use table grouping for replicas a different group name is required: create a replica of TAB1 with the group name 'MY_REPLICA_GROUP':

```
CREATE TABLE REP_SCHEMA.TAB1 GROUP NAME MY_REPLICA_GROUP LIKE SRC_SCHEMA.TAB1
ASYNCHRONOUS REPLICA;
```

The table placement rule determines that REP_SCHEMA.TAB1 is created on a worker node with the specified group name. If this replica is the first table in the group then the HANA engine chooses a location out of the possible worker locations.

4) Create another replica table TAB2 with the same group name:

```
CREATE TABLE REP_SCHEMA.TAB2 GROUP NAME MY_REPLICA_GROUP LIKE SRC_SCHEMA.TAB2
ASYNCHRONOUS REPLICA;
```

The table placement rule determines that the replica is created on a worker node and the same group name automatically co-locates the replica on the same worker node as REP_SCHEMA.TAB1.

> ⓘ Note
>
> When creating a replica if you don't specify a group name, the group name of the source table is ignored and it is not inherited by the replica. This is by design because the replica table cannot be located at the same node as the source. See the following example:
>
> Create a replica table TAB1 in SRC_SCHEMA and assign it to a group:
>
> ```
> CREATE TABLE SRC_SCHEMA.TAB1 (A INT) GROUP NAME MY_REPLICA_GROUP;
> ```
>
> Create a replica in the replica schema without specifying a group name:
>
> ```
> CREATE TABLE REP_SCHEMA.TAB1 LIKE SRC_SCHEMA.TAB1 ASYNCHRONOUS REPLICA;
> ```
>
> The group information of source table SRC_SCHEMA.TAB1 will not be copied to replica REP_SCHEMA.TAB1.

Partitioned Tables

Creating a replica of a partitioned table works in the same way:

```
CREATE TABLE REP_SCHEMA.PART_TAB1 LIKE SRC_SCHEMA.PART_TAB1 ASYNCHRONOUS REPLICA;
```

By default, the replica's partitions will be created like the source partitions' host distribution. Therefore the number of hosts used by the replica is the same as for the source table.

The alternative syntax ALTER TABLE ... ADD ... REPLICA is also supported:

```
ALTER TABLE SRC_SCHEMA.PART_TAB1 ADD ASYNCHRONOUS REPLICA;
```

Related Information

Table Placement Rules [page 373]
Table Placement Locations [page 376]

8.11.17.6 Set an Explicit Table Location

You can set an explicit table location with SQL commands.

These examples show creating replica tables at specified locations for unpartitioned and partitioned tables.

Example

SQL Command	Result
`CREATE TABLE REP_SCHEMA.TAB1 LIKE SRC_SCHEMA.TAB1 ASYNCHRONOUS REPLICA AT 'host1:port1';`	**Unpartitioned tables:** this command creates the first replica table on the specified location 'host1:port1'. Note that replica table cannot be created on master index server and it is not allowed to create multiple replica tables on the same replica node.
`ALTER TABLE SRC_SCHEMA.TAB1 ADD ASYNCHRONOUS REPLICA AT 'host2:port2';` `ALTER TABLE SRC_SCHEMA.TAB1 ADD ASYNCHRONOUS REPLICA AT 'host3:port3';`	Additional replica tables can be created at specified locations.
`CREATE TABLE REP_SCHEMA.PART_TAB1 LIKE SRC_SCHEMA.PART_TAB1 ASYNCHRONOUS REPLICA AT ('host1:port1', 'host2:port2', …);`	**Partitioned tables:** this command creates the first replica table for a partitioned table. The replica partitions will be distributed on the specified nodes.
`ALTER TABLE SRC_SCHEMA.PART_TAB1 ADD ASYNCHRONOUS REPLICA AT ('host3:port3', 'host4:port4', …);`	For the same partitioned table, this command creates additional replica tables on other replica nodes. The replica partitions will be distributed on the specified nodes.

Related Information

Table Placement Locations [page 376]
Data Definition Statements (SQL Reference Guide)

8.11.17.7 Distribution of Table Replicas Using Groups

Table groups can be applied to replica tables and partitions so that you can manage replicas in groups in a HANA scale-out landscape.

Table Groups

Table groups can be applied to replica tables to manage replicas in groups. Grouping can be used with both synchronous and asynchronous replication types (ATR and OSTR) and can also be used with partitioned replica tables.

The following table group attributes can be applied to replicas: group name, group type, subtype and the 'group lead' attribute (see also *Table Classification (Groups)* SYS.TABLE_GROUPS: GROUP_NAME, GROUP_TYPE, SUBTYPE, IS_GROUP_LEAD).

Statements are available to CREATE or ALTER tables, replica tables and partitions and assign them to table groups. Some examples are given here to illustrate the basic features of grouping, full details of the statements and syntax are given in the *SAP HANA Cloud SQL Reference Guide*. You can verify the groups that have been applied by referring to the view SYS.TABLE_PARTITIONS which provides details of group information at partition level.

Examples

1. Examples using CREATE TABLE WITH REPLICA GROUP

This example creates a replica table at a specific location with all three grouping attributes:

```
CREATE COLUMN TABLE "SYSTEM"."T1" (A INT, B INT)
                WITH REPLICA GROUP NAME GROUPB GROUP TYPE type1 GROUP LEAD AT
LOCATION 'MyHost:31340’;
```

To create tables at specific locations the locations must be individually specified (the ALL LOCATIONS keywords, for example are not supported). This example creates a table and two replica tables all at different locations. Replica GROUPD is the lead table:

```
CREATE COLUMN TABLE "SYSTEM"."T1" (A INT, B INT) AT LOCATION 'MyHost:31303'
                WITH REPLICA GROUP NAME GROUPC AT LOCATION 'MyHost:31340'
                WITH REPLICA GROUP NAME GROUPD GROUP LEAD AT LOCATION
'MyHost:31343’;
```

This example creates a table and a replica at different locations with a group for the source table and a group for the replica:

```
CREATE COLUMN TABLE "SYSTEM"."T1" (A INT, B INT) GROUP NAME SRC_GROUP AT
LOCATION 'MyHost:31303'
                WITH REPLICA GROUP NAME REP_GROUP AT LOCATION 'MyHost:31340';
```

2. Examples using ADD REPLICA and ALTER REPLICA

Add a replica table with a group name to an existing table:

```
ALTER TABLE "SYSTEM"."T1" ADD  REPLICA GROUP NAME GROUPA GROUP LEAD AT LOCATION
'MyHost:31340’;
```

Create a group of all existing replica tables at a specified location:

```
ALTER TABLE "SYSTEM"."T1" ALTER REPLICA SET GROUP NAME GROUPD GROUP LEAD AT
LOCATION 'MyHost:31343' ;
```

3. Examples applying groups to replica partitions

To illustrate this functionality firstly create a table with two partitions:

```
CREATE COLUMN TABLE "SYSTEM"."T2" (A INT, B INT) GROUP NAME GRP02
                PARTITION BY RANGE (A) (PARTITION 0 <= values <100, PARTITION
100 <= values < 200) ;
```

You can then assign the partitions to groups. In both of these examples partitions 1 and 2 are assigned to a group:

```
ALTER TABLE "SYSTEM"."T2" ALTER PARTITION (1,2) SET GROUP NAME GROUPZ ;
```

```
ALTER TABLE "SYSTEM"."T2" ALTER REPLICA PARTITION (1, 2) SET GROUP NAME GROUPC;
```

Use the following statement to clear all group information from these partitions; it is not necessary to name individual groups or group types (this would generate an error).

```
ALTER TABLE "SYSTEM"."T2" ALTER PARTITION (1,2) UNSET GROUP;
```

Related Information

Table Classification (Groups) [page 371]

ALTER TABLE Statement (Data Definition)
TABLE_PARTITIONS System View
SAP HANA SQL Reference Guide for SAP HANA Platform

8.11.17.8 Asynchronous Table Replication

Asynchronous table replication can help reduce workload on hosts by balancing load across replica tables on worker hosts in a distributed SAP HANA system.

Asynchronous replication has a number of key characteristics:

- Table replication
  Only a selected list of tables can be set as replicated tables, this is different to system replication which replicates the entire database.
- Asynchronous replication
  Write operations on those replicated tables are propagated to their replica tables asynchronously with almost no impact to the response time of source write transactions. That is, the write transaction is commited without waiting for its propagation to the replica.
- Transactional replication
  Read queries routed to the replica may not see the up-to-date committed result by the nature of asynchronous replication. But, the cross-table transactional consistency is guaranteed by preserving the source transaction boundary and their commit order on log replay at the replica side.
- Parallel log replay
  Although the read queries routed to the replica may see outdated data, the propagation delay is minimized by using parallel log replay at the replica side.

8.11.17.8.1 Configure Asynchronous Table Replication

To set up asynchronous table replication you create a replica schema, create replica tables, and activate replication on the system.

Context

In the steps listed here SRC_SCHEMA, REP_SCHEMA, TAB1 and PART_TAB1 indicate source schema name, replica schema name, normal table name and partitioned table name respectively.

Procedure

1. Create a replica schema

   ```
   CREATE SCHEMA REP_SCHEMA;
   ```

   This creates the replica schema called REP_SCHEMA.
2. Create replica tables

   You can choose the location of your replica tables using *Table Placement Rules* or you can *Set an Explicit Table Location*.

3. Activate replication.

   ```
   ALTER SYSTEM ENABLE ALL ASYNCHRONOUS TABLE REPLICAS;
   ```

   After creating asynchronous replica tables, replication is not activated without this activation command. We recommend first creating all of your necessary replicas and then activating them once.

Results

You have created and activated your table replicas. You can check this in the monitoring view M_TABLE_REPLICAS with, for example, the following command:

```
SELECT * FROM M_TABLE_REPLICAS [WHERE SOURCE_SCHEMA_NAME = SRC_SCHEMA AND
SOURCE_TABLE_NAME = TAB1];
```

This will show all replica tables created for SRC_SCHEMA.TAB1.

Related Information

Using Table Placement Rules to Manage Replicas [page 387]
Set an Explicit Table Location [page 391]

8.11.17.8.2 Operations for Asynchronous Table Replication

There are a number of operations you can perform on replica tables, such as activating and deactivating replication, querying, adding, dropping, and monitoring tables.

Activate and Deactivate Replication

You can activate table replication globally or for a specific named table. Global activation may incur a high-cost job, if your system already has many replica tables or if it is actively replicating. To activate the overall replication operation of all replication tables use:

```
ALTER SYSTEM ENABLE ALL ASYNCHRONOUS TABLE REPLICAS;
```

To activate table replication for a specific table, use the following command, which requires global replication to be turned on:

```
ALTER TABLE SRC_SCHEMA.TAB2 ENABLE ASYNCHRONOUS REPLICA;
```

This example activates the replication operation for SRC_SCHEMA.TAB2. You can use the `disable` parameter instead of enable to deactivate replication. Note that during the table-level replication activation phase, transactional consistency of the target replica table is not guaranteed.

To deactivate the overall replication operation of all replication tables use:

```
ALTER SYSTEM DISABLE ALL ASYNCHRONOUS TABLE REPLICAS;
```

Monitor replication progress with the following query:

```
SELECT * FROM M_TABLE_REPLICAS WHERE REPLICATION_STATUS != 'ENABLED'.
```

Querying Replica Tables

With asynchronous replication it may be necessary to query a specific replica table. To do this replication must be enabled, if it is disabled, all queries on replica tables are automatically re-routed to the source host and tables (this is called status aware routing).

If you submit a simple query to select data from a table which has multiple replica tables then one of the replica tables is automatically selected to service the query. However, it is also possible to use query hints to select a specific replica (identified by volume id) or to use the `result_lag` hint which only selects replica data if it is within an acceptable (specified) lag time. The following examples illustrate these methods:

Automatic Query Distribution

Using the following type of query one of the replica tables will be automatically selected using a round-robin system:

```
SELECT * FROM REP_SCHEMA.TAB1;
```

Explicit Connection Using Hint for Routing

To access one specific replica table you can use the `route_to` hint to make an explicit connection to the location of the replica by including the volume ID number:

```
SELECT * FROM REP_SCHEMA.TAB1 with hint(route_to(4));
```

In this example '4' in the `route_to` hint identifies the volume id of the index server. If the specified volume has the replica table, it is selected to service the query.

You can use the following query to retrieve the volume id of a specific replica:

```
SELECT V.VOLUME_ID, C.SCHEMA_NAME, C.TABLE_NAME, C.PART_ID, C.RECORD_COUNT FROM
M_VOLUMES V, M_CS_TABLES C
WHERE V.HOST = C.HOST and V.PORT = C.PORT AND SCHEMA_NAME = 'REP_SCHEMA' AND
TABLE_NAME LIKE '%TAB1%';
```

This example uses M_CS_TABLES to select a column table. Replace this with M_RS_TABLES to check for row tables.

Connection Using Hint to Avoid Stale Data

You can query replica tables with the `result_lag` hint as shown in the following example.

```
SELECT * FROM SRC_SCHEMA.TAB1 WITH HINT(RESULT_LAG('hana_atr', [seconds]));
```

A preconfigured hint class exists for Asynchronous Table replication called `hint_result_lag_hana_atr`. If the current lag time of the data on the replica is within the acceptable delay period (that is, if the current lag

time is smaller than the stated value for the *[seconds]* parameter), then the query is executed on the replica. Otherwise the query is routed to the source table.

The seconds parameter is optional and if no value is entered on the SQL command line, the default value defined in configuration parameter `atr_default_lag_time` will be applied (configuration details are given in section *Performance: Using Hints to Query Data Snapshots*).

> **Note**
>
> Note that if a query is submitted repeatedly and the staleness of the replica data in relation to the seconds parameter changes between query executions (from acceptable to unacceptable or from unacceptable to acceptable) a recompilation of the query would be triggered. Recompilation is triggered whenever the seconds parameter value of the hint is evaluated and the result causes a switch to a new data source.
>
> It is important therefore in order to minimize the recompilation overhead to set an appropriate value for the seconds parameter in relation to how often the query is submitted and how frequently the data is refreshed.

Drop Replica Tables

The following examples show how to drop replica tables. Note that if a source table is dropped its corresponding replica table is dropped as well.

This example drops REP_SCHEMA and all replica tables in the schema as well:

```
DROP SCHEMA REP_SCHEMA CASCADE;
```

This example drops the replica table for SRC_SCHEMA.TAB1 using AT to name the location:

```
ALTER TABLE SRC_SCHEMA.TAB1 DROP REPLICA AT '<replica host>:<replica port>';
```

Monitoring Replica Tables

Use the system view M_TABLE_REPLICAS to monitor replica tables. This view can be reset and values accumulated since the RESET_TIME can be seen in M_TABLE_REPLICAS_RESET.

The field LAST_ERROR_CODE displays error codes. More detailed information will be described in field LAST_ERROR_MESSAGE. You can look up the meaning of an error code in the system view M_ERROR_CODES. The error codes 2, 4 and 1025 are typically shown during replication and those are categorized as "ERR_GENERAL", "FATAL_OUT_OF_MEMORY" and "ERR_COM" respectively in M_ERROR_CODES.

Related Information

M_TABLE_REPLICAS System View
M_TABLE_REPLICAS_RESET System View

Using Hints to Query Data Snapshots [page 159]

8.11.17.8.3 Row to Column Table Replication

You can replicate data from row store tables to column store replicas, for mixed data types this may give optimal performance.

In a scale-out environment you can replicate data asynchronously from a row store source table to a column store replica table. Row store tables typically provide better performance for transactional (OLTP) workload in comparison to column store tables. Similarly, column store tables offer the best performance for analytics workload. Row to column table replication may therefore be an optimal replication configuration for mixed workload types to get the best performance from both types of table.

Row-to-Column Table Replication Operations

You can configure asynchronous Row-to-Column table replication with the following SQL commands:

- Create a row store source table:

  ```
  CREATE ROW TABLE SRC_SCHEMA.TAB (A INT)
  ```

- Create a column store replica table. For the first replica creation you need to specify the 'COLUMN' keyword:

  ```
  CREATE COLUMN TABLE REP_SCHEMA.TAB LIKE SRC.TBL ASYNCHRONOUS REPLICA AT
  '<host>:<port>'
  ```

  or alter an existing table:

  ```
  ALTER TABLE SRC_SCHEMA.TAB ADD ASYNCHRONOUS REPLICA AT '<host:<port>'
  ```

All other operations are the same as general asynchronous table replication (such as table placement rule, activate replication, deactivate replication and drop replica tables).

Limitations

The following limitations apply:

- Only asynchronous mode is supported for Row-to-Column table replication. Synchronous mode is not supported because the replay performance of the column store does not fully catch up with the throughput of the row store source table.
- Binary object data is not supported, a row store source table which has an LOB type field cannot have a column store replica table.

8.11.17.8.4 Replicate Aging Tables

You can selectively replicate only the hot (current) partitions of aging tables, which means you can have the same benefit of the hot (current) partitions without increasing memory used for cold (old) partitions.

Procedure

1. Create an aging table; there are a number of possible ways to do this.
   a. You can create an aging table with the following CREATE COLUMN TABLE command:

   ```
   CREATE COLUMN TABLE SRC_AGING_SCHEMA.AGING_TABLE (PK INT, TEMPERATURE
   VARCHAR(8) default '00000000', PRIMARY KEY (PK)
   PARTITION BY RANGE(TEMPERATURE)
   (PARTITION VALUE = '00000000' IS CURRENT, PARTITION OTHERS)
   WITH PARTITIONING ON ANY COLUMNS ON
   at 'host:port','host:port'
   ```

   b. Promote a non-partitioned table into a hash-aging table:

   ```
   ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE partition by range (temperature)
   (partition value = '00000000' IS CURRENT, PARTITION OTHERS) WITH
   PARTITIONING ON ANY COLUMNS ON
   ```

   c. Promote a hash-partitioned table into an aging table

   ```
   ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE PARTITION BY
   HASH(A) PARTITIONS 2,
   RANGE(TEMPERATURE) (PARTITION VALUE = '00000000' IS CURRENT)
   WITH PARTITIONING ON ANY COLUMNS ON
   FOR DEFAULT STORAGE NON CURRENT PARTITIONS PAGE LOADABLE
   FOR NON CURRENT PARTITIONS UNIQUE CONSTRAINTS OFF;
   ```

   d. If an Aging table exists, use ADD PARTITION to create further cold partitions, for example:

   ```
   ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE ADD PARTITION 20000101 <= VALUES
   < 20020101
   ```

   Partitioning parameter values:

   - Use RANGE for Range partitioning
   - TIME SELECTION is the internal name for this aging implementation
   - PAGED ATTRIBUTES is an optional property that may be specified in order to use Paged Attributes for cold partitions
   - NO UNIQUE CHECK is an optional property that disables the unique check on cold partitions
   - TEMPERATURE is the VARCHAR(8) temperature column
   - 00000000 is the identifier for the hot partition
   - <ranges> shall be substituted with actual dates. For example, specify '20130101-20140101, 20140101-20150101'

2. Enable Actual Only Replication.

   The configuration parameter `async_rep_actual_only_replication_enabled` in the indexserver.ini file must be enabled; it is enabled by default.

3. Create a replica schema.

   ```
   CREATE SCHEMA REP_SCHEMA
   ```

4. Activate replication.

   ```
   ALTER SYSTEM ENABLE ALL ASYNCHRONOUS TABLE REPLICAS;
   ```

   This statement will activate all the other replicas except actual-only replication (actual-only replicas will be created in the next step). The actual-only replication should be enabled separately (in step 6) after all the other replicas are already enabled here.

5. Create replica tables.

   a. Create replica table with Table Placement rule.

      The following commands create table placement rules for a replica schema:

      ```
      ALTER SYSTEM ALTER configuration ('global.ini','SYSTEM') SET
      ('table_placement','repl_volumes')='6,7,8' WITH RECONFIGURE;
      ```

      Here, *repl_volumes* is an alias name used to apply the table placement rule. '6,7,8' means multiple nodes are used as replica hosts and each number indicates the volume ID.

      ```
      ALTER SYSTEM ALTER TABLE PLACEMENT (SCHEMA_NAME => 'REP_SCHEMA') SET
      (LOCATION => 'repl_volumes');
      ```

      With this table placement rule, any of tables in REP_SCHEMA will be created at the locations mapped in “repl_volumes”.

      > ### Note
      >
      > Replica tables cannot be created on master indexserver and it is not allowed to create multiple replica tables on the same replica node.

      To create the first replica table without assigning its location use the following SQL statement. The replica table will be created on one of replica nodes.

      ```
      CREATE TABLE REP_AGING_SCHEMA.AGING_TABLE LIKE
      SRC_AGING_SCHEMA.AGING_TABLE ASYNCHRONOUS REPLICA;
      ```

      If you want to create more than one replica for SRC_SCHEMA, use:

      ```
      ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE ADD ASYNCHRONOUS REPLICA;
      ```

      This creates the second replica table on other replica nodes without assigning its location. The second replica table will be created on one of replica nodes which does not have the same replica table. If there are no more replica nodes to add a replica table to, an error will be returned. You can create the third, the fourth, and more replica tables in the same manner.

   b. Create replica table with an explicit table location.

      To create the first replica table on the specified location 'host:port'.

      ```
      CREATE TABLE REP_AGING_SCHEMA.AGING_TABLE LIKE
      SRC_AGING_SCHEMA.AGING_TABLE ASYNCHRONOUS REPLICA AT 'host:port';
      ```

> **Note**
>
> Replica tables cannot be created on master indexserver and it is not allowed to create multiple replica tables on the same replica node.

To create additional replica tables on the specified location. You can create partitioned tables on more than one host using a comma separated list enclosed in parentheses as in the following examples:

```
CREATE TABLE REP_AGING_SCHEMA.AGING_TABLE LIKE
SRC_AGING_SCHEMA.AGING_TABLE ASYNCHRONOUS REPLICA AT ('host1:port1',
'host2:port2', ...);
```

```
ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE ADD ASYNCHRONOUS REPLICA AT
('host1:port1', 'host2:port2', ...);
```

6. Turn a partition on or off.

```
ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE [ENABLE/DISABLE] ASYNCHRONOUS
REPLICA PARTITION [logical partition id];
```

Only hot partitions can be turned on/off. If only the hot partition is enabled, the others are not replicated.

7. Check Replica Tables

To view all replica tables created for SRC_AGING_SCHEMA.AGING_TABLE use:

```
SELECT * FROM M_ASYNCHRONOUS_TABLE_REPLICAS WHERE SOURCE_SCHEMA_NAME =
'SRC_SCHEMA' AND SOURCE_TABLE_NAME = 'AGING_TABLE';
```

8. Query on aging tables with a hint

You can read hot or cold data from an aging table using the following SQL suffix. The RANGE_RESTRICTION is a filter for the range partitioning.

```
WITH RANGE_RESTRICTION('CURRENT') or WITH RANGE_RESTRICTION('DATE');
```

Use DATE in the format “yyyy-mm-dd”. If you specify a date, it will always consider the hot partition as well. CURRENT is the hot partition.

```
SELECT * FROM SRC_AGING_SCHEMA.AGING_TABLE WITH RANGE_RESTRICTION('CURRENT')
SELECT * FROM REP_AGING_SCHEMA.AGING_TABLE WITH RANGE_RESTRICTION('CURRENT')
SELECT * FROM SRC_AGING_SCHEMA.AGING_TABLE WITH
RANGE_RESTRICTION('2000-01-01')
SELECT * FROM REP_AGING_SCHEMA.AGING_TABLE WITH
RANGE_RESTRICTION('2000-01-01')
```

9. Deactivate Replication

```
ALTER SYSTEM DISABLE ALL ASYNCHRONOUS TABLE REPLICAS;
```

This command deactivates the overall replication operation of all replication tables.

You can monitor its progress using:

```
SELECT * FROM M_TABLE_REPLICAS WHERE REPLICATION_STATUS != 'ENABLED';
```

You can turn off a specific table only using:

```
ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE DISABLE ASYNCHRONOUS REPLICA;
```

10. Drop Replica Tables

> **Note**
>
> If a source table is dropped, its corresponding replica table is dropped as well.

```
DROP SCHEMA REP_SCHEMA CASCADE;
```

Drops REP_SCHEMA schema and all replica tables in the schema as well.

```
ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE DROP REPLICA AT ALL LOCATIONS;
```

Drops all replica tables of the specified source table SRC_AGING_SCHEMA.AGING_TABLE.

```
ALTER TABLE SRC_AGING_SCHEMA.AGING_TABLE DROP REPLICA AT 'host:port' ;
```

Drops the replica located at '<replica host>:<replica port>'.

8.11.17.8.5 Query Aging Tables

An actual partition on a replica is only able to be accessed by using the CURRENT range restriction on a replica table. Otherwise, all queries are routed to a source table even though the queries are on a replica table.

Procedure

1. Access to actual partition(hot data) on replica

   You can get hot data from an actual partition by using the CURRENT range restriction on a replica table.

   ```
   SELECT * FROM REP_AGING_SCHEMA.AGING_TABLE WITH RANGE_RESTRICTION('CURRENT')
   ```

2. Access to both actual(hot data) and history partition(cold data) on replica

   You can get hot and cold data from both actual and history partitions by using the DATE range restriction on a replica table. The query is routed to a source table. Even though the DATE range restriction indicates only hot data, the query is routed to a source table.

   ```
   SELECT * FROM REP_AGING_SCHEMA.AGING_TABLE WITH RANGE_RESTRICTION('yyyy-mm-dd')
   ```

3. Access to replica without RANGE RESTRICTION

   You can get data from all actual and historical partitions. The query is routed to a source table.

   ```
   SELECT * FROM REP_AGING_SCHEMA.AGING_TABLE
   ```

8.11.17.9 Synchronous Table Replication

With synchronous table replication the source and replica are always synchronized, this is therefore a more transparent solution.

Because the source and replica are always synchronized, synchronous table replication does not require any SQL or application changes. Activation of the replica is also not required as table replication is activated at commit time.

> ⓘ Note
>
> In SAP HANA 2.0 a new version of synchronous table replication has been implemented Optimistic Synchronous Table Replication (OSTR) and replica tables which were created in earlier versions may no longer be valid. In this case an error message *Feature not supported* will be generated. If this error occurs the replica tables must be dropped and recreated using the ALTER TABLE commands for DROP and ADD:

```
ALTER TABLE {SCHEMA_NAME}.{TABLE_NAME} DROP REPLICA AT 'replica_location'
```

```
ALTER TABLE {SCHEMA_NAME}.{TABLE_NAME} ADD [SYNCHRONOUS] REPLICA AT
'replica_location'
```

Related Information

Configure Synchronous Table Replication [page 403]
Operations for Asynchronous Table Replication [page 395]

8.11.17.9.1 Configure Synchronous Table Replication

You can configure synchronous replication using the SQL editor by adding replica tables.

Context

In the example commands below, the following placeholders are used:

SRC_SCHEMA *	table schema name	REP_SCHEMA *	replica table schema name
SRC_TABLE *	table name	REP_TABLE	replica table name
SRC_PART_TABLE *	partitioned table name	REP_PART_TABLE	replica partitioned table name

*We assume that SRC_SCHEMA, REP_SCHEMA and SRC_TABLE and SRC_PART_TABLE already exist in your system.

Procedure

1. Use the following commands to create a table and add replica tables at the specified location 'host:port'.

Non-partitioned Tables	Partitioned Tables
Note that, for non-partitioned tables, it is not allowed to create multiple replica tables on the same replica node.	Note that, for partitioned tables, replica partitions cannot be distributed on separate nodes. They should be located on the same replica node.
Explicit synchronous table creation `CREATE COLUMN TABLE REP_SCHEMA.REP_TABLE LIKE SRC_SCHEMA.SRC_TABLE SYNCHRONOUS REPLICA AT 'host:port'`	Explicit synchronous table creation `CREATE COLUMN TABLE REP_SCHEMA.REP_PART_TABLE LIKE SRC_SCHEMA.SRC_PART_TABLE SYNCHRONOUS REPLICA AT 'host:port'`
Implicit synchronous table creation `ALTER TABLE SRC_SCHEMA.SRC_TABLE ADD SYNCHRONOUS REPLICA AT 'host:port'`	Implicit synchronous table creation `ALTER TABLE SRC_SCHEMA.SRC_PART_TABLE ADD SYNCHRONOUS REPLICA AT 'host:port';`

2. As synchronous replicas are automatically activated, there is no need to enable them. If necessary you can activate and deactivate table replication as described in the following topic.
3. Check your replica tables by querying the M_TABLE_REPLICAS view.

   ```
   SELECT * FROM M_TABLE_REPLICAS WHERE SOURCE_TABLE_NAME = 'SRC_TABLE'
   ```

   This view can be reset and values accumulated since the RESET_TIME can be seen in M_TABLE_REPLICAS_RESET.

Related Information

Table Replication Limitations [page 382]

8.11.17.9.2 Operations for Synchronous Replication

There are a number of operations you can perform for synchronous replication such as activating or deactivating replication, and dropping replica tables.

Using synchronous replication, you do not have to modify queries to access replica tables. When you access a source table, either the source table or the replica table is automatically selected and the query is routed to the selected table in a round robin manner. So, just by adding replica tables, the read-only query workload can be load-balanced without any SQL string or application change.

Activate and Deactivate Replication

The following command deactivates all synchronous replication:

```
ALTER SYSTEM DISABLE ALL SYNCHRONOUS TABLE REPLICAS
```

The following command deactivates a specific synchronous table replica:

```
ALTER TABLE SRC_SCHEMA.SRC_TABLE DISABLE SYNCHRONOUS REPLICA
```

The following command activates all synchronous replication:

```
ALTER SYSTEM ENABLE ALL SYNCHRONOUS TABLE REPLICAS
```

The following command activates a specific synchronous table replica:

```
ALTER TABLE SRC_SCHEMA.SRC_TABLE ENABLE SYNCHRONOUS REPLICA
```

You can check a table's current replication status; this is shown in M_TABLE_REPLICAS.

Drop Replica Tables

The following command drops a replica table at a specific location. Note that if a source table is dropped, its corresponding replica tables are all dropped as well.

```
SRC_TABLE'. ALTER TABLE SRC_SCHEMA.SRC_TABLE DROP REPLICA AT '<replica
host>:<replica port>
```

8.11.18 Redistributing Tables in a Scaleout SAP HANA System

In a scaleout SAP HANA system, tables and table partitions are assigned to an index server on a particular host when they are created. As the system evolves over time you may need to optimize the location of tables and partitions by running automatic table redistribution.

There are several occasions when tables or partitions of tables need to be moved to other servers, for example, the tables and partitions which grow fastest in size may need to be split and redistributed. Table redistribution aims to balance the workload across all hosts and optimize the location of tables and partitions so that tables which are often used together are located on the same node. Table redistribution is based on the table placement rules, these determine, for example, table sizes, partitioning threshold values, and preferred partition locations.

Although it is possible to move tables and table partitions manually from one host to another, this is not practical for large-scale redistribution of data. The table redistribution function offers a range of options to perform balancing, optimization and housekeeping tasks. Redistribution is a two stage process: the first stage is to generate a redistribution plan, this can be done iteratively and the distribution configuration can be modified and tweaked until the desired result is achieved; secondly the plan is executed.

You can run table redistribution from the SAP HANA administration tools, studio and cockpit, or from the SQL command line.

Related Information

Table Distribution and Partitioning
Table Partitioning and Redistribution in SAP HANA Studio
Table Redistribution Commands [page 407]
Table Placement [page 370]

8.11.18.1 Table Distribution in SAP HANA Cockpit

Use the SAP HANA cockpit to manage table distribution. You can view and save the current table distribution, automatically generate an optimized table redistribution, re-run a previously executed plan, or restore a saved plan.

Many features related to Table Distribution are available within SAP HANA cockpit, including:

- Table placement rules
- Generating and executing table redistribution plans
- Modifying table distributions
- Table group advisor

All information related to these features is given in the *SAP HANA Cockpit Administration Guide*. Use the links on this page to go directly to the latest version of the cockpit documentation.

Related Information

Table Distribution in SAP HANA cockpit
Table Placement Rules in SAP HANA cockpit
Generating and executing table redistribution plans in SAP HANA cockpit
Mofifying table distributions in SAP HANA cockpit
Table group advisor in SAP HANA cockpit

8.11.18.2 Table Redistribution in SAP HANA Studio

Administrators can use the table redistribution feature in the SAP HANA studio to create a plan for redistributing and repartitioning tables. The administrator can review the plan and execute it.

SAP HANA supports several redistribution operations that use complex algorithms as well as configurable table placement rules and redistribution parameters to evaluate the current distribution and determine a better distribution depending on the situation.

Redistribution operations are available to support the following situations:

- You are planning to remove a host from your system
- You have added a new host to your system
- You want to optimize current table distribution
- You want to optimize table partitioning

Refer to the *SAP HANA Studio Guide* for details of the features available.

Data Distribution Optimizer

To plan, adjust and analyze landscape redistribution, you can also use the Data Distribution Optimizer. The Data Distribution Optimizer is an SAP HANA XS-based tool included in the SAP HANA Data Warehousing Foundation option. The Data Distribution Optimizer provides packaged tools for large scale SAP HANA use cases to support more efficient data management and distribution in an SAP HANA landscape. For more information, see the *SAP HANA Data Warehousing Foundation - Data Distribution Optimizer Administration Guide*.

Related Information

Managing Table Redistribution (SAP HANA Studio)
SAP HANA Data Warehousing Foundation - Data Distribution Optimizer Administration Guide
Table Partitioning [page 317]
Table Replication [page 380]
Table Placement [page 370]

8.11.18.3 Table Redistribution Commands

As an alternative to using Table Redistribution in SAP HANA Cockpit you can also run it from the command line; this approach offers additional functionality including the option to modify at run time some of the configuration parameters which control redistribution.

Table redistribution is based on the table placement rules defined in the table TABLE_PLACEMENT, these determine, for example, table sizes, partitioning threshold values and preferred partition locations. Tables

which have been grouped using the properties GROUP, GROUP NAME and SUBTYPE are also distributed to maximize colocation so that partitions with common properties are located together. Note that location preferences take precedence over grouping properties.

Redistribution is a two stage process: firstly to generate the plan and secondly to execute the plan; separate commands are used for each stage:

1. The plan generation command is a multi-purpose tool which requires an algorithm number as a parameter to determine which actions are executed. Depending on the algorithm selected, additional optional parameter values may also be available to give more control over the execution.
2. The plan execution command returns a numeric plan id value. You can also retrieve this value (REORG_ID) from the REORG_OVERVIEW system view - see System Views below.

The syntax for these commands is:

- `CALL REORG_GENERATE(<algorithm integer>, <optional parameter string>);`
- `CALL REORG_EXECUTE(?)`

Optimized Processing

Tables must inevitably be locked for some time during table redistribution. To minimize the locking time and move tables as efficiently as possible an 'online' processing mode is used based on asynchronous replication. This approach is applied wherever possible and can be used for non-partitioned tables (replicated tables with a global index cannot be moved in this way). Table grouping is a key aspect of redistribution and to optimize performance tables are moved as far as possible in parallel in groups.

Partitioning of Large Tables

The REORG_GENERATE command includes a check on table sizes and large unpartitioned tables which are close to the maximum size limit of 2 billion records (or a lower value which can be set in configuration) will be identified as candidates for partitioning. A check is made to identify primary key columns which can be used for hash partitioning. If no suitable columns are found then any of the first five columns in the table are used as the basis for hash partitioning. If no suitable columns are found for hash partitioning then round-robin partitioning is used instead.

This feature can be enabled by setting the configuration parameter `enable_partition_spec_creator` to TRUE. Users should bear in mind that this feature can have a negative impact on performance as tables must be locked during the process.

The maximum table size at which partitioning takes place is defined in the parameter `max_rows_per_partition` in the `table_placement` section of the global.ini file (the default value is 2 billion).

Pinning Tables

To prevent tables from being moved during distribution you can pin them to the current host. You can do this using the NOT MOVABLE clause of the CREATE/ALTER table command. Refer to the*SAP HANA SQL and System Views Reference* for full details.

Native Storage Extension

You can also include NSE tables in the redistribution - this option is also available as an optimization goal in SAP HANA cockpit. To use this feature the `balance_by_table_reads` configuration parameter in the `table_redist` section of the indexserver.ini file must be enabled. When this is enabled KPI data for NSE tables and partitions is collected, analyzed, and included in the table redistribution plan. If you choose to execute the plan then the redistribution actions for NSE tables are applied across the server landscape.

Required Privileges

RESOURCE ADMIN and CATALOG READ privileges are required to call REORG_GENERATE(). The command only operates on tables and partitions which the executing user is allowed to see as catalog objects.

Generating the Plan: Algorithms and Options

The following table gives an overview of the most commonly-required algorithms and a summary of the options available for each one - see examples and details of the options which follow.

Table Redistribution Algorithms and Options

Num	Algorithm Name	Description
6	Balance landscape	This function checks if tables in the landscape are placed on invalid servers according to the table placement rules, and checks if a split or merge is necessary in order to achieve optimal positions for the partitions and tables and to evenly distribute tables across the indexserver hosts. **Options**: TOP | USE_GROUP_ADVISOR | SCHEMA_NAME | TABLE_NAME | GROUP_NAME | GROUP_TYPE | GROUP_SUBTYPE | RECALC | NO_PLAN | NO_SPLIT | SCOPE
1	Add server	Run this check after adding one or more index servers to the landscape. If new partitions can be created a plan will be generated to split the tables and move the new partitions to the newly added indexservers. **Options**: TOP | USE_GROUP_ADVISOR | SCHEMA_NAME | TABLE_NAME | GROUP_NAME | GROUP_TYPE | GROUP_SUBTYPE | RECALC | NO_PLAN
2	Clear server	Moves all partitions from a named server to other servers in the landscape. **Options**: USE_GROUP_ADVISOR
4	Save	Save current landscape setup. No optional parameter.
5	Restore	Restore a saved landscape setup. Enter the plan ID value as the optional parameter value.
7	Check number of partitions	This function checks if partitioned tables need to be repartitioned and creates a plan to split tables if the partitions exceed a configured row count threshold. No optional parameter.
12	Execute Group Advisor	Calls the Group Advisor and creates an executable plan from its output. See Group Advisor in the following topic.
14	Check table placement	Check current landscape against table placement rules and (if necessary) provide a plan to move tables and partitions to the correct hosts. **Additional Options**: LEAVE_UNCHANGED_UNTOUCHED | KEEP_VALID | NO_SPLIT
15	Rerun plan	Rerun failed items from previously executed plans. **Option**: RERUN_ALL

Num	Algorithm Name	Description
16	Housekeeping	Perform housekeeping tasks. Additional privileges may be required for specific actions. **Housekeeping Options**: OPTIMIZE_COMPRESSION | DEFRAG | LOAD_TABLE | MERGE_DELTA | ALL

Optional Parameters

The following table gives more details of the optional parameters which are available.

Option	Type	Detail
TOP	String	Restrict redistribution to the top n items. Example: `CALL REORG_GENERATE(6,'TOP=>4')` executes algorithm #6 and optimizes only the top four table groups.
USE_GROUP_ADVISOR	String	Calls the Group Advisor and creates an executable plan from its output.
SCHEMA_NAME	String	Restrict redistribution to the named schema(s) - comma-separated list.
TABLE_NAME	String	Restrict redistribution to the named table(s) - comma-separated list.
GROUP_NAME	String	Restrict redistribution to the named group(s) - comma-separated list.
GROUP_TYPE	String	Restrict redistribution to the named group types(s) - comma-separated list.
GROUP_SUBTYPE	String	Restrict redistribution to the named sub types(s) - comma-separated list.
RECALC	True / False	If true then recalculate the landscape data of the last REORG_GENERATE run. This option works only if REORG_GENERATE has been called before within the same connection session. This parameter can be used to speed up plan generation with different parameters.
NO_PLAN	True / False	If true then the planning stage of generating the plan is skipped. This can be used with external tools when landscape data needs to be collected and a distribution must be calculated but might be modified.
SCOPE	Keyword	Use one or more of the following values (see example which follows) to restrict the scope of the redistribution to include only the named items specified by these keywords. The default value is 'ALL' so that all tables visible to the user are included in the redistribution. LOADED — Tables which are loaded or partially loaded UNLOADED — Tables which are not loaded FILLED — Tables with a record count greater than 10 EMPTY — Tables with a record count less than or equal to 10 USED — Tables with a total execution count greater than 10

Option	Type	Detail	
		UNUSED	Tables with a total execution count of less than or equal to 10
		LOB	Tables with LOB columns
		NOLOB	Tables without LOB columns

Examples

Add server (algorithm 1)

With this algorithm you can use the optional filter parameters to, for example, restrict redistribution to specified schemas, tables, table groups and so on. The following example uses the SCHEMA_NAME option to generate a plan for all tables in schema SAPBWP.

```
CALL REORG_GENERATE(1, 'SCHEMA_NAME => SAPBWP')
```

Balance Landscape / Table (algorithm 6)

The following examples show the usage of optional parameters with this balancing algorithm:

If the options parameter string is left blank a plan is generated for all visible tables:

```
CALL REORG_GENERATE(6,'');
```

This example uses the GROUP_NAME option to generate a plan for all tables in three specified groups:

```
CALL REORG_GENERATE(6,'GROUP_NAME=>TABLEGROUP1, TABLEGROUP2, TABLEGROUP3');
```

This example uses the SCHEMA_NAME option to generate a plan for all tables in schema SAPBWP:

```
CALL REORG_GENERATE(6,'SCHEMA_NAME => SAPBWP');
```

This example show usage of the SCOPE option. The plan is restricted to only tables with a record count greater than 10 and which have no LOB columns.

```
CALL REORG_GENERATE(6, 'SCOPE=>FILLED,NOLOB');
```

System Views

The following system views show details of table redistribution. The last two views in this list show information about the most recent distribution operation; the details are deleted when the current connection to the database is closed.

- REORG_OVERVIEW Provides an overview of landscape redistributions.
- REORG_STEPS Shows details of the individual steps (items) of each plan.
- REORG_PLAN This view contains details of the last table redistribution plan generated with this database connection.
- REORG_PLAN_INFOS Showing details (as key-value pairs) of the last executed redistribution (algorithm value and parameters used).

Related Information

Table Placement Rules [page 373]
Generate and Execute a Table Redistribution Plan (SAP HANA Cockpit)
Table Partitioning and Redistribution in SAP HANA Studio
SAP HANA SQL Reference Guide for SAP HANA Platform
Configuration of Table Redistribution [page 412]

8.11.18.3.1 Group Advisor

The Group Advisor tool can be executed from within table redistribution to make recommendation about creating table groups.

Table grouping identifies tables which are often used together so that during redistribution they can be located together on the same node in order to avoid cross-node communication in the landscape. The group advisor tool can be integrated into table redistribution to execute a preparation step which creates groups automatically. The tool determines which tables are often used together by analyzing the current statement cache to find relationships between tables and it then (internally) makes recommendations about which tables should be located together. Table redistribution evaluates these recommendations before generating the plan.

When the plan is executed the table grouping information is persisted in the table SYS.TABLE_GROUP by setting a group name and setting the group type 'sap.join'. Note that existing table groups which have been defined for other applications (Business Warehouse, for example) are not modified in any way by the Group Advisor.

From the command line you can execute Group Advisor in the REORG_GENERATE command (algorithm 12) which then creates an executable plan from its output. Group Advisor is configurable by submitting configuration values in JSON format either directly on the command line or by writing the JSON code to a configuration parameter. To submit JSON code as a configuration parameter use `group_advisor_parameter` in the `table_redist` section of the indexserver.ini file.

8.11.18.3.2 Configuration of Table Redistribution

The operation of table redistribution is managed by configuration parameters; some of these can be reconfigured for the current session at run-time.

Configuration

The precise operation of table redistribution is managed by sets of configuration parameters located in the service configuration file (typically `indexserver.ini`) section `table_redist`. Firstly, there are parameters controlling common behavior of table redistribution and secondly a set of session parameters for fine tuning redistribution by applying weighting values.

These parameters can be defined by the administrator for the system generally, but for some values the system settings can be overridden by values submitted on the command line. This is done by entering the key-value pair as a parameter value as shown in the following example. These settings can be combined with other parameters in a comma-separated string.

```
CALL REORG_GENERATE(6, 'BALANCE_BY_EXECUTION_COUNT=>True');
```

Settings passed in this way are only valid for the current user session.

The following tables list the configuration parameters which can be set:

Parameter	Data Type
ALL_MOVES_PHYSICAL	BOOLEAN
ASYMETRIC_CORRECTION_BY	INTEGER
BALANCE_BY_TABLE_READS	BOOLEAN
DEBUG_BALANCE_DETAIL	BOOLEAN
DEBUG_EXPORT_DETAIL	BOOLEAN
DEBUG_SCORECARD_DETAIL	BOOLEAN
ENABLE_CONSISTENCY_CHECK	BOOLEAN
ENABLE_CONTENT_TABLE_REDIST	BOOLEAN
ENABLE_ENSURE_MOVES	BOOLEAN
ENABLE_MERGE	BOOLEAN
ENABLE_MULTI_STORE_TABLES	BOOLEAN
ENABLE_OPTIMIZE_COMPRESSION	BOOLEAN
ENABLE_RELOAD_TABLES	BOOLEAN
ENABLE_REPARTITIONING_WITH_GCD	BOOLEAN
ENABLE_ROW_STORE_TABLES	BOOLEAN
ENABLE_SYS_TABLE_REDIST	BOOLEAN
EXPORT_DATA	BOOLEAN
FORCE_PARTNUM_TO_SPLITRULE	BOOLEAN
MAX_PARTITIONS	INTEGER
MAX_PARTITIONS_LIMITED_BY_LOCATIONS	BOOLEAN
MAX_PLAN_ITERATIONS	INTEGER

Parameter	Data Type
MAX_ROWS_PER_PARTITION	INTEGER
MEM_LOAD_TRESHOLD	INTEGER
MIN_PLAN_ITERATIONS	INTEGER
MOVE_ROW_STORE	STRING
USE_GROUPS_FOR_DEPENDENCY	BOOLEAN
WORK_SEQUENCE	STRING
WORK_SEQUENCE_ASC	BOOLEAN
WORK_SEQUENCE_SORT	STRING

The following parameters (can also be set from the command line) control the optimization process. They are in pairs: firstly a boolean parameter and a corresponding numeric 'weight' parameter with the suffix '_WEIGHT', for example:

- BALANCE_BY_PARTNUM - boolean parameter: set to True to activate this pair
- BALANCE_BY_PARTNUM_WEIGHT - numeric value used as a factor (default = 1)

The parameters can be activated by setting the first one to `true` and then setting a numeric value for the weight parameter, this operates as a multiplication factor to calculate a priority value for this aspect of redistribution.

Parameter	Optimize by...
BALANCE_BY_PARTNUM	- by the number of partitions placed on the indexserver
BALANCE_BY_MEMUSE	- by memory usage based on size of table / partition
BALANCE_BY_TABLE_SIZE_HOSTED	
BALANCE_BY_ROWS	- by the number of rows of the tables/partitions
BALANCE_BY_WORKLOAD	
BALANCE_BY_EXECUTION_COUNT	- by an external provided execution count
BALANCE_BY_EXECUTION_TIME	- by an external provided execution time
BALANCE_BY_RANDOMIZER	
BALANCE_BY_TABLE_CLASSIFICATION	

8.12 Workload Management

The load on an SAP HANA system can be managed by selectively applying limitations and priorities to how resources (such as the CPU, the number of active threads and memory) are used. Settings can be applied globally or at the level of individual user sessions by using workload classes.

On an SAP HANA system there are many different types of workload due to the capabilities of the platform, from simple or complex statements to potentially long-running data loading jobs. These workload types must be balanced with the system resources that are available to handle concurrent work. For simplicity we classify workload queries as transactional (OLTP) or analytic (OLAP). With a transactional query the typical response time is measured in milliseconds and these queries tend to be executed in a single thread. Analytic queries on the other hand tend to feature more complex operations using multiple threads during execution, this can lead to higher CPU usage and memory consumption compared with transactional queries.

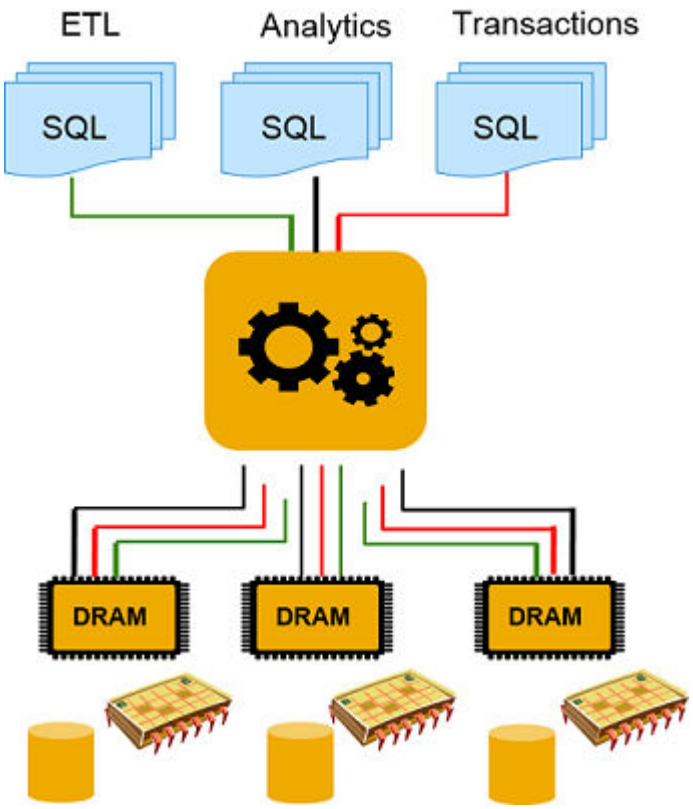
To manage the workload of your system aim to ensure that the database management system is running in an optimal way given the available resources. The goal is to maximize the overall system performance by balancing the demand for resources between the various workloads, not just to optimize for one particular type of operation. If you achieve this then requests will be carried out in a way that meets your performance expectations and you will be able to adapt to changing workloads over time. Besides optimizing for performance you can also optimize for robustness so that statement response times are more predictable.

8.12.1 Workload in the Context of SAP HANA

Workload in the context of SAP HANA can be described as a set of requests with common characteristics.

We can look at the details of a particular workload in a number of ways. We can look at the source of requests and determine if particular applications or application users generate a high workload for the system. We can examine what kinds of SQL statements are generated: are they simple or complex? Is there a prioritization of work done based on business importance, for example, does one part of the business need to have more access at peak times? We can also look at what kind of service level objectives the business has in terms of response times and throughput.

The following figure shows different types of workload such as Extract Transform and Load operations (used in data warehouses to load new data in batches from source system) as well as analytic and transactional operations:



Types of Workload

When we discuss workload management we are really talking about stressing the system in terms of its resource utilization. The main resources we look at (shown in the above illustration) are CPU, memory, disk I/O, and network. In the context of SAP HANA, disk I/O comes into play for logging, for example, in an OLTP scenario many small transactions result in a high level of logging compared to analytic workloads (although SAP HANA tries to minimize this). With SAP HANA, network connections between nodes in a scale out system can be optimized as well, for example, statement routing is used to minimize network overhead as much as possible.

However, when we try to influence workload in a system, the main focus is on the available CPUs and memory being allocated and utilized. Mixed transactional and analytic workloads can, for example, compete for resources and at times require more resources than are readily available. If one request dominates there may be a queuing effect, meaning the next request may have to wait until the previous one is ready. Such situations need to be managed to minimize the impact on overall performance.

Related Information

Persistent Data Storage in the SAP HANA Database [page 180]
Scaling SAP HANA [page 1079]

8.12.1.1 Options for Managing Workload

Workload management can be configured at multiple levels: at the operating system-level, by using global initialization settings, and at the session level.

There are a number of things you can do to influence how workload is handled:

- Outside the SAP HANA system on the operating system level you can set the affinity of the available cores.
- You can apply static settings using parameters to configure execution, memory management and peak load situations.
- You can influence workload dynamically at system runtime by defining workload classes.

All of these options have default settings which are applied during the HANA installation. These general-purpose settings may provide you with perfectly acceptable performance in which case the workload management features described in this chapter may not be necessary. Before you begin with workload management, you should ensure that the system generally is well configured: that SQL statements are tuned, that in a distributed environment tables are optimally distributed, and that indexes have been defined as needed.

If you have specific workload management requirements the following table outlines a process of looking at ever more fine-grained controls that can be applied with regard to CPU, memory and execution priority.

Options for Controlling Workload Management

Area	Possible Actions
CPU Configure CPU at Operating System level	Settings related to *affinity* are available to bind server processes to specific CPU cores. Processes must be restarted before these changes become effective. For more information, see *Controlling CPU Consumption*.
CPU Thread Pools Configure CPU at HANA System level	Global *execution* settings are available to manage CPU thread pools and manage parallel execution (concurrency). For more information, see *Controlling Parallel Execution of SQL Statements*.
Memory Apply settings for memory management	Global *memorymanager* settings are available to apply limits to the resources allocated to expensive SQL statements. For more information, see *Setting a Memory Limit for SQL Statements*.
Admission Control Configuration options for peak load situations	Global *admission control* settings can be used to apply system capacity thresholds above which SQL statements can be either rejected or queued. For more information, see *Managing Peak Load (Admission Control)*.

Area	Possible Actions
Priority and Dynamic Workload Class Mapping Manage workload and workload priority using classes	A more targeted approach to workload management is possible by setting up pre-configured classes which can be mapped to individual user sessions. You can, for example, map an application name or an application user to a specific workload class. Classes include the option to apply a workload priority value. You can set up classes using: • SAP HANA cockpit • SQL commands For more information, see *Managing Workload with Workload Classes*.

At the end of this section is a set of scenarios giving details of different hardware configurations and different usage situations. For each scenario, suggestions are made about appropriate workload management options which could be used.

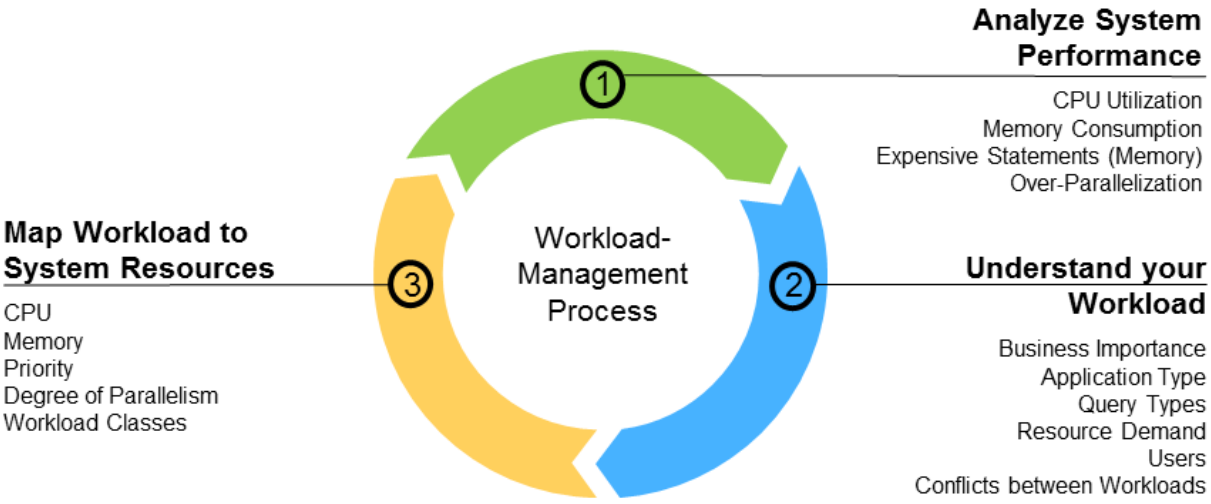
Related Information

Controlling CPU Consumption [page 421]
Controlling Parallel Execution of SQL Statements [page 428]
Setting a Memory Limit for SQL Statements [page 431]
Managing Peak Load (Admission Control) [page 434]
Managing Workload with Workload Classes [page 439]
Example Workload Management Scenarios [page 457]

8.12.1.2 Understand your Workload

Managing workload can be seen as an iterative three-part process: Analyze the current system performance, understand the nature of your workload and map your workload to the system resources.

There is no one single workload management configuration that fits all scenarios. Because workload management settings are highly workload dependent you must first understand your workload. The following figure shows an iterative process that you can use to understand and optimize how the system handles workload.



1. First you look at how the system is currently performing in terms of CPU usage and memory consumption. What kinds of workloads are running on the system, are there complex, long running queries that require lots of memory?
2. When you have a broad understanding of the activity in the system you can drill down in to the details such as business importance. Are statements being run that are strategic or analytic in nature compared to standard reporting that may not be so time critical. Can those statements be optimized to run more efficiently?
3. Then, when you have a deeper understanding of the system you have a number of ways to influence how it handles the workload. You can start to map the operations to available resources such as CPU and memory and determine the priority that requests get by, for example, using workload classes.

8.12.1.3 Analyzing System Performance

You can use system views to analyze how effectively your system is handling the current workload.

This section lists some of the most useful views available which you can use to analyze your workload and gives suggestions about actions that you might take to improve performance. Refer also to the scenarios section for more details of how these analysis results can help you to decide about which workload management options to apply.

Other performance analysis issues are described in the *SAP HANA Troubleshooting and Performance Analysis Guide*.

Analyzing SQL Statements

Use these views to analyze the performance of SQL statements:

- M_ACTIVE_STATEMENTS
- M_PREPARED_STATEMENTS

- M_EXPENSIVE_STATEMENTS

If these views indicate problems with statements you can use workload classes to tune the statements by limiting memory or parallelism.

Consider also the setting of any session variables (in M_SESSION_CONTEXT) which might have a negative impact on these statements. The following references provide more detailed information on this:

- SAP Note 2215929 *Using Client Info to set Session Variables and Workload Class settings* describes how client applications set session variables for dispatching workload classes.
- The *SAP HANA Developer Guide* (Setting Session-Specific Client Information).

Analyzing CPU Activity

Use these views to analyze CPU activity:

- M_SERVICE_THREADS
- M_SERVICE_THREAD_SAMPLES
- M_EXPENSIVE_STATEMENTS.CPU_TIME (column)
- M_SERVICE_THREAD_CALLBACKS (stack frame information for service threads)
- M_JOBEXECUTORS (job executor statistics)

These views provide detailed information on the threads that are active in the context of a particular service and information about locks held by threads.

If these views show many threads for a single statement, and the general system load is high you can adjust the settings for the set of 'execution' ini-parameters as described in the topic *Controlling Parallel Execution*.

Related Information

Controlling CPU Consumption [page 421]
Controlling Parallel Execution of SQL Statements [page 428]
Setting a Memory Limit for SQL Statements [page 431]
Managing Workload with Workload Classes [page 439]
Example Workload Management Scenarios [page 457]
Managing and Monitoring SAP HANA Performance [page 148]
SAP Note 2215929

8.12.2 Controlling CPU Consumption

SAP HANA can exploit the processor's NUMA architecture to optimize performance by applying NUMA core affinity settings to server processes and by specifying NUMA node location preferences for specific database objects.

Non-uniform memory access (NUMA) processors have their own local memory and have access to memory shared between processors. The memory access time depends on the memory location relative to the processor: processors can access their own local memory faster than non-local memory.

SAP HANA can exploit the processor's NUMA architecture to optimize performance by:

1. Applying NUMA core affinity settings to server processes. With this option you can use the affinity configuration parameter to restrict CPU usage of SAP HANA server processes to certain CPUs or ranges of CPUs. You can set restrictions for the following servers: nameserver, indexserver, compileserver, preprocessor, and xsengine.
2. Specifying NUMA node location preferences for specific database objects. With this option you can specify in SQL NUMA node location preferences for specific database objects using either the CREATE TABLE or ALTER TABLE statements. NUMA node location preferences can be applied at any of the following levels:

- Table (column store)
- Table Partition (range partitioning)
- Column

Before making any changes, however, you must begin by analyzing the system topology using, for example, the Linux `lscpu` command or the SAP HANA monitoring views.

> → Tip
>
> As an alternative to applying CPU affinity settings you can achieve similar performance gains by changing the parameter [execution] `max_concurrency` in the global.ini configuration file. Changing the configuration file may be more convenient and doesn't require the system to be offline.

This section gives details and examples of these procedures. Further information can also be found in KBA 2470289: *FAQ: SAP HANA Non-Uniform Memory Access (NUMA)*.

Related Information

Examining the System Topology [page 422]
Applying Affinity Configuration Changes [page 424]
Applying NUMA Location Preferences to Tables [page 426]
SAP Note 2470289

8.12.2.1 Examining the System Topology

On systems with many CPU cores and NUMA nodes it is important to understand the system topology in order to optimize the system to achieve good performance and robust response times.

Firstly, to confirm the physical and logical details of your CPU architecture, analyze the system using operating system commands and / or system views.

For Xen and VMware, the users in the VM guest system see what is configured in the VM host. So, the quality of the reported information depends on the configuration of the VM guest. Therefore, SAP can't give any performance guarantees in this case.

Operating System Commands

lscpu

The `lscpu` command returns a listing of details of the system architecture. The following table gives a commentary on the most useful values based on an example system with 2 physical chips (sockets), each containing 8 physical cores. These cores are hyperthreaded to give a total of 32 logical cores.

#	Feature	Example Value
1	Architecture	x86_64
2	CPU op-modes	32-bit, 64-bit
3	Byte Order	LittleEndian
4	CPUs	32
5	On-line CPUs list	0–31
6	Threads per core	2
7	Cores per socket	8
8	Sockets	2
9	NUMA nodes	2
21	NUMA node0 CPUs	0-7,16-23
22	NUMA node1 CPUs	8-15,24-31

- 4–5: This example server has 32 logical cores numbered 0–31
- 6–8: Logical cores ("threads") are assigned to physical cores. Hyperthreading is where multiple threads are assigned to a single physical core In this example, there are 2 sockets, with each socket containing 8 physical cores (total 16). Two logical cores (#6) are assigned to each physical core, thus, each core exposes two execution contexts for the independent and concurrent execution of two threads.
- 9: In this example there are 2 NUMA nodes, one for each socket. Other systems may have multiple NUMA nodes per socket.

- 21–22: The 32 logical cores are numbered and assigned to one of the two NUMA nodes.

> **Note**
>
> Even on a system with 32 logical cores and two sockets the assignment of logical cores to physical CPUs and sockets can be different. It's important to collect the assignment in advance before making changes. A more detailed analysis is possible using the system commands described in the next step. These provide detailed information for each core including how CPU cores are grouped as siblings.

Topology Commands

You can also use the set of CPU topology commands in the `/sys/devices/system/cpu/` directory tree. For each logical core, there is a corresponding numbered subdirectory (`/cpu12/` in the following examples). The examples show how to retrieve this CPU information and the table gives details of some of the most useful commands available:

> **Example**
>
> ```
> cat /sys/devices/system/cpu/present
> ```
>
> ```
> cat /sys/devices/system/cpu/cpu12/topology/thread_siblings_list
> ```

Command	Example Output	Commentary
present	0–15	The number of logical cores available for scheduling.
cpu12/topology/core_siblings_list	4–7, 12-15	The cores on the same socket.
cpu12/topology/thread_siblings_list	4, 12	The logical cores assigned to the same physical core (hyperthreading).
cpu12/topology/physical_package_id	1	The socket of the current core - in this case cpu12.

Other Linux commands that are relevant here are `sched_setaffinity` and `numactl`. `sched_setaffinity` limits the set of CPU cores available (by applying a CPU affinity mask) for execution of a specific process (this can be used, for example, to isolate tenants) and `numactl` controls NUMA policy for processes or shared memory.

hdbcons

The database management client console `hdbcons` can also be used to retrieve topology information, using for example, the job executor information command. In this case the ID of the indexserver process is required as a parameter:

```
hdbcons -p <PID> "jexec info"
```

SAP HANA Monitoring Views

Information about the system topology and the CPU configuration is available in system views. You can run the example queries given here on the tenant or system database.

M_HOST_INFORMATION provides host information such as machine and operating system configuration. Data in this view is stored in key-value pair format, the values are updated once per minute. For most keys the INIFILE ADMIN privilege is required to view the values. Select one or more key names for a specific host to retrieve the corresponding values:

```
select * from SYS.M_HOST_INFORMATION where key in
('cpu_sockets','cpu_cores','cpu_threads');
```

M_NUMA_RESOURCES provides information on overall resource availability for the system:

```
select MAX_NUMA_NODE_COUNT, MAX_LOGICAL_CORE_COUNT from SYS.M_NUMA_RESOURCES;
```

M_NUMA_NODES provides resource availability information on each NUMA node in the hardware topology, including inter-node distances and neighbor information, for example:

```
select HOST, NUMA_NODE_ID, NUMA_NODE_DISTANCES, MEMORY_SIZE from
SYS.M_NUMA_NODES;
```

M_CS_COLUMNS has details of tables, columns and partitions and the following example may be helpful to check the NUMA locations where tables are distributed across NUMA nodes. In this case CPU load is distributed across NUMA nodes by NUMA-local processing, but memory consumption is also better distributed to the memory attached to each NUMA node.

```
select table_name, column_name, part_id, string_agg(numa_node_index, ',') from
m_cs_columns
group by table_name, column_name, part_id
order by table_name, column_name, part_id;
```

Refer to the *SAP HANA SQL and System Views Reference Guide* for full details of these views.

Related Information

SAP HANA SQL Reference Guide for SAP HANA Platform

8.12.2.2 Applying Affinity Configuration Changes

You can use the `affinity` configuration parameter to restrict CPU usage of SAP HANA server processes to certain CPUs or ranges of CPUs.

Based on the information returned from the examination of the topology, apply affinity settings in `daemon.ini` to bind specific processes to logical CPU cores. This approach applies primarily to the use cases of SAP HANA tenant databases and multiple SAP HANA instances on one server; you can use this approach, for example, to partition the CPU resources of the system by tenant database.

You can set restrictions for the following servers: nameserver, indexserver, compileserver, preprocessor, and xsengine (each server has a section in the daemon.ini file). The affinity setting is applied by the TrexDaemon when it starts the other HANA processes using the command `sched_setaffinity`. Changes to the affinity settings take effect only after restarting the HANA process.

To make the changes described here you require the privilege INIFILE ADMIN.

After making changes you can rerun the topology analysis queries and operating system commands to verify the changes you have made.

Examples

The following examples and commentary show the syntax for the ALTER SYSTEM CONFIGURATION commands required.

Example

To restrict the nameserver to two logical cores of the first CPU of socket 0, use the following affinity setting:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET ('nameserver',
'affinity') = '0,16'
```

Example

To restrict the preprocessor and the compileserver to all remaining cores (that is, all except 0 and 16) on socket 0, use the following affinity settings:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('preprocessor', 'affinity') = '1-7,17-23'
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('compileserver', 'affinity') = '1-7,17-23'
```

Example

To restrict the indexserver to all cores on socket 1, use the following affinity settings:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET ('indexserver',
'affinity') = '8-15,24-31'
```

Multi-Tenant Database Examples

You can assign affinities to different tenants of a multi-tenant database on the same host as shown here. Run these SQL statements on the SYSTEMDB.

Example

In this scenario tenant NM1 already exists, here we add another tenant NM2:

```
CREATE DATABASE NM2 ADD AT LOCATION 'host:30040' SYSTEM USER PASSWORD
Manager1;
```

Set the configuration parameter to bind CPUs to specific NUMA nodes on each tenant. You can use the following notation with a dot to identify the specific tenant:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini','SYSTEM')  SET
('indexserver.NM1', 'affinity') ='0-7,16-23';
```

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini','SYSTEM')  SET
('indexserver.NM2', 'affinity') ='8-15,24-31';
```

To assign affinities to multiple indexservers of the same tenant on the same host, execute the following SQL statements on the SYSTEMDB to apply the `instance_affinity[port]` configuration parameter:

Example

In this scenario, an indexserver is already running on tenant NM1 on port 30003. Here we add another indexserver on a different port:

```
ALTER DATABASE NM1 ADD 'indexserver' AT LOCATION 'host:30040';
```

Set the different instances of the `instance_affinity[port]` configuration parameter to bind CPUs to specific NUMA nodes on each indexserver. The configuration parameter has a 1–2 digit suffix to identify the final significant digits of the port number, in this example 3000**3** and 300**40**:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini','SYSTEM')  SET
('indexserver.NM1', 'instance_affinity[3]')='0-7,16-23';
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini','SYSTEM')  SET
('indexserver.NM1', 'instance_affinity[40]')='8-15,24-31';
```

Restart the indexserver processes to make the affinity settings effective.

8.12.2.3 Applying NUMA Location Preferences to Tables

You can specify NUMA node location preferences for specific database objects in SQL using either the CREATE TABLE or ALTER TABLE statements.

NUMA Node Preferences

To apply NUMA node location preferences in SQL for tables, columns or partitions, you can use the <numa_node_preference_clause> (NUMA NODE keywords) followed by a list of one or more preferred node locations. Refer to the previous subsection for how to use the `lscpu` command to understand the system topology. Refer to the *SAP HANA SQL and System Views Reference* for full details of this feature.

You can specify either individual nodes or a range of nodes as shown in the following example:

```
CREATE COLUMN TABLE T1(A int, B varchar(10)) NUMA NODE (‘1’, ‘3’ TO ‘5’)
```

In this example table T1 will be processed by NUMA node 1 if possible and otherwise by any of NUMA nodes 3-5. Preferences are saved in system table NUMA_NODE_PREFERENCE_.

Use the following statement to remove any preferences for an object:

```
ALTER TABLE T1 NUMA NODE NULL
```

By default, preferences are only applied the next time the table is loaded (this is the default DEFERRED option). You can also use ALTER TABLE with this clause and the keyword IMMEDIATE to apply the preference immediately:

```
ALTER TABLE T1 NUMA NODE (‘3’) IMMEDIATE
```

The IMMEDIATE option is not effective if the Fast Restart Option or PMEM are enabled on this table; in this case the new NUMA node will only be used after the column is rewritten into persistent memory. You can force this by either running a delta merge or explicitly deleting persistent memory so that it is rewritten with the next load: `UNLOAD <table> DELETE PERSISTENT MEMORY;`

Granularity

NUMA node location preferences can be applied at any of the following levels:

- Table (column store only)
- Table Partition (range partitioning only)
- Column

If multiple preferences for a column or partition have been defined then the column preference is applied first, then the partition preference, then the table.

The following example shows the statement being used to apply a preference for column A in table T1:

```
CREATE COLUMN TABLE T1(A int NUMA NODE (‘2’), B varchar(10))
```

The following examples show statements to apply a preference for partition A in table T1:

```
CREATE COLUMN TABLE T1(A int , B varchar(10)) PARTITION BY RANGE(A) (PARTITION
VALUE = 2 NUMA NODE (‘4’))
```

```
ALTER TABLE T1 ADD PARTITION (A) VALUE = 3 NUMA NODE (‘1’) IMMEDIATE
```

It is also possible to identify a partition by its logical partition ID number and set a preference using ALTER TABLE as shown here:

```
ALTER TABLE T1 ALTER PARTITION 2 NUMA NODE ('3')
```

You can verify the assignment of partitions to nodes by querying the values for NODE_ID and NUMA_NODE_INDEXES from the TABLE_PARTITIONS system view.

Transferring Preferences

Using the CREATE TABLE LIKE statement the new table can be created with or without the NUMA preference. In the following example any preference which has been applied to T2 will (if possible) apply on new table T1. The system checks the topology of the target system to confirm if it has the required number of nodes, if not, then the preference is ignored:

```
CREATE TABLE T1 LIKE T2
```

The keyword WITHOUT can be used as shown in the following example to ignore any preference which has been applied to T2 when creating the new table T1:

```
CREATE TABLE T1 LIKE T2 WITHOUT NUMA NODE
```

A similar approach is used with the IMPORT and EXPORT Statements; any preference is saved in the exported table definition and applied, if possible, in the target environment when the table is imported. In this case you can use the IGNORE keyword to import a table and ignore any node preferences:

```
IMPORT SYSTEM."T14" FROM '/tmp/test/' WITH REPLACE THREADS 40 IGNORE NUMA NODE
```

Related Information

Applying Affinity Configuration Changes [page 424]

SAP HANA SQL Reference Guide for SAP HANA Platform
Create Table Statement (see numa_node_preference_clause)

8.12.3 Controlling Parallel Execution of SQL Statements

You can apply ini file settings to control the two thread pools SqlExecutor and JobExecutor that control the parallelism of statement execution.

> ⚠ Caution
>
> The settings described here should only be modified when other tuning techniques like remodeling, repartitioning, and query tuning have been applied. Modifying the parallelism settings requires a thorough understanding of the actual workload since they have impact on the overall system behavior. Modify the settings iteratively by testing each adjustment. For more information, see *Understand your Workload*.

On systems with highly concurrent workload, too much parallelism of single statements may lead to sub-optimal performance. Note also that partitioning tables influences the degree of parallelism for statement execution; in general, adding partitions tends to increase parallelism. You can use the parameters described in this section to adjust the CPU utilization in the system.

Two thread pools control the parallelism of the statement execution. Generally, target thread numbers applied to these pools are soft limits, meaning that additional available threads can be used if necessary and deleted when no longer required:

- SqlExecutor
  This thread pool handles incoming client requests and executes simple statements. For each statement execution, an SqlExecutor thread from a thread pool processes the statement. For simple OLTP-like statements against column store as well as for most statements against row store, this will be the only type of thread involved. With OLTP we mean short running statements that consume relatively little resources, however, even OLTP-systems like SAP Business Suite may generate complex statements.
- JobExecutor
  The JobExecutor is a job dispatching subsystem. Almost all remaining parallel tasks are dispatched to the JobExecutor and its associated JobWorker threads.
  In addition to OLAP workload the JobExecutor also executes operations like table updates, backups, memory garbage collection, and savepoint writes.

You can set a limit for both SqlExecutor and JobExecutor to define the maximum number of threads. You can use this for example on a system where OLAP workload would normally consume too many CPU resources to apply a maximum value to the JobExecutor to reserve resources for OLTP workload.

> ⚠ Caution
>
> Lowering the value of these parameters can have a drastic effect on the parallel processing of the servers and reduce the performance of the overall system. Adapt with caution by iteratively making modifications and testing. For more information, see *Understand your Workload* and *SAP Note 2222250 - FAQ SAP HANA Workload Management* which contains more details of the workload configuration parameters.

A further option to manage statement execution is to apply a limit to an individual user profile for all statements in the current connection using 'THREADLIMIT' parameter. This option is described in *Setting User Parameters*.

Parameters for SqlExecutor

The following SqlExecutor parameters are in the `sql` section of the `indexserver.ini` file.

`sql_executors` - sets a soft limit on the target number of logical cores for the SqlExecutor pool.

- This parameter sets the target number of threads that are immediately available to accept incoming requests. Additional threads will be created if needed and deleted if not needed any more.
- The parameter is initially not set (0); the default value is the number of logical cores in a system. As each thread allocates a particular amount of main memory for the stack, reducing the value of this parameter can help to avoid memory footprint.

`max_sql_executors` - sets a hard limit on the maximum number of logical cores that can be used.

- In normal operation new threads are created to handle incoming requests. If a limit is applied here, SAP HANA will reject new incoming requests with an error message if the limit is exceeded.
- The parameter is initially not set (0) so no limit is applied.

> ⚠ Caution
>
> SAP HANA will not accept new incoming requests if the limit is exceeded. Use this parameter with extreme care.

Parameters for JobExecutor

The following JobExecutor parameters are in the `execution` section of the `global.ini` or `indexserver.ini`.

`max_concurrency` - sets the target number of logical cores for the JobExecutor pool.

- This parameter sets the size of the thread pool used by the JobExecutor used to parallelize execution of database operations. Additional threads will be created if needed and deleted if not needed any more. You can use this to limit resources available for JobExecutor threads, thereby saving capacity for SqlExecutors.

- The parameter is initially not set (0); the default value is the number of logical cores in a system. Especially on systems with at least 8 sockets consider setting this parameter to a reasonable value between the number of logical cores per CPU up to the overall number of logical cores in the system. In a system that supports tenant databases, a reasonable value is the number of cores divided by the number of tenant databases.

`max_concurrency_hint` - limits the number of logical cores for job workers even if more active job workers would be available.

- This parameter defines the number of jobs to create for an individual parallelized operation. The JobExecutor proposes the number of jobs to create for parallel processing based on the recent load on the system. Multiple parallelization steps may result in far more jobs being created for a statement (and hence higher concurrency) than this parameter.
- The default is 0 (no limit is applied but the hint value is never greater than the value for `max_concurrency`). On large systems (that is more than 4 sockets) setting this parameter to the number of logical cores of one socket may result in better performance but testing is necessary to confirm this.

`default_statement_concurrency_limit` - restricts the actual degree of parallel execution per connection within a statement.

- This parameter controls the maximum overall parallelism for a single database request. Set this to a reasonable value (a number of logical cores) between 1 and `max_concurrency` but greater or equal to the value set for `max_concurrency_hint`.
- The default setting is 0; no limit is applied. Note, however, that a new default value for this parameter may also be set during installation using the SAP HANA database lifecycle manager (HDBLCM) tool. In this case, a limit may already be in force. See also SAP Note 3011356 - *Default Value of global.ini [execution] default_statement_concurrency_limit Changed*.

`default_statement_concurrency_max_limit` - can be used to relax the parallel processing restrictions defined in the above parameter. If a max limit is set, when load on the system is low then more resources than would be otherwise available can be used. The parameter can be set as either a numeric value (interpreted as a number of running jobs) or as a percentage, which is interpreted as a fraction of the available computation resources. For example, if a setting of 50% is applied and less than 50% of resources are in use by JobWorkers, then the limit applied to the resources that an individual statement can consume is relaxed, up to this limit of 50%.

- The default value is zero so the parameter is not initially active.

Related Information

Understand your Workload [page 418]
Example Workload Management Scenarios [page 457]
Setting User Parameters for Workload [page 437]
SAP Note 2222250
SAP Note 3011356

8.12.4 Setting a Memory Limit for SQL Statements

You can set a statement memory limit to prevent single statements from consuming too much memory.

Prerequisites

To apply these settings you must have the system privilege INIFILE ADMIN.

For these options, `enable_tracking` and `memory_tracking` must first be enabled in the `global.ini` file. Additionally, `resource_tracking` must be enabled in this file if you wish to apply different settings for individual users (see Procedure below).

Context

You can protect an SAP HANA system from uncontrolled queries consuming excessive memory by limiting the amount of memory used by single statement executions per host. By default, there is no limit set on statement memory usage, but if a limit is applied, statement executions that require more memory will be aborted when they reach the limit. To avoid canceling statements unnecessarily you can also apply a percentage threshold value which considers the current statement allocation as a proportion of the global memory currently available. Using this parameter, statements which have exceeded the hard-coded limit may still be executed if the memory allocated for the statement is within the percentage threshold. The percentage threshold setting is also effective for workload classes where a statement memory limit can also be defined.

You can also create exceptions to these limits for individual users (for example, to ensure an administrator is not prevented from doing a backup) by setting a different statement memory limit for each individual.

These limits only apply to single SQL statements, not the system as a whole. Tables which require much more memory than the limit applied here may be loaded into memory. The parameter `global_allocation_limit` limits the maximum memory allocation limit for the system as a whole.

You can view the (peak) memory consumption of a statement in M_EXPENSIVE_STATEMENTS.MEMORY_SIZE.

Procedure

1. Enable statement memory tracking.

   In the `global.ini` file, expand the resource_tracking section and set the following parameters to **on**:
   - `enable_tracking = on`
   - `memory_tracking = on`
2. `statement_memory_limit` - defines the maximum memory allocation per statement in GB. The parameter is not set by default.
   - In the `global.ini` file, expand the memorymanager section and locate the parameter. Set an integer value in GB between 0 (no limit) and the value of the global allocation limit. Values that are too small can block the system from performing critical tasks.

- An option is also available to apply a limit in relation to the system's effective allocation limit. To do this, set the parameter to a value of '-1'. In this case a memory limit is applied calculated as 25% of the smaller of the global_allocation_limit and the process_allocation_limit.
- A value for this parameter may also be set during installation or upgrade by hdblcm.
- The value defined for this parameter can be overridden by the corresponding workload class property STATEMENT_MEMORY_LIMIT.
- When the statement memory limit is reached, a dump file is created with 'compositelimit_oom' in the name. The statement is aborted, but otherwise the system is not affected. By default only one dump file is written every 24 hours. If a second limit hits in that interval, no dump file is written. The interval can be configured in the memorymanager section of the `global.ini` file using the `oom_dump_time_delta` parameter, which sets the minimum time difference (in seconds) between two dumps of the same kind (and the same process).

After setting this parameter, statements that exceed the limit you have set on a host are stopped by running out of memory.

3. `statement_memory_limit_threshold` - defines a percentage of the global allocation limit. Parameter `statement_memory_limit` is respected only if total used memory exceeds this threshold. The default value is 0% (of the global_allocation_limit) so `statement_memory_limit` is always respected.
   - In the `global.ini` file, expand the memorymanager section and set the parameter as a percentage of the global allocation limit.
   - This parameter provides a means of controlling when the `statement_memory_limit` is applied. If this parameter is set, when a statement is issued the system will determine if the amount of memory it consumes exceeds the defined percentage value of the overall `global_allocation_limit` parameter setting. The statement memory limit is only applied if the current SAP HANA memory consumption exceeds this statement memory limit threshold as a percentage of the global allocation limit.
   - This is a way of determining if a particular statement consumes an inordinate amount of memory compared to the overall system memory available. If so, to preserve memory for other tasks, the statement memory limit is applied and the statement fails with an exception.
   - Note that the value defined for this parameter also applies to the workload class property STATEMENT_MEMORY_LIMIT.
4. `total_statement_memory_limit` - a value in gigabytes to define the maximum memory available to all statements running on the system. The default value is 0 (no limit).
   - This limit does not apply to users with the administrator role SESSION ADMIN or WORKLOAD ADMIN who need unrestricted access to the system. However, a check of the user's privileges allowing the administrator to by-pass the limit is only made for the first request when a connection is made. The privileged user would have to reconnect to be able to bypass the statement memory limit again (see also Admission Control).
   - The value defined for this parameter cannot be overridden by the corresponding workload class property TOTAL_STATEMENT_MEMORY_LIMIT.
   - There is a corresponding parameter for use with system replication on an Active/Active (read enabled) secondary server. This is required to ensure that enough memory is always available for essential log shipping activity. See also `sr_total_statement_memory_limit` in section *Memory Management*.
5. User parameters can limit memory for statements. For further information, refer to *Setting User Parameters for Workload*.

Results

The following example and scenarios show the effect of applying these settings:

Example showing statement memory parameters

Parameter	Value
Physical memory	128 GB
`global_allocation_limit`	The unit used by this parameter is MB. The default value is: 90% of the first 64 GB of available physical memory on the host plus 97% of each further GB; or, in the case of small physical memory, physical memory minus 1 GB.
`statement_memory_limit`	1 GB (the unit used by this parameter is GB.)
`statement_memory_limit_threshold`	60%

Scenario 1:

A statement allocates 2GB of memory and the current used memory size in SAP HANA is 50GB.

- 0,9 * 128GB = 115,2 (global allocation limit)
- 0,6 * 115,2 = 69,12 (threshold in GB)
- 50 GB < 69,12 GB (threshold not reached)

The statement is executed, even though it exceeds the 1GB `statement_memory_limit`.

Scenario 2:

A statement allocates 2GB and the current used memory size in SAP HANA is 70GB

- 70 GB > 69,12 GB (threshold is exceeded)

The statement is cancelled, as the threshold is exceeded, the `statement_memory_limit` is applied.

Related Information

Change the Global Memory Allocation Limit [page 179]
Resource Management [page 862]
Setting User Parameters for Workload [page 437]
Managing Peak Load (Admission Control) [page 434]

8.12.5 Managing Peak Load (Admission Control)

Use the admission control feature to apply processing limits and to decide how to handle new requests if the system is close to the point of saturation.

You can apply thresholds using configuration parameters to define an acceptable limit of activity in terms of the percentage of memory usage or percentage of CPU capacity.

Admission Control can be maintained either from the SQL command line or in SAP HANA cockpit. For further specific details refer to *Manage Admission Control* in the SAP HANA cockpit documentation.

Limits can be applied at two levels so that firstly new requests will be queued until adequate processing capacity is available or a timeout is reached, and secondly, a higher threshold can be defined to determine the maximum workload level above which new requests will be rejected. If requests have been queued, items in the queue are processed when the load on the system reduces below the threshold levels. If the queue exceeds a specified size or if items are queued for longer than a specified period of time they are rejected.

If a request is rejected an error message is returned to the client that the server is temporarily overloaded: `1038,'ERR_SES_SERVER_BUSY','rejected as server is temporarily overloaded'`, in this case the connection is rolled back.

The load on the system is measured by background processes which gather a set of performance statistics covering available capacity for memory and CPU usage. The statistics are moderated by a configurable averaging factor (exponentially weighted moving average) to minimize volatility, and the moderated value is used in comparison with the threshold settings.

There are some situations where it is not recommended to enable admission control, for example, during planned maintenance events such as an upgrade or the migration of an application. In these cases it is expected that the load level is likely to be saturated for a long time and admission control could therefore result in the failure of important query executions.

By-passing admission control The admission control filtering process does not apply to all requests. In particular, requests that release resources will always be executed, for example, Commit, Rollback, Disconnect and so on.

The filtering also depends on user privileges: administration requests from SESSION_ADMIN and WORKLOAD_ADMIN are always executed. However, this is limited: the check on user privileges is only made for the first request when a connection is made. For subsequent requests no check is made and all users, regardless of privileges, are subject to the workload controls in place. The user with administrator privileges would have to reconnect to be able to bypass the controls again. This functionality was implemented to reduce the overhead incurred in checking privileges for routed connections. No privilege check takes place when admission control is disabled (see following section on Configuration).

Admission control limitation with Active-Active (Read Only): admission control evaluates requests at the session layer as part of statement preparation before decoding the request packet from the client, whereas the decision about routing the request is made later in the SQL engine. This means that in an active-active setup if the admission control load threshold is exceeded on the primary then the incoming request is queued on the primary system. Statements which have been prepared and a decision to route the request to the secondary has already been made would directly connect to the secondary.

Related Information

Manage Admission Control (SAP HANA cockpit)

8.12.5.1 Configuring Admission Control

Threshold values for admission control to determine when requests are queued or rejected are defined as configuration parameters.

The admission control feature is enabled by default and the related threshold values and configurable parameters are available in the indexserver.ini file. A pair of settings is available for both memory (not active by default) and CPU which define firstly the queuing level and secondly the rejection level. A high-level alert is raised if there are any session requests queued or rejected by admission control.

These parameters, in the *admission_control* section of the ini file, are summarized in the following table (see also Queue Management below):

Parameter	Default	Detail
enable	True	Enables or disables the admission control feature.
queue_cpu_threshold	90	The percentage of CPU usage above which requests will be queued. Queue details are available in the view M_ADMISSION_CONTROL_QUEUES. The value 0 or 100 means that no requests are queued.
queue_memory_threshold	0	The percentage of memory usage above which requests will be queued. The value 0 or 100 means that no requests are queued.
reject_cpu_threshold	0	The percentage of CPU usage above which requests will be rejected. The default value 0 means that no requests are rejected, but may be queued.
reject_memory_threshold	0	The percentage of memory usage above which requests will be rejected. The default value 0 means that no requests are rejected, but may be queued.

The following parameters are available to manage the statistics collection process by defining how frequently statistics are collected and setting the averaging factor which is used to moderate volatility. Smoothing is done by an exponentially weighted moving average calculation based on a configurable percentage factor value. By setting a collection frequency value and setting the averaging factor you can moderate the effect of sudden peaks in usage. In practice a low factor value has a strong moderating effect which may not detect peaks in the load. A value above 80% may be too high for a volatile workload.

Parameter	Default	Detail
statistics_collection_interval	1000	Unit milliseconds. The statistics collection interval is set by default to 1000ms (1 second) which has a negligible effect on performance. Values

Parameter	Default	Detail
		from 100ms are supported. Statistics details are visible in the view M_ADMISSION_CONTROL_STATISTICS.
averaging_factor	70	This percentage value gives a weighting to the statistic averaging process: a low value has a strong moderating effect (but may not adequately reflect real CPU usage) and a value of 100% means that no averaging is performed, that is, only the current value for memory and CPU consumption is considered.
averaging_factor_cpu	0	This parameter can be used specifically to smooth statistics on CPU usage. It is not set by default (value zero) but if it is set then it overrides the averaging_factor parameter and this CPU value is applied as a factor to average CPU statistics only.
averaging_factor_memory	0	This parameter can be used specifically to smooth statistics on memory usage. It is not set by default (value zero) but if it is set then it overrides the averaging_factor parameter and this memory value is applied as a factor to average memory statistics only.

Events and Rejection Reasons

If statements are being rejected you may need to investigate why this is happening. Events related to admission control are logged and can be reviewed in the M_ADMISSION_CONTROL_EVENTS view. The key information items here are the event type (such as a statement was rejected or a statement was queued or dequeued) and the event reason which gives an explanatory text related to the type. Other details in this view include the length of time the statement was queued and the measured values for memory and CPU usage.

Two parameters are available to manage the event log in the *admission_control_events* section of the ini file:

Parameter	Default	Detail
queue_wait_time_threshold	5000000	The length of time measured in microseconds for which a request must be queued above which it is included in the event log (default is 5 seconds). If the parameter is set to 0 then events are not logged.
record_limit	10000	The maximum record count permitted in the monitor of historical events.

Queue Management

If requests have been queued, items in the queue are processed when capacity becomes available. A background job continues to evaluate the load on the system in comparison to the thresholds and when the load is reduced enough queued requests are submitted in batches on an oldest-first basis.

The queue status of a request is visible in the M_CONNECTIONS view; the connection status value is set to *queuing* in column M_CONNECTIONS.CONNECTION_STATUS.

There are several configuration parameters (in the *admission_control* section of the ini file) to manage the queue and how the requests in the queue are released. You can apply a maximum queue size or a queue timeout value; if either of these limits are exceeded then requests which would otherwise be queued will be rejected. An interval parameter is available to determine how frequently to check the server load so that de-queueing can start, and a de-queue batch size setting is also available.

Parameter	Default	Detail
max_queue_size	10000	The maximum number of requests which can be queued. Requests above this number will be rejected.
dequeue_interval	1000	Unit: milliseconds. Use this parameter to set the frequency of the check to reevaluate the load in comparison to the thresholds. The default is 1000ms (1 second). This value is recommended to avoid overloading the system, though values from 100ms are supported.
dequeue_size	50	Use this parameter to set the de-queue batch size, that is, the number of queued items which are released together once the load is sufficiently reduced. This value can be between 1 and 9999 queued requests.
queue_timeout	600	Unit: seconds. Use this parameter to set the maximum length of time for which items can be queued. The default is 10 minutes. The minimum value which can be applied is 60 seconds, there is no maximum limit. Requests queued for this length of time will be rejected. Note that the timeout value applies to all entries in the queue. Any changes made to this configuration value will be applied to all entries in the existing queue.
queue_timeout_check_interval	10000	Unit: milliseconds. Use this parameter to determine how frequently to check if items have exceeded the queue timeout limit. The default is 10 seconds. The minimum value which can be applied is 100 milliseconds, there is no maximum limit.

> **Note**
>
> If Admission Control has been configured and is active it takes precedence over any other timeout value which might have been applied. This means other timeouts which apply to a query (such as a query timeout) would not be effective until the query has been dequeued or rejected by the queue time out.

8.12.6 Setting User Parameters for Workload

In workload management, there are three user parameters that you can use in order to control resource allocation.

It is possible to set values for the parameters STATEMENT MEMORY LIMIT, STATEMENT THREAD LIMIT and PRIORITY on user level. You can perform this from the SQL command line. Alternatively, you can set the parameters in the *Workload Classes* page of the SAP HANA cockpit, choosing the *User-Specific Parameters* tab. USER ADMIN privileges are required for setting these values. For further information, refer to *Create User-Specific Parameters*.

The settings for a given user can be queried from the USER_PARAMETERS system view where they are stored as key-value pairs:

```
SELECT * FROM USER_PARAMETERS WHERE USER_NAME = 'SYSTEM';
```

The values of the user parameters currently applied for the current session are shown in M_CONNECTIONS. The following example shows the current connection for the PRIORITY user parameter:

```
SELECT PRIORITY FROM M_CONNECTIONS WHERE CONNECTION_ID = CURRENT_CONNECTION;
```

STATEMENT MEMORY LIMIT

Set a value (in GB) for the statement memory limit with the following SQL statement:

```
ALTER USER MyUserId SET PARAMETER STATEMENT MEMORY LIMIT = '1';
```

This statement excludes a user from the global limit.

Note that this user parameter-based approach to limiting memory for statements is not supported for cross-database queries, nor is it effective in XSC developed applications. In these cases you can apply memory limits using workload classes in the remote tenant database. For further information on cross-database queries, refer to *Workload Management and Cross-Database Queries*.

If both a global and a user statement memory limit are set, the user-specific limit takes precedence, regardless of whether it is higher or lower than the global statement memory limit. If the user-specific statement memory limit is removed, the global limit takes effect for the user.

> **Note**
>
> Setting the statement memory limit to 0 will disable any statement memory limit for the user. Alternatively, to reset a user-specific limit use the CLEAR option:
>
> ```
> ALTER USER MyUserId CLEAR PARAMETER STATEMENT MEMORY LIMIT
> ```

STATEMENT THREAD LIMIT

Set a value to limit the concurrency of statements with the following SQL statement:

```
ALTER USER MyUserId SET PARAMETER STATEMENT THREAD LIMIT = '20';
```

PRIORITY

You can set a user-level priority value for all statements in the current connection; the range of possible values is from 0 to 9 (the default is 5). Use the following statement:

```
ALTER USER MyUserId SET PARAMETER PRIORITY = '9';
```

Note that the user priority value is not effective in XSC developed applications. For XSC applications you can apply a priority value using workload classes.

Related Information

Create User-Specific Parameters
Workload Management and Cross-Database Queries [page 94]
Setting a Memory Limit for SQL Statements [page 431]

8.12.7 Managing Workload with Workload Classes

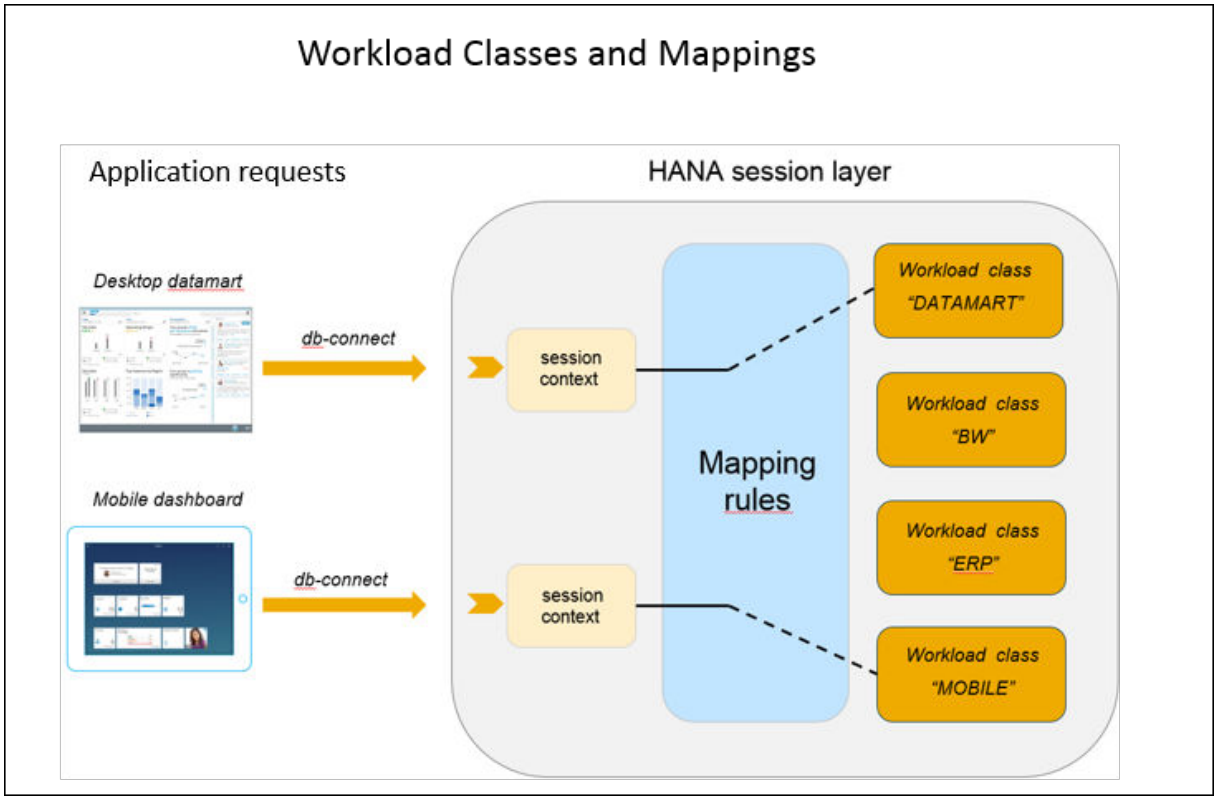
You can manage workload in SAP HANA by creating workload classes and workload class mappings. Appropriate workload parameters are then dynamically applied to each client session.

You can classify workloads based on user and application context information and apply configured resource limitations (related to statement memory or thread limits) or a priority value. Workload classes allow SAP HANA to influence dynamic resource consumption on the session or statement level.

Workload classes and workload class mappings can be maintained either from the SQL command line or in SAP HANA cockpit. For further specific details refer to Workload Management Statements in the *SAP HANA SQL and System Views Reference Guide* and to *Managing Workload Classes in SAP HANA cockpit* in the SAP HANA cockpit documentation.

Workload class settings override other configuration settings (ini file values) which have been applied. Workload class settings also override user parameter settings which have been applied by the SQL command ALTER USER, however, workload class settings only apply for the duration of the current session, whereas changes applied to the user persist. More detailed examples of precedence are given in a separate section.

To apply workload class settings client applications can submit client attribute values (session variables) in the interface connect string as one or more property-value pairs. The key values which can be used to work with workload classes are: database user, client, application name, application user and application type. Based on this information the client is classified and mapped to a workload class. If it cannot be mapped it is assigned to the default workload class. The configuration parameters associated with the workload class are read and this sets the resource variable in the session or statement context.



The following table gives an overview of the properties available, see the following topic for details:

Property	Value / Unit	Default / Configuration Parameter
Priority [page 443]	0-9	Default 5
Statement Thread Limit [page 443]	1-number of logical cores	Default 0 (no limit). Can be set in `default_statement_concurrency_limit`
Statement Memory Limit [page 443]	0-Global Allocation Limit (GB)	Default can be set in `statement_memory_limit`
Total Statement Thread Limit [page 443]	1-number of logical cores	Default 0 (no limit). Can be set in `default_statement_concurrency_limit`
Total Statement Memory Limit [page 443]	0-Global Allocation Limit (GB)	Default can be set in `total_statement_memory_limit`
Statement Timeout [page 445]	Seconds	Default 0 (disabled). Can be set in `statement_timeout`
Write Transaction Lifetime [page 445]	Minutes	Default 0 (disabled). Can be set in `write_transaction_lifetime`

Property	Value / Unit	Default / Configuration Parameter
Idle Cursor Lifetime [page 445]	Minutes	Default 0 (disabled). Can be set in `idle_cursor_lifetime` (default 720 minutes)
Admission Control Reject CPU Threshold [page 446]	0-100 (percent)	Default defined in admission control parameter `reject_cpu_threshold` (default 0 - disabled)
Admission Control Reject Memory Threshold [page 446]	0-100 (percent)	Default defined in admission control parameter `reject_memory_threshold` (default 0 - disabled)
Admission Control Queue CPU Threshold [page 446]	1-100 (percent)	Default defined in admission control parameter `queue_cpu_threshold` (default 90%)
Admission Control Queue Memory Threshold [page 446]	1-100 (percent)	Default defined in admission control parameter `queue_memory_threshold` (default 90%)

The list of supported applications includes: HANA WebIDE (XS Classic), HANA Studio, ABAP applications, Lumira, and Crystal Reports. Full details of the session variables available in each supported client interface which can be passed in the connect string are given in SAP Note 2331857 *SAP HANA workload class support for SAP client applications*. Refer also to the *SAP HANA Client Interface Programming Reference*, 'Setting Session-Specific Client Information'. Session variables for workload classes are also supported for use with remote sources as described in the topic 'Session-Specific Information for Connections'.

> **Note**
>
> In a scale-out environment workload classes are created for the complete SAP HANA database and do not have to be created for each single node. However, restrictions defined in these workload classes are applied to each single node and not to the complete SAP HANA database.
>
> Note also that workload classes cannot be used on an Active/Active (read only) secondary node.

Privileges

Managing workload classes requires the ‘WORKLOAD ADMIN’ privilege. Any changes made to workload classes or mappings are effective almost immediately with only a minimal delay. In terms of the privileges of the executing user (DEFINER or INVOKER), the workload mapping is always determined on the basis of invoking user, regardless of if the user has definer or invoker privileges.

Dependencies

Users, classes and mappings are interrelated: if you drop a user in the SAP HANA database, all related workload mappings are dropped and if you drop a workload class, the related mappings are also dropped.

> **Note**
>
> Note that mappings may also have dependencies to procedures, in this case, if a procedure is dropped any related workload mappings are invalidated. However, if the procedure is restored the related mappings are also restored and can be used again. You can see the status of a mapping from the WORKLOAD_MAPPINGS.IS_VALID column (see Monitoring Workload Classes).

Related Information

Managing Peak Load (Admission Control) [page 434]
Setting Session-Specific Client Information (Client Interface Guide)
Workload Management Statements (SAP HANA SQL and System Views Reference Guide)
Managing Workload Classes (SAP HANA Cockpit)
Session-Specific Information for Connections [page 1655]
SAP Note 2331857
SAP Note 2215929
SAP HANA Predictive Analysis Library (PAL)

8.12.7.1 Properties for Workload Classes and Mappings

This topic gives details of the properties available for workload classes and workload class mappings.

This topic has the following subsections:

- Properties for Workload Classes
- Properties for Workload Class Mappings
- Hints for Workload Classes
- Other operations: importing and exporting, enabling and disabling workload classes.

Properties for Workload Classes

You can use workload classes to set values for the properties listed here. Each property also has a default value which is applied if no class can be mapped or if no other value is defined, in many cases this is a value defined in a corresponding configuration parameter.

You can enter values for these properties which include decimal fractions (such as 1.5GB) but these numbers are rounded down and the whole number value is the effective value which is applied.

Priority

Parameter	Value
PRIORITY	To support better job scheduling, this property prioritizes statements in the current execution. Numeric value of 0 (lowest priority) to 9 (highest). Default value: 5.

CPU and Memory Limits

The properties in this group can be used to apply memory limits and thread limits for individual statements or for all statements currently being executed in the working class. The following rules apply about how this group of properties can be used together:

- These properties can be used on their own; for example: set total statement memory limit = 40
- They can be mixed together (that is, memory limits combined with thread limits); for example: set total statement memory limit = 40, statement thread limit = 10)
- But the statement level properties cannot be used with the SAME aggregated property, for example the following combination would generate an error: set total statement memory limit = 40, statement memory limit = 10)

A hierarchical structure is supported so that an aggregate limit can be defined at a 'parent' level, and a statement level limit can be defined for one or more 'children'. This is described in the following topic *Hierarchies of Workload Classes*.

Parameter	Value
STATEMENT THREAD LIMIT	To avoid excessive concurrent processing due to too many small jobs this property sets a limit on the number of parallel JobWorker threads per statement and process. Numeric value between 0 (no limit) and the number of logical cores. Default: If the parameter is not set in a workload class definition then the thread limit is applied in the sequence of: user parameter value followed by the value of the `default_statement_concurrency_limit` ini file setting, but both of these are overridden by any value (including the 0 setting) which is set for this workload class parameter.

Parameter	Value
STATEMENT MEMORY LIMIT	To prevent a single statement execution from consuming too much memory this property sets a memory allocation limit in GB per statement. Numeric value between 0 (no limit) and the value of the global allocation limit. Default: If the parameter is not set in a workload class definition then the memory limit is applied in the sequence of: user parameter value followed by the value of the`statement_memory_limit` ini file setting, but both of these are overridden by any value (including the 0 setting) which is set for the statement memory limit workload class parameter. Note that if a percentage threshold value has been defined in the global `statement_memory_limit_threshold` parameter it is also effective for the workload class and may soften the effect of the statement memory limit.
TOTAL STATEMENT THREAD LIMIT	Similar to the STATEMENT THREAD LIMIT this property sets an aggregated thread limit which applies to all statements currently being executed within the workload class as a whole. Default: a value can be set in the `default_statement_concurrency_limit` ini file setting.
TOTAL STATEMENT MEMORY LIMIT	Similar to the STATEMENT MEMORY LIMIT this property sets an aggregated memory limit which applies to all statements currently being executed within the workload class as a whole. Default: a value can be set in the `total_statement_memory_limit` ini file setting.

Timeout Values

The timeout period for these properties is not exact: there may be a delay for timeout values of less than 60 minutes.

Parameter	Value
STATEMENT TIMEOUT	This property applies a time limit after which any query which has not completed execution will time out generating the following error: `ERR_API_TIMEOUT` . Numeric value in seconds. Corresponding configuration parameter: `statement_timeout` (indexserver.ini file, session section). **Note** There are three ways of applying a query timeout value: • At the command line (JDBC, SQLDBC) using a *querytimeout* instruction. • Applying a time value (in seconds) in the `statement_timeout` configuration parameter. • Applying a time value using workload classes. If multiple values have been defined using different methods, precedence rules apply: the workload class takes precedence over the ini file value, and, if a querytimeout value has been applied then the smallest (strictest) value which has been defined applies. See examples which follow. Note in particular that all three of these methods are only applied to statements submitted from client connections. Requests submitted directly from internal applications are not limited. For example, an MDS request submitted from SAP Analytics Cloud (such as Analysis for Office) is handled by XSC. Similarly, a request submitted through an XSJS service is handled by XSC and will not be limited by any of these timeout methods.
WRITE TRANSACTION LIFETIME	This property sets a time value to limit the lifetime of long-running uncommitted write transactions. Numeric value in minutes. Default value: not set (blank) Corresponding configuration parameter: `write_transaction_lifetime` (indexserver.ini file, transaction section). Not set by default (0) but can be set to a number of minutes. If a workload class property has been defined it takes precedence over the configuration file value.
IDLE CURSOR LIFETIME	This property sets a time value to limit the lifetime of long-lived cursors. Numeric value in minutes. Default value: not set (blank) Corresponding configuration parameter: `idle_cursor_lifetime` (indexserver.ini file, transaction section). Set by default to 720 minutes. If a workload class property has been defined it takes precedence over the configuration file value.

Properties for Admission Control

The properties in this group can be used to override the default settings of the admission control feature which are described in detail in the section 'Managing Peak Load (Admission Control)'.

Using these properties you can apply resource consumption thresholds to statements in a more targeted manner using any of the mapping properties supported by workload classes (such as user, database schema, application and so on).

Admission Control manages workload by denying any further SQL requests when the load on the system is equal to or exceeds a predefined threshold. Administrators can define separate thresholds for memory and CPU consumption expressed as a percentage of CPU and memory usage and also set thresholds at different load levels so that requests can be either queued or rejected.

The current load is compared firstly with the reject threshold values and if a statement is not rejected then the queuing thresholds are evaluated. A statement which exceeds a reject threshold is rejected with the SQL error 616: 'rejected by workload class configuration'. A statement which exceeds a queue threshold is queued for up to 10 minutes, after this time the statement is rejected with the SQL error 616: 'queue wait timeout exceeded'.

Note that workload class queue and reject thresholds take precedence over existing admission control thresholds. If, for example, a request is admitted by the workload class it may be queued by admission control but not rejected.

To set values for these workload class properties enter a percentage value (without a % symbol, see examples below). After making a change, the new value will be applied to all new incoming statements.

Parameter	Value
ADMISSION CONTROL REJECT CPU THRESHOLD ADMISSION CONTROL REJECT MEMORY THRESHOLD	These properties determine the levels at which statements will be rejected. Numeric value 0-100 (percentage). The value zero will always reject statements and a value of 100 will never reject statements. Default value: not set (blank) Corresponding admission control parameters: `reject_cpu_threshold` and `reject_memory_threshold,` both are set by default to 0 percent so that no requests are rejected.
ADMISSION CONTROL QUEUE CPU THRESHOLD ADMISSION CONTROL QUEUE MEMORY THRESHOLD	These properties determine the levels at which statements will be queued. Numeric value 1-100 (percentage). The value of 100 will never queue statements. Default value: not set (blank) Corresponding admission control parameters: `queue_cpu_threshold` set by default to 90 percent, `queue_memory_threshold` set by default to 0 percent (no requests are queued).

Depending on the level of user privileges, some statements may not be subject to these controls: administration requests from SESSION_ADMIN and WORKLOAD_ADMIN are always executed; the privilege check which grants the authorization to by-pass a workload class threshold is carried out when the user connection is established. Also, requests that release resources (commit, rollback, disconnect and so on) will always be executed.

Statistics about the numbers of statements admitted and rejected are available in the monitoring view M_WORKLOAD_CLASS_STATISTICS.

Examples

You can set values for one or more resource properties in a single SQL statement. This example creates a workload class called **`MyWorkloadClass`** with values for all three properties:

```
CREATE WORKLOAD CLASS "MyWorkloadClass" SET 'PRIORITY' = '3', 'STATEMENT MEMORY
LIMIT' = '2' , 'STATEMENT THREAD LIMIT' = '20'
```

This example creates a workload class and sets values for the admission control reject/queue properties. In this example the reject properties are set to 100% so nothing is rejected and therefore only the queuing values CPU 70% and memory 80% are effective:

```
create workload class "MyWorkloadClass" set 'ADMISSION CONTROL QUEUE CPU THRESHOLD'
= '70', 'ADMISSION CONTROL QUEUE MEMORY THRESHOLD' = '80', 'ADMISSION CONTROL
REJECT CPU THRESHOLD' = '100', 'ADMISSION CONTROL REJECT MEMORY THRESHOLD' = '100';
```

Examples of Precedence for Query Timeout

If multiple values have been defined using the different timeout methods available then precedence rules apply. Firstly, if a valid matching workload class value has been defined this takes precedence over the ini file setting. Secondly, if a querytimeout value has been applied then the smallest (strictest) valid value which has been defined applies. The following table shows some examples; in each case the values marked by an asterisk are the ones which apply.

QueryTimeout	25	25	25	**25***
statement_timeout (ini)	10	**10***	**10***	10 (ignored)
STATEMENT TIMEOUT (Workload class)	**20***	no match	no value	0 (disabled)

Creating a Workload Mapping

Mappings link workload classes to client sessions depending on the value of a specific client information property. The properties supported are listed here; the workload class with the greatest number of matching properties is mapped to the database client. If two workload classes have the same number of matching properties then they are matched in the prioritized order given here.

Some of the mapping properties correspond to session context variables which are available as key-value pairs that can be set in the client interface. The key names are given in the following table and values for these variables are set either by the application using the SET command (`ALTER SYSTEM ALTER SESSION SET`) or by a client API call. Details of each variable and the current value can be retrieved from the monitoring view M_SESSION_CONTEXT. A complete list of user session variables is given in the topc *Session Variables* in the *SQL Reference Guide*.

The SAP HANA application sends client context information in the 'ClientInfo object'. In some cases client applications automatically set values for these context variables: ABAP clients, for example, automatically set the APPLICATION_SOURCE; in this case the DBSL calls the SQLDBC API `setCommandInfo()` method and sets the value. See also: *Setting Session-Specific Client Information* in the *SAP HANA Developer Guide*.

Property Name	Description
OBJECT NAME	Object types PROCEDURE, PACKAGE and AREA are supported. This property only applies to procedures. This includes AFLLANG procedure which is a standard execution method to execute the application function. Example: If a workload class is matched to an object with type AREA, then it will apply the workload class definition to all AFL-LANG procedures which call application functions in the given AFL AREA. Object type PACKAGE works in a similar way. If more than one workload class is matched by the OBJECT NAME then the more specific object type has the higher priority: PROCEDURE > PACKAGE > AREA.
SCHEMA NAME	Schema name of object defined in the OBJECT NAME property.
XS APPLICATION USER NAME*	Name of the XS application business user. Session variable key name: `XS_APPLICATIONUSER`.
APPLICATION USER NAME*	Name of the application user, usually the user logged into the application. Session variable key name: `APPLICATIONUSER`.
CLIENT*	The client number is usually applied by SAP ABAP applications like SAP Business Suite / Business Warehouse. Session variable key name: `CLIENT`.
APPLICATION COMPONENT NAME*	Name of the application component. This value is used to identify sub-components of an application, such as CRM inside the SAP Business Suite. Session variable key name: `APPLICATIONCOMPONENT`.
APPLICATION COMPONENT TYPE*	This value is used to provide coarse-grained properties of the workload generated by application components. Session variable key name: `APPLICATIONCOMPONENTTYPE`.
APPLICATION NAME*	Name of the application. Session variable key name: `APPLICATION`.
USER NAME	The name of the SAP HANA database user, that is, the 'CURRENT_USER' of the session of the database the application is connected to. You can use either the user name or the user group name; if both user name and group are provided a validation error is triggered.
USERGROUP NAME	The name of the SAP HANA database user's user group. You can use either the user name or the user group name.
APPLICATION SOURCE*	For this property only the statement level variable is considered; the value for the variable must be set using the `setCommandInfo()` method. Session variable key name: `M_PREPARED_STATEMENTS.APPLICATION_SOURCE`.

The properties marked with an asterisk support the use of wildcard characters, the default wildcard character is % - see the second example here.

Example

This example creates a workload mapping called **MyWorkloadMapping** which applies the values of the **MyWorkloadClass** class to all sessions where the application name value is **HDBStudio**:

```
CREATE WORKLOAD MAPPING "MyWorkloadMapping" WORKLOAD CLASS "MyWorkloadClass" SET
'APPLICATION NAME' = 'HDBStudio';
```

This example creates the **MyWorkloadMapping1** workload mapping using the default wildcard character (does not need to be specified) to match any application name beginning with 'BW':

```
CREATE WORKLOAD MAPPING "MyWorkloadMapping1" WORKLOAD CLASS "MyWorkloadClass"
SET 'APPLICATION NAME' = 'BW%' WITH WILDCARD;
```

This example specifies a different wildcard character '_' to alter the **MyWorkloadMapping1** mapping:

```
ALTER WORKLOAD MAPPING "MyWorkloadMapping1" WORKLOAD CLASS "MyWorkloadClass" SET
'APPLICATION NAME' = 'BW_' WITH WILDCARD '_' ;
```

The default wildcard character can also be used on its own without any prefix characters. In this case ANY value which has been set will create a match; the only case which would not match is if no value has been set. This is used in the following example for APPLICATION USER NAME so that any user name that is provided will successfully match:

```
CREATE WORKLOAD MAPPING "MyWorkloadMapping1" WORKLOAD CLASS "MyWorkloadClass"
SET 'APPLICATION USER NAME' = '%' WITH WILDCARD;
```

Refer also to Workload Management Statements in the *SAP HANA SQL and System Views Reference Guide* and for details of maintaining workload classes in SAP HANA cockpit refer to *Managing Workload Classes in SAP HANA Cockpit* in the cockpit guide.

Hints for Workload Classes

To give control over workload classes at run-time a workload_class hint is available. You can use this as the query is executed to apply alternative properties compared to the ones otherwise defined. For example, workload class 'YOUR_WORKLOAD_CLASS' applies the values: PRIORITY 5, THREAD 5, MEMORY 50GB. This is then overridden by the values defined in a new class, as a hint, to apply a higher priority value, a lower thread limit and a lower memory threshold:

```
SELECT * FROM T1 WITH HINT( WORKLOAD_CLASS("MY_WORKLOAD_CLASS") );
```

This example applies more restrictive limits than those already defined and by default workload class hints can only be used in this way; the hint is ignored if any of the new values weaken the restrictions or if any values are invalid. You can change this default behavior, however, by switching the following configuration parameter in the `session_workload_management` section of the indexserver.ini file: `allow_more_resources_by_hint`. If this parameter is set to **True** then any hint can be applied.

When are Restrictions Applied?

Generally, limitations applied by configuration settings, user profile parameters and workload class settings are applied as queries are compiled. For workload classes, you can see this by querying the

M_EXPENSIVE_STATEMENTS monitoring view to see the operation value (COMPILE, SELECT, FETCH and so on) when the workload class was applied to a statement string. This is not true, however, if the workload class mapping was overridden by a hint, and also does not apply to the OBJECT NAME and SCHEMA NAME workload class properties. In these cases, the statement can only be evaluated after compiling.

Importing and Exporting Class Details

If you need to import and export workload classes, the normal SQL command for IMPORT will not work because workload classes do not belong to a schema (you cannot import into the SYS schema because it is maintained by the system). A script is available to support this functionality if you need to import a class. The script is typically shipped in the `exe/python_support` directory.

To use this you must first export the monitor views SYS.WORKLOAD_CLASSES and SYS.WORKLOAD_MAPPINGS to text (csv) format. You can then use the script to reimport the class:

1. Execute SQL EXPORT command:
   ```
   EXPORT SYS.WORKLOAD_CLASSES, SYS.WORKLOAD_MAPPINGS AS CSV INTO '<PATH>' WITH REPLACE
   ```
2. Load CSV files using python script `importWorkloadClass.py` specifiying the host as a parameter:
   ```
   python importWorkloadClass.py --host='<host>' --SID='<SID>' --user='<DB-user>' --password='<PW>' <PATH>
   ```

Disabling and Enabling Workload Classes

After creating one or more workload classes it is also possible to disable them. This may be necessary, for example, for testing purposes.

1. Disable or enable a single named class:
   ```
   ALTER WORKLOAD CLASS '<Class Name>' {enable | disable}
   ```
2. Disable all workload classes using the 'all' switch:
   ```
   ALTER WORKLOAD CLASS all {enable | disable}
   ```

Related Information

Hierarchies of Workload Classes [page 451]
Setting a Memory Limit for SQL Statements [page 431]
Managing Peak Load (Admission Control) [page 434]

Links to other documents

Setting Session-Specific Client Information (Client Interface Guide)
Managing Workload Classes (SAP HANA Cockpit)
Workload Management Statements (SAP HANA SQL and System Views Reference Guide)
SET [SESSION] Statement (Session Management) (SAP HANA SQL and System Views Reference Guide)
Session Variables (SAP HANA SQL and System Views Reference Guide)

M_SESSION_CONTEXT System View (SAP HANA SQL and System Views Reference Guide)

8.12.7.2 Hierarchies of Workload Classes

A hierarchical relationship between workload classes can be established so that values for memory limits and thread limits can be inherited by other workload classes.

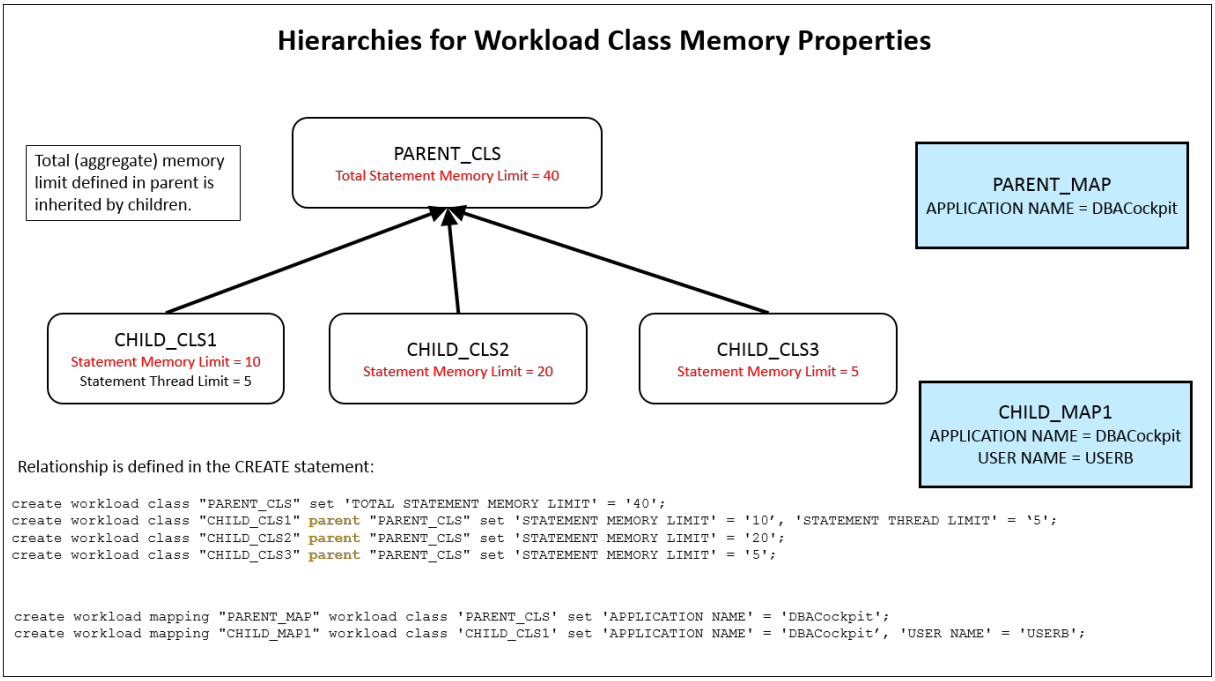
The workload class properties to manage memory and thread limits can be defined either at the statement level or as total aggregated values for all workload classes:

- Statement Thread Limit - thread limit applying to individual statements
- Statement Memory Limit - memory limit applying to individual statements
- Total Statement Thread Limit - when used as a parent in a hierarchy then the limit applies to all statements mapped to any workload class in the hierarchy.
- Total Statement Memory Limit - when used as a parent in a hierarchy then the limit applies to all statements mapped to any workload class in the hierarchy.

For these properties a hierarchical structure is supported so that an aggregate limit can be defined at a 'parent' level, and a statement level limit can be defined for one or more 'children'. The total aggregate limit is then inherited by the children.

This is illustrated here in an example showing the pair of memory limit properties and corresponding mappings. The `Statement Thread Limit` and `Total Statement Thread Limit` properties can be used in exactly the same way. The scope of the supported relationship structure is as illustrated here:

- A child workload class can only have one single parent and cannot also be a parent.
- In a hierarchical relationship the aggregate properties are defined in the parent and cannot be defined in the child class. The corresponding individual limit must be defined in the child.
- Additional properties may be defined in the child classes (see CHILD_CLS1 in the illustration) and other properties may also be defined in the parent workload class but these are not inherited.
- A general limitation which applies to all workload classes is that you can combine the thread and memory statement limits in the same workload class, but you cannot combine the total statement limit and the single statement limit of the same property in one workload class (see the second example below).



If the mapping matches with the parent class only the properties defined for that class are applied; for example, requests from a user named USERA working in DBACockpit could be mapped to PARENT_CLS in the illustration. If the mapping matches a user with a child class then values from the parent are inherited: for example, requests from a user named USERB working in DBACockpit could be mapped with CHILD_CLS1 in the illustration; in that case the Total Memory Limit of 40 for all statements currently being executed within the hierarchy applies and a limit of 10 for any individual statement.

You can define a parent-child relationship when the child workload class is created by including a value for the <INHERITANCE> clause which uses the keyword 'parent' and the name of the parent workload class, for example:

```
create workload class "CHILD_CLS1" parent "PARENT_CLS" set 'STATEMENT THREAD
LIMIT' = '10';
```

You can remove a relationship by setting the inheritance value to null, for example:

```
alter workload class "CHILD_CLASS" parent null;
```

Refer to the *SAP HANA SQL Reference Guide* for full details.

Examples

The statements used in the above illustration are as follows:

```
create workload class "PARENT_CLS" set 'TOTAL STATEMENT MEMORY LIMIT' = '40';
create workload class "CHILD_CLS1" parent "PARENT_CLS" set 'STATEMENT MEMORY
LIMIT' = '10', 'STATEMENT THREAD LIMIT' = '5';
create workload class "CHILD_CLS2" parent "PARENT_CLS" set 'STATEMENT MEMORY
LIMIT' = '20';
create workload class "CHILD_CLS3" parent "PARENT_CLS" set 'STATEMENT MEMORY
LIMIT' = '5';
```

This next example shows how properties can be combined in workload classes which are not hierarchical. Class2 sets a limit for memory at class level and a limit for threads at statement level, that is, different

properties are combined which have different scopes. In this case (and also similar cases with hierarchical workload classes) it would not be possible to use both TOTAL STATEMENT MEMORY LIMIT and STATEMENT MEMORY LIMIT properties in the same class:

```
create workload class "Class1" set 'TOTAL STATEMENT MEMORY LIMIT' = '40';
create workload class "Class2" set 'TOTAL STATEMENT MEMORY LIMIT' = '40',
'STATEMENT THREAD LIMIT' = '40';
```

Monitoring Views

You can see the workload class relationships in the view WORKLOAD_CLASSES (column PARENT_WORKLOAD_CLASS_NAME).

Statement limit details can be seen in the relevant monitoring views: M_{PREPARED|ACTIVE|EXPENSIVE}_STATEMENTS (columns TOTAL_STATEMENT_MEMORY_LIMIT, TOTAL_STATEMENT_THREAD_LIMIT, STATEMENT_MEMORY_LIMIT and STATEMENT_THREAD_LIMIT)

Related Information

Properties for Workload Classes and Mappings [page 442]
CREATE WORKLOAD CLASS Statement (Workload Management)

8.12.7.3 Monitoring Views for Workload Classes

You can use system views to monitor, export, disable and view details of workload classes.

The following system views allow you to monitor workload classes and workload mappings:

- WORKLOAD_CLASSES
- WORKLOAD_MAPPINGS

The monitoring view M_WORKLOAD_CLASS_STATISTICS has details of the resources that have been consumed by statements that belong to a specified workload class. This includes statistics about the numbers of statements admitted, queued, and rejected.

In these system views the field `WORKLOAD_CLASS_NAME` shows the effective workload class used for the last execution of that statement:

- M_ACTIVE_STATEMENTS
- M_PREPARED_STATEMENTS
- M_EXPENSIVE_STATEMENTS (enable_tracking and memory_tracking must first be enabled in the global.ini file for this view)
- M_CONNECTIONS
- M_SERVICE_THREADS and M_SERVICE_THREAD_SAMPLES

If no workload class is applied then these views display the pseudo-workload class value "_SYS_DEFAULT".

You can also use queries such as the following examples to read data from these views:

Sample Code

```
-- get overview of available workload classes and workload class mappings
select wc.*, wm.workload_mapping_name, user_name, application_user_name,
application_name, client
from workload_classes wc, workload_mappings wm where wc.workload_class_name =
wm.workload_class_name;
```

Sample Code

```
-- get sum of used memory of all prepared statements grouped by workload
class which are executed in the last 10 minutes; requries memory tracking
select workload_class_name, sum(memory_size), count(*) statement_count,
       count(distinct connection_id) as distinct_connection_count,
       count(distinct application_name) as distinct_application_count,
       count(distinct app_user) as distinct_applicationuser_count
from sys.m_expensive_statements
where add_seconds(start_time, 600) >= now() and memory_size >= 0
group by workload_class_name;
```

Sample Code

```
-- get information about priorities assigned to prepared statements executed
in the last 10 minutes
select workload_class_name, min(priority) min_priority, max(priority)
max_priority, count(*) statement_count,
       count(distinct connection_id) as distinct_connection_count,
       count(distinct application_name) as distinct_application_count,
       count(distinct app_user) as distinct_applicationuser_count
from sys.m_expensive_statements
where add_seconds(start_time, 600) >= now()
group by workload_class_name;
```

Sample Code

```
-- collect workload related information for active statements
with job_count_per_statement as (select statement_id, count(0) num_active_jobs
   from sys.m_service_threads
   where statement_hash <> '' and is_active = 'TRUE'
   group by statement_id, statement_hash)
select s.statement_id, s.statement_string, s.memory_size, s.duration_microsec,
       s.application_source, s.application_name, s.app_user,
s.db_user, s.priority, s.statement_thread_limit, s.statement_memory_limit,
s.workload_class_name, st.num_active_jobs
from sys.m_expensive_statements s, job_count_per_statement st
where st.statement_id = s.statement_id;
```

Sample Code

```
-- collect workload related information for active statements
select s.statement_id, s.statement_string, s.memory_size, s.cpu_time,
       s.application_source, s.application_name, s.app_user, s.db_user,
s.workload_class_name
from sys.m_expensive_statements s;
```

Sample Code

```
-- get information from system views
select * from sys.m_prepared_statements;
select * from sys.m_active_statements;
select * from sys.m_expensive_statements;
select * from m_service_threads;
select * from m_service_thread_samples;
```

8.12.7.4 Workload Class Examples

Here we give examples to show how the workload management features interact together.

Workload class settings override other ini file configuration settings which have been applied and also override user parameter settings which have been applied by the SQL command ALTER USER, however, workload class settings only apply for the duration of the current session, whereas changes applied to the user persist. More detailed examples of this and also how hints are applied to invoke a workload class at run-time are given in this section. The SAP HANA SQL Reference Guide also includes examples of precedence including code samples.

Note

For the examples shown in this section where memory limits are defined, the resource tracking parameters *enable_tracking* and *memory_tracking* in the global.ini file must be set to **true**.

Precedence of Workload Management Properties

In general, workload management settings are applied in this sequence: workload class settings, followed by user parameters, followed by other ini file settings.

Precedence of Workload Class Values

The following table shows a scenario where a statement memory limit setting has been applied in the global ini file, restrictions have been applied at the level of the user profile and also in a workload class. In this basic example, if the workload class is mapped to a statement then the workload class settings take precedence and so for the duration of the current session the following values are effective: thread limit 10, memory limit 50 GB, priority 7.

Basic Scenario: Precedence of Workload Management Properties

	Global.ini	User Profile	Workload Class
Concurrency (job executor threads)	N/A	Thread limit 5	Thread limit 10
Statement Memory limit	50 GB	30 GB	50 GB
Priority	N/A	Priority 5	Priority 7

The following two variations on this scenario show the chain of precedence:

1. If the workload class thread limit is undefined the user profile value is effective: 5
2. If the workload class memory limit is undefined and the user profile memory limit is undefined then the global value is effective: 50 GB
3. If the workload class memory limit is 0 then there is no limit on memory allocation per statement.

Total Statement Values for Workload Classes

For workload classes using the aggregated properties (TOTAL STATEMENT THREAD LIMIT and TOTAL STATEMENT MEMORY LIMIT) the same principle of precedence applies as shown above. If a workload class with aggregated properties is used, memory and concurrency limits for single statements from ini parameters or user profile are not applied for statements running inside this class.

Using Hints to specify a Workload Class

If a workload class is specified with an SQL statement as a hint at run-time then the settings of the matching hint class take precedence. In the following example, two workload classes are matched to the current execution request. The limitations specified by the hint class are successfully applied: thread limit 4, statement memory limit 30 GB, and priority 4.

Example showing use of hints

	Global.ini	User Profile	Workload Class	Hint Workload Class
Concurrency	N/A	Thread limit 5	Thread limit 5	Thread limit 4
Statement Memory limit	50 GB	30 GB	50 GB	30 GB
Priority	N/A	Priority 5	Priority 5	Priority 4

The following two variations on this scenario show the effects of additional conditions:

1. If, for example, the Concurrency (Thread limit) value of the hint workload class is left undefined then all values of the hint workload class would be ignored. In the above example, the workload class settings take precedence since hint workload class is ignored. And so for the duration of the current session the following values are effective: thread limit 5, memory limit 50 GB, priority 5. If the workload class memory limit is undefined the user profile value is effective: 30.
2. By default, the hint can only be used to specify more restrictive values; in this scenario if the hint workload class memory limit was 70 GB then all values of the hint workload class would be ignored. This behavior can be reversed by setting the ini file parameter `allow_more_resources_by_hint` to True; in that case the more generous limit of 70 GB would then be applied.

Related Information

Managing Workload with Workload Classes [page 439]
SAP HANA SQL Reference Guide for SAP HANA Platform

8.12.8 Example Workload Management Scenarios

Here, we give a number of scenarios to illustrate how workload management settings can be applied for systems of different sizes, different workload types (analytics and transactional) and different usage scenarios (a system which is optimized for robustness as opposed to high performance).

> **Note**
>
> All settings are tuning parameters and must be tested and validated before being used in production. See the process description in *Understand Your Workload Management*.

System Details

The scenarios which follow are based on the system specifications given here: firstly describing hardware resources and secondly the workload types which the system is expected to handle.

System Types 1: Small and Large Hardware Configurations

	Small (maximum 4 sockets)	**Large**
Sockets (processors)	2	16
Physical cores per socket	8	2 x 15 = 30
Logical cores (threads)	32	16 x 30 = 480
Memory	64 GB	256GB

Note that related to memory resources, the setting for global memory allocation limit is very important. By default, it is calculated as 90% of the first 64 GB of available physical memory on the host plus 97% of each further GB; or, in the case of small physical memory, physical memory minus 1 GB.

Secondly, we give details of two contrasting types of workload: pure analytics processing and mixed transactions, to show how the system can be configured to handle these different situations:

System Types 2: Workload and Processing Types

	Analytics Workload	**Mixed Workload**
Installed Applications	SAP Business Warehouse (or similar analytic application)	SAP Business Suite (OLTP) and Smart Business Apps (OLAP)
Workload type	Only OLAP.	Mixed OLAP and OLTP.
Processing characteristics	CPU and Memory intensive - long transaction times (+ 1 second)	Both applications have short-running statements (milliseconds or microseconds) where response time is critical as well as long-running CPU and memory-intensive OLAP statements.

	Analytics Workload	Mixed Workload
Data	Bulk loading	
Concurrent queries, Concurrent users.	Few (10-100)	Many (> 1000)

Scenario 1: Mixed Workload (OLAP and OLTP) Optimized for Robustness

In the first scenario, the focus is on achieving robust statement execution time and high availability of the system.

Small System with Mixed Workload Optimized for Robustness

Tuning Option	Example Setting
`statement_memory_limit`	Start testing with this parameter set to 25% of the global allocation limit.
`default_statement_concurrency_limit`	Assign 33% of the logical cores on the system.
`max_concurrency_hint`	For systems with at least 80 logical cores consider `max_concurrency_hint` equal to the number of logical cores per socket.
Workload Classes	Fine-grained control is possible with workload classes. In a mixed workload scenario, the OLTP load could be configured with a higher priority than OLAP.

Scenario 2: Mixed Workload (OLAP and OLTP) Optimized for Performance

For the second scenario we take exactly the same system as scenario 1 but we optimize for performance instead of robustness by relaxing some of the settings:

Small System with Mixed Workload Optimized for Performance

Tuning Option	Example Setting
`max_concurrency_hint`	For systems with at least 80 logical cores consider `max_concurrency_hint` equal to the number of logical cores per socket.
Workload Classes	Consider assigning higher priority to OLTP statements than to OLAP statements by using workload classes based on the application.

Scenario 3: Analytics

Pure OLAP scenarios tend to be more performance oriented, hence, we remove the limitations on concurrency here (leading to a best-effort approach). On the other hand, to avoid out-of-memory situations, we keep the memory limits.

Small Analytic System

Tuning Option	Example Setting
`statement_memory_limit`	Start testing with this parameter set to 25% of the global allocation limit.
`statement_memory_limit_threshold`	Start testing with this parameter set to 50% of the global allocation limit.

Check M_EXPENSIVE_STATEMENTS.MEMORY_SIZE for the typical memory usage of statements.

Scenario 4: Large System with at least 4 Sockets - All Workload Types

Large systems usually need to handle many concurrent statements therefore it is usually reasonable to limit the concurrency of statements; but also, this may help to avoid cases where HANA over-parallelizes. Less parallelism per statement on these large systems tends to have better performance because statements run on one single NUMA node and we tend to avoid cross NUMA node communication.

Large System, All Workload Types

Tuning Option	Example Setting
`statement_memory_limit`	Start testing with this parameter set to 25% of the global allocation limit.
`default_statement_concurrency_limit`	Assign 33% of the logical cores on the system.
`max_concurrency`	From SPS12, the default setting should give good performance in most cases. If necessary, consider reducing the value of this parameter. Start testing with the parameter set to 50% of the available logical cores.
`max_concurrency_hint`	Consider setting `max_concurrency_hint` equal to the number of logical cores per socket.
Workload Classes	In mixed scenarios fine-grained control is possible with workload classes.

Related Information

Understand your Workload [page 418]
Analyzing System Performance [page 419]

8.13 Getting Support

Many support resources are available to help you work with SAP HANA. Various tools can help you to analyze, diagnose, and resolve problems.

Support Resources

The SAP Support Portal is the starting point for all support questions; an SAP user ID (S-user) is required for access: https://support.sap.com. From here you have access to all resources including support incidents, software downloads, documentation, SAP Notes, Knowledge Base Articles, and options for contacting Support. The Support portal also gives access to SAP for Me and the SAP Community.

Guided Answers

Guided Answers is an interactive online support tool to help you to diagnose and solve problems using decision trees. It covers many SAP products including SAP HANA and offers a set of step-by-step problem-solving documents each one designed to address a specific issue. Guided Answers is available in the SAP Support portal at the following address: https://ga.support.sap.com/dtp/viewer/

> **Note**
>
> Two Guided Answers trees may be particularly helpful in making an initial assessment of any problem:
>
> - The *SAP HANA Component Guide* tree will help you to troubleshoot an issue and determine the correct component reference (such as *HAN-DB-BAC* for SAP HANA Backup and Recovery). If you need to contact Support the component ID will enable Support analysts to immediately identify where the problem is located. SAP HANA Component Guide
> - The *SAP HANA Troubleshooting and Problem Categorization* tree is a general high-level troubleshooting tree for SAP HANA. SAP HANA Troubleshooting

YouTube

SAP Support has a YouTube channel with a wide range of materials in video format. Separate channels are available for example, for SAP Products and Services, SAP Community and so on: Services and Support from SAP

Contacting Support

For help with questions that are not covered by these resources, you may need to open a support incident.

The Support team may require you to prepare analysis documents and provide diagnostic information (such as trace files) in order to efficiently analyze your scenario. See Diagnostic Files and Logs [page 461] and Traces [page 462].

Incidents, Calls, and Chat Sessions

In addition to incidents, the *Expert Chat* and *Schedule an Expert* support options may be preferable alternatives. A chat session can be either a simple text chat within your browser or you can share screens

with an expert to demonstrate a specific issue or question. If you can plan your support call in advance there is also the option of scheduling a 30-minute call with a support engineer. Full details of these options are available in the *Get Support* area of the portal: Product Support.

See also SAP Notes 2570790 and 2392095.

Related Information

SAP Note 2570790
SAP Note 2392095
Diagnostic Files and Logs [page 461]
Traces [page 462]

8.13.1 Diagnostic Files and Logs

Diagnosis files include log and trace files, as well as a mixture of other diagnosis, error, and information files. In the event of problems with the SAP HANA database, you can check these diagnosis files for errors. Traces generally need to be explicitly enabled and configured.

Location of Diagnosis Files

Diagnosis files of the system database are stored at the following default location on the SAP HANA server: `/usr/sap/<SID>/HDB<instance>/<host>/trace`.

Trace files of tenant databases are stored in a sub-directory named `DB_<database_name>`.

> → Recommendation
>
> Monitor disk space that is used for diagnosis files and delete files that are no longer needed.

If the local secure store (LSS) is active, it writes trace and dump files to a folder in its shared file system (`/usr/sap/<SID>/lss/shared/data/trace`). This folder is accessible only by the user `<sid>crypt`. For more information, see the *SAP HANA Security Guide*.

Tools for Viewing Diagnosis Files

You can view diagnosis files in SAP HANA Database Explorer where files are grouped on the basis of host and service in the catalog browser. You can access the files by right-clicking one of the folders.

You can also view diagnosis files in SAP HANA Studio on the Diagnosis Files tab of the Administration editor.

Related Information

Traces [page 462]
Monitoring Disk Space
Tenant Databases [page 17]
LSS Trace and Dump Files
View Diagnostic Files in the SAP HANA Database Explorer (SAP HANA Database Explorer)
View Diagnosis Files in SAP HANA Studio

8.13.2 Traces

SAP HANA provides various traces for obtaining detailed information about the actions of the database system for troubleshooting and error analysis.

You can configure tracing in either SAP HANA database explorer or SAP HANA Studio. System privilege TRACE ADMIN is required. See Related Links below to jump to the relevant topics.

The following table gives an overview of the main traces available in SAP HANA:

Configure or Run...	To...	Using...
Database trace	Understand activity in the components of the SAP HANA database in general, or for a specific user or client operation **Note** Database tracing is always active. Information about error situations is always recorded in alert trace files.	SAP HANA database explorer SAP HANA studio. See also KBA 2380176 - FAQ: SAP HANA Database Trace.
SQL trace	Collect information about all SQL statements executed on the index server	SAP HANA database explorer SAP HANA studio
Expensive statements	Record information about individual SQL statements whose execution time exceeded a configured threshold	SAP HANA cockpit SAP HANA database explorer SAP HANA studio

Configure or Run...	To...	Using...
Plan trace	Visualize the execution plans of SQL SELECT statements for in-depth query performance analysis **Note** The plan trace is part of the SAP HANA tools for query execution analysis: the Web-based **SQL analyzer**, which is integrated into the SAP HANA cockpit and SAP Web IDE for SAP HANA, and the **SAP HANA PlanViz** perspective of the SAP HANA studio.	SQL analyzer SAP HANA PlanViz perspective of the SAP HANA studio
Deadlock Detection	Create a *waitgraph* file that contains thread information about deadlocks.	SAP HANA database explorer (SQL Statement CREATE WAITGRAPH)
Performance trace	Record performance indicators of individual query processing steps in the database kernel for detailed performance analysis by SAP Support	SAP HANA studio
Kernel profiler	Generate a profiler trace for detailed performance analysis by SAP Support	SAP HANA database explorer SAP HANA studio (SQL Statement START KERNEL PROFILER)
Full system information dump	Create a runtime dump if requested by SAP support to understand the behavior of SAP HANA in a problem situation	SAP HANA cockpit or SAP HANA studio (SQL statement CREATE RUNTIMEDUMP)
Plan decision recorder	Record detailed SQL optimizer information about recompiled plans by logging it in a dedicated trace file.	SAP HANA database explorer (SQL statement ALTER SYSTEM ALTER CONFIGURATION). The plan decision recorder is disabled by default.
SAP Web Dispatcher HTTP traces	Analyze HTTP requests	SAP HANA cockpit or SAP HANA studio

Related Information

Links to following topics in this document:

Database Trace (Basic, User-Specific, and End-to-End) [page 464]
SQL Trace [page 469]
Performance Trace [page 473]
Expensive Statements Trace [page 475]
Deadlock Detection Using SQL [page 479]
Kernel Profiler [page 479]
Traces and Trace Configuration for Internal Web Dispatcher [page 484]
Collecting Diagnosis Information for SAP Support [page 486]

Links to other documents:
Configure Traces in SAP HANA Studio
Configure Tracing in the SAP HANA Database Explorer
Analyzing SQL Execution with the Plan Visualizer (Troubleshooting Guide)
Monitor and Analyze Statements with Plan Trace (SQL Analyzer) (SAP HANA Cockpit)
SAP Note 2380176

8.13.2.1 Database Trace (Basic, User-Specific, and End-to-End)

The database trace records information about activity in the components of the SAP HANA database. You can use this information to analyze performance and to diagnose and debug errors.

Database Trace Files

Each service of the SAP HANA database writes to its own trace file. The file names follow the default naming convention:

```
<service>_<host>.<port_number>.<3_digit_file_counter>.trc
```

> Example
>
> `indexserver_veadm009.34203.000.trc`

Information recorded by the `alert` trace component of the database trace is written to a separate file to ensure that critical information is easy to find. The file names follow the default naming convention:

```
<service>_alert_<host>.trc
```

> Note
>
> The trace files generated for the Web Dispatcher service are different. For more information, see *Trace Configuration for Internal Web Dispatcher*.

You access database trace files with other diagnosis files.

Basic Database Trace Configuration

Database tracing is always active. This means that information about error situations is always recorded. However, for more detailed analysis of a specific problem or component, you may need to configure a certain trace component to a trace level higher than the default value. For example, the `backup` trace component records information about backup operations, the `authorization` component records information about authorization operations, and so on.

→ Tip

For more information about which trace component to use for which situation, see SAP Note 2380176.

Changing the trace level of a trace component

If a trace component is available in all services, the trace level can be configured for all services at once. It is also possible to configure the trace level of a component individually for a specific service. The trace level of a component configured at service level overrides the trace level configured for all services. Some components are only available in a particular service and cannot therefore be changed globally.

Example

You change the trace level of the `memory` component to ERROR for all services and for the indexserver service, you change it to WARNING. This means that the `memory` component of the indexserver service will trace up to level WARNING and the `memory` component of all other services will trace to the level ERROR.

You can configure the trace levels of database trace components in the SAP HANA database explorer or the SAP HANA studio. Alternatively, you can modify the parameters in the `trace` section of the `global.ini` configuration file (for all services) or service-specific files such as `indexserver.ini`. The individual parameters correspond to trace components and the parameter value is the trace level.

Example

Use the following statement to set the trace level of the `authentication` component to DEBUG:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET ('trace',
'authentication') = 'DEBUG' WITH RECONFIGURE
```

Note

In the SAP HANA database explorer and SAP HANA studio, a trace level that has been inherited from the global or *ALL SERVICES* configuration (either the default or system configuration) is shown in brackets.

What trace levels are possible?

The higher the trace level, the more detailed the information recorded by the trace. The following trace levels exist:

- NONE (0)
- FATAL (1)
- ERROR (2)
- WARNING (3)
- INFO (4)
- DEBUG (5)

Note

Even if you select trace level NONE, information about error situations is still recorded.

Alert Trace Component

The `alert` trace component is used to ensure that critical information is easy to find by duplicating high-priority trace messages to separate alert trace files. By default, trace messages with trace level ERROR and higher are written to the alert trace file.

> ### Example
>
> If the `alert` component is set to WARNING, the alert trace contains messages created with trace levels WARNING, FATAL and ERROR.

User-Specific and End-to-End Database Traces

User-specific and end-to-end traces extend the configured database trace by allowing you to change the trace level of components in the context of a particular user or end-to-end analysis. The trace levels configured for components in these contexts override those configured in the database trace.

> ### Example
>
> In the database trace, you changed the trace level of the `memory` component to ERROR for all services, and for the indexserver service you changed it to WARNING. Now, you create a user-specific trace for User1 and increase the trace level for all services to WARNING. For the indexserver service, you increase it to DEBUG. This results in the following tracing behavior for the `memory` component:
>
> - For all users except User1, all services except the indexserver will trace to ERROR
> - For all users except User1, the indexserver will trace to WARNING
> - For User1, all services except the indexserver will trace to WARNING
> - For User1, the indexserver will trace to DEBUG

If you are connected to a database of version HANA 2.0 SPS 03, then you can use the *Statement Hash* option when configuring user-specific tracing in the SAP HANA database explorer. This option allows you to filter your tracing according to specific SQL statements.

To run tracing on a specific SQL statement, you must first find the unique identifier (statement hash) for that statement and enter it in the *Statement Hash* field when configuring your user-specific trace session.

For example, if you want to see trace information when running the statement `SELECT * FROM DUMMY;`, then execute the following statements to find the its statement hash:

```
SELECT * FROM DUMMY;
SELECT statement_string, statement_hash FROM m_sql_plan_cache WHERE
statement_string = 'SELECT * FROM DUMMY;';
```

Enter the result of the second statement into the *Statement Hash* field of the *User-Specific Trace* configuration dialog to include tracing for the `SELECT * FROM DUMMY;` statement in your user-specific trace.

End-to-End Traces

End-to-end traces are triggered by applications outside of the SAP HANA database. The default trace levels for the SAP HANA database components are normally sufficient and do not need to be changed. For more

information about end-to-end analysis in your landscape, see the following link 'End-to-End Analysis Overview' in the SAP Library for Solution Manager .

Further Information

An SAP Community blog *User-specific Database Traces filtered by connection_id and statement_hash* gives further details of this feature.

Related Information

Diagnostic Files and Logs [page 461]
Traces and Trace Configuration for Internal Web Dispatcher [page 484]
End-to-End Analysis Overview
SAP Note 2380176
SAP Community Blog: User-specific Database Traces filtered by connection_id and statement_hash

8.13.2.1.1 Database Trace Configuration in Tenant Databases

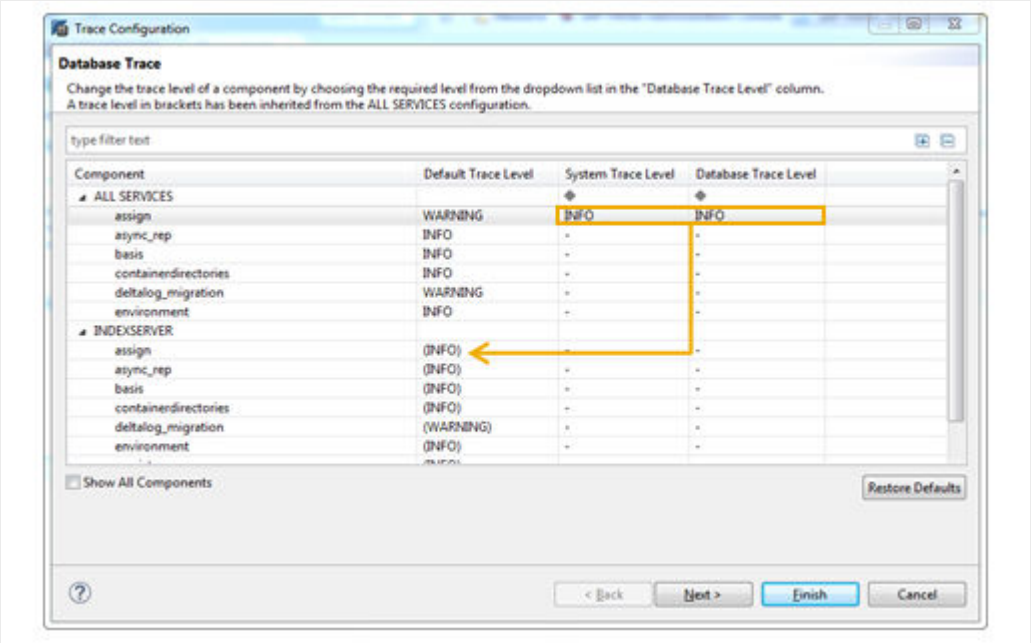
Tenant databases inherit the database trace level configured in the system database unless you change the trace level in the tenant database.

The trace level of trace components in a tenant database is inherited from the system database as the default value. If you want to configure a different trace level for a particular component in the tenant database, either globally for all services or for a specific service, you can do so by changing the trace level of the relevant component.

> **Note**
>
> In the SAP HANA studio, the trace level of a component will also be displayed as the system trace level, and you cannot change the system trace level. This is because from the perspective of the tenant database, the database and the system are effectively the same. The true system trace level (that is, the value configured in the system database) appears as the default trace level for the tenant database.

> **Note**
>
> A trace level that has been inherited from the ALL SERVICES configuration (either the default or database configuration) is shown in brackets.

> **Example**
>
> In the following example, you can see that the default trace level for the component `assign` inherited from the system database is WARNING. The trace level was changed to INFO for all services in this database. The indexserver service therefore inherits this trace level. All the other components keep the default configuration.



Database Trace Configuration in a Tenant Database (Studio View)

8.13.2.1.2 Configure Database Trace File Rotation

Trace file rotation prevents trace files from growing indefinitely by limiting the size and number of trace files. You can configure trace file rotation globally for all services in the database and for individual services.

Prerequisites

You have the system privilege INIFILE ADMIN.

Context

You configure trace file rotation in the `global.ini` configuration file (all services) or the service-specific file (for example, indexserver.ini).

Procedure

Depending on whether you are configuring trace file rotation for all system services or for an individual service, proceed as follows:

Option	Description
All services	In the `trace` section of the `global.ini` file, configure the following parameters: • `maxfiles` by specifying the maximum number of trace files that may exist • `maxfilesize` by specifying in bytes the maximum size an individual trace file may reach **Note** The default configuration for trace file rotation in the `global.ini` file is `maxfiles=10` and `maxfilesize=10000000`.
Individual service	In the `trace` section of the `global.ini` file, configure the following parameters. If there is no trace section, add one. • `maxfiles` by specifying the maximum number of trace files that may exist • `maxfilesize` by specifying in bytes the maximum size an individual trace file may reach in bytes **Note** If these two parameters do not exist in the `trace` section or if you created a new `trace` section, create them.

Results

When a trace file reaches the specified maximum file size, it is closed, and a new file created. When the specified maximum number of files is reached, the next time a new file is created, the first file is deleted, and so on.

> **Note**
>
> The system checks the size and number of diagnosis files regularly. The threshold values for these checks (check 50 and 51) should be in line with the configured trace file rotation.

Related Information

Configuring SAP HANA System Properties (INI Files) [page 126]
Configure Traces in SAP HANA Studio
Configure Check Thresholds

8.13.2.2 SQL Trace

The SQL trace collects information about all SQL statements executed on the index server (tenant database) or name sever (system database) and saves it in a trace file for further analysis. The SQL trace is inactive by default.

Information collected by the SQL trace includes overall execution time of each statement, the number of records affected, potential errors (for example, unique constraint violations) that were reported, the database

connection being used, and so on. The SQL trace is a good starting point for understanding executed statements and their potential effect on the overall application and system performance, as well as for identifying potential performance bottlenecks at statement level.

SQL Trace Files

SQL trace information is saved as an executable python program (by default `sqltrace_<...>.py`), which can be used to replay the traced database operations. You can also use the SQL Trace Analyzer tool to automate the analysis of the file.

Enabling and Configuring the SQL Trace

You can enable and configure the SQL trace in the SAP HANA database explorer or SAP HANA studio. Alternatively, you can modify the parameters in the`sqltrace` section of the `indexserver.ini` (tenant database) or `nameserver.ini` (system database).

> ### Example
>
> Use the following statement to enable the SQL trace:
>
> ```
> ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET ('sqltrace',
> 'trace') = 'on' WITH RECONFIGURE
> ```

> ### → Recommendation
>
> Do not leave the SQL trace enabled all the time as writing trace files consumes storage space on the disk and can impact database performance significantly.

Trace Levels

You can set the level for the SQL trace by changing the value of the configuration parameter `[sqltrace] level` in the `indexserver.ini` file (tenant database) or `nameserver.ini` file (system database). Trace information includes details such as executed timestamp, thread ID, connection ID, and statement ID.

Trace Level	Description
NORMAL	All statements that have finished successfully are traced.
ERROR	All statements that returned errors are traced.
ERROR_ROLLBACK	All statements that are rolled back are traced.
ALL	All statements including status of normal, error, and rollback are traced.

Trace Level	Description
ALL_WITH_RESULTS	In addition to the trace generated with trace level ALL, the result returned by select statements is also included in the trace file. **Note** An SQL trace that includes results can quickly become very large. **Caution** An SQL trace that includes results may expose security-relevant data, for example, query result sets.

Trace Details

You can configure trace detail information by setting the parameter `[sqltrace] details`. You can select one or more categories of information to include in the trace, for example: '`basic,resource_consumption`'. Possible values are listed in the following table. Note that for resource consumption information (this is also included in the 'all' option) the following two parameters in the `global.ini` file `[resource_tracking]` section, must be set to 'on':

- enable_tracking
- memory_tracking

You may also wish to limit the maximum memory allocation per statement by setting a value for the `[memorymanager] statement_memory_limit` parameter in the `global.ini` file. Set this to 5, for example, to apply a limit of 5GB.

Trace Details	Description
basic	Connection information and statement information (default)
all	Include all comments of connection and statement
user_variables	User-defined variables in the session context
statement	Statement information such as executed timestamp, thread ID, connection ID, statement ID, statement hash and duration
session_variables	System-defined variables in the session context
resource_consumption	Statement resource consumption information such as local (+remote, if available) cpu-time and memory-size.
passport	Decoded passport contents
connection	Connection information such as session ID, transaction ID, client PID, client IP, user name, schema name, and session variable:value pairs
"empty"	Trace without these comments

Additional Configuration Options

Option	Configuration Parameter	Default	Description
Trace file name	`tracefile`	`sqltrace_$HOST_$ {PORT}_$ {COUNT:3}. py`	User-specific name for the trace file If you do not enter a user-specific file name, the file name is generated according to the following default pattern: `DB_<dbname>/sqltrace_$HOST_$ {PORT}_${COUNT:3}.py`, where: • DB_<dbname> is the sub-directory where the trace file is written if you are running on a tenant database • $HOST is the host name of the service (for example, indexserver) • $PORT is the port number of the service • $COUNT:3 is an automatically generated 3-digit number starting with 000 that increments by 1 and serves as a file counter when several files are created.

User, application, object, and statement filters	`user`
	`application_user`
	`application`
	`object`
	`statement_type`
		Empty string	Filters to restrict traced statements to those of particular database or application users and applications, as well as to certain statement types and specific objects (tables, views, procedures).
			All statements matching the filter criteria are recorded and saved to the specified trace file.
			For `user`, `application_user`, and `application` the use of wildcards is supported (see following subsection *Using Wildcards*).

Flush limit	`flush_interval`	16	During tracing, the messages of a connection are buffered. As soon as the flush limit number of messages is buffered (or if the connection is closed), those messages are written to the trace file. When set to 0, every SQL trace statement is immediately written to the trace file

Using Wildcards

If you apply filters for the `user`, `application_user`, and `application` parameters, the use of wildcards and exceptions is also supported. The asterisk wildcard character denotes any number of characters and the exclamation mark denotes an exclusion. For example:

```
user=SM*,JONES,!GREEN,!BRO*
```

In this case all users starting with SM will be traced, JONES will be traced, user GREEN will not be traced and all users starting with BRO will not be traced. If terms in the string conflict with each other then the sequence in which the terms occur determines the result. In the following example user SMALL will be traced in spite of the exclusion; the exclusion is ignored because it occurs after the first wildcard.

```
user=SM*,JONES,!SMALL,!BRO*
```

Trace File Rotation

The size and number of trace files are controlled by the following parameters.

Parameter	Default	Description
`max_files`	**`1`**	Sets the maximum number of trace files
`filesize_limit`	**`1610612736`** (or 1.5 GB)	Sets the maximum size of an individual trace file in bytes

> ⚠ Caution
>
> If both the maximum number of files and the maximum file size are reached, SQL tracing stops. If this happens, you can increase the values of `max_files` and `filesize_limit`. See SAP Note 2629103.

SAP HANA SQL Trace Analyzer

SAP HANA SQL trace analyzer is a Python tool you can use to analyze the HANA SQL trace output. The tool gives you an overview of the top SQL statements, the tables accessed, statistical information on different statement types and on transactions executed.

For more information about the installation and usage of SAP HANA SQL trace analyzer, see SAP Knowledge Base Article 2412519 *FAQ: SAP HANA SQL Trace Analyzer*.

Related Information

Diagnostic Files and Logs [page 461]
SAP Note 2412519
SAP Note 2629103

8.13.2.3 Performance Trace

The performance trace is a performance tracing tool built into the SAP HANA database. It records performance indicators for individual query processing steps in the database kernel. You may be requested by SAP Support to provide a performance trace.

Information collected includes the processing time required in a particular step, the data size read and written, network communication, and information specific to the operator or processing-step-specific (for example, number of records used as input and output). The performance trace can be enabled in multiple tenant databases at the same time to analyze cross-database queries.

Performance Trace Files

Performance trace results are saved to the trace files with file extension *.tpt or *.cpt, which you can access with other diagnosis files. To analyze these files, you need a tool capable of reading the output format (*.tpt and *.cpt). SAP Support has tools for evaluating performance traces.

Enabling and Configuring the Performance Trace

You can enable and configure the performance trace in the SAP HANA studio or using the `ALTER SYSTEM * PERFTRACE` SQL statements.

> ### Example
>
> To start the performance trace execute `ALTER SYSTEM START PERFTRACE`.

Configuration Options

Option	Description
Trace file name	The name of the file to which the trace data is automatically saved after the performance trace is stopped
User and application filters	Filters to restrict the trace to a single specific database user, a single specific application user, and a single specific application
Trace execution plans	You can trace execution plans in addition to the default trace data.
Function profiler	The function profiler is a very fine-grained performance tracing tool based on source code instrumentation. It complements the performance trace by providing even more detailed information about the individual processing steps that are done in the database kernel.
Duration	How long you want tracing to run If a certain scenario is to be traced, ensure that you enter a value greater than the time it takes the scenario to run. If there is no specific scenario to trace but instead general system performance, then enter a reasonable value. After the specified duration, the trace stops automatically.

Additional filter options are available in extended mode to restrict the trace data further.

For more information about how to configure the performance trace using SQL, see the *SAP HANA SQL and System Views Reference*.

Related Information

ALTER SYSTEM {START | STOP} PERFTRACE Statement (System Management)
ALTER SYSTEM SAVE PERFTRACE Statement (System Management)
ALTER SYSTEM LOAD PERFTRACE Statement (System Management)
M_PERFTRACE System View

8.13.2.4 Expensive Statements Trace

Expensive statements are individual SQL statements whose execution time exceeds a configured threshold. The expensive statements trace records information about these statements for further analysis and is inactive by default.

If, in addition to activating the expensive statements trace, you enable per-statement memory tracking, the expensive statements trace will also show the peak memory size used to execute the expensive statements.

Expensive Statements Trace Information

If you have the TRACE ADMIN privilege, then you can view expensive statements trace information in the following ways:

- In the *Expensive Statements* app of the SAP HANA cockpit
- In the *Statement Library* in SAP HANA database explorer by searching for *Expensive Statements Analysis*.
- In the M_EXPENSIVE_STATEMENTS system view (see Tips for Performance below)

Enabling and Configuring the Expensive Statements Trace

You can enable and activate the expensive statements trace in the SAP HANA cockpit or the SAP HANA database explorer. Alternatively, you can modify the parameters in the `expensive_statement` section of the `global.ini` configuration file. If you set a value for more than one of the three threshold filters a statement that exceeds any one of the filters is selected as an expensive statement; it is not necessary for all filters you apply to be exceeded.

Configuration Options

> **Note**
>
> The following table shows the configuration parameters which are available; not all of these may be available in the SAP HANA cockpit or the SAP HANA database explorer.

Option	Configuration Parameter	Default Value	Description
Trace status	`enable`	`off`	Specifies the activation status of the trace.

Option	Configuration Parameter	Default Value	Description
Threshold CPU time	`threshold_cpu_time`	**`-1`** (disabled)	Specifies the threshold CPU time of statement execution in microseconds. When set to 0, all SQL statements are traced. **Note** Resource tracking and CPU time tracking must also be enabled. You can do this by configuring the corresponding parameters in the `resource_tracking` section of the `global.ini` file.
Threshold memory	`threshold_memory`	**`-1`** (disabled)	Specifies the threshold memory usage of statement execution in bytes. When set to 0, all SQL statements are traced. **Note** Resource tracking and memory tracking must also be enabled. You can do this by configuring the corresponding parameters in the `resource_tracking` section of the `global.ini` file.
Threshold duration	`threshold _duration`	**`1000000`** (microseconds = 1 second)	Specifies the threshold execution time in microseconds. When set to 0, all SQL statements are traced. In the SAP HANA database explorer, you can set the threshold duration to be measured in seconds or milliseconds.

User, application, and object filters	`user`
	`application_user`
	`application`
	`object`
		Empty string	Specifies filters to restrict traced statements to those of a particular database, application user, application, or tables/views. For `user`, `application_user`, and `application` the use of wildcards is supported (see following subsection *Using Wildcards*).

Option	Configuration Parameter	Default Value	Description
Passport trace level	`passport_tracelevel`	Empty string	If you are activating the expensive statements trace as part of an end-to-end trace scenario with the Process Monitoring Infrastructure (PMI), you can specify the passport trace level as an additional filter. This means that only requests that are marked with a passport of the specified level are traced. **Note** Process tracing is possible only for components in the ABAP and Business Objects stacks.
Trace parameter values	`trace_parameter_values`	**`true`**	In SQL statements, field values may be specified as parameters (using a "?" in the syntax). If these parameter values are not required, then you can disable this setting to reduce the amount of data traced.
Trace flush interval	`trace_flush_interval`	10	Specifies the number of records after which a trace file is flushed.
Use in-memory tracing	`use_in_memory_tracing`	**`true`**	If in-memory tracing is active, then information is cached in memory. Otherwise, the data is written directly to file.
In-memory tracing records	`in_memory_tracing_records`	**`30000`**	Specifies the maximum number of trace records (per service) stored in memory. This setting only takes effect when in memory tracing is active.

Using Wildcards

If you apply filters for the `user`, `application_user`, and `application` parameters, the use of wildcards and exceptions is also supported. The asterisk wildcard character denotes any number of characters and the exclamation mark denotes an exclusion. For example:

```
user=SM*,JONES,!GREEN,!BRO*
```

In this case all users starting with SM will be traced, JONES will be traced, user GREEN will not be traced and all users starting with BRO will not be traced. If terms in the string conflict with each other then the sequence in which the terms occur determines the result. In the following example user SMALL will be traced in spite of the exclusion; the exclusion is ignored because it occurs after the first wildcard.

```
user=SM*,JONES,!SMALL,!BRO*
```

Trace File Rotation

To prevent expensive statement trace information from growing indefinitely, you can limit the size and number of trace files using the following parameters in the `expensive_statement` section of the `global.ini` file.

Parameter	Default	Description
`maxfiles`	**`10`**	Specifies the maximum number of trace files. When the maximum number of trace files reached, the oldest trace file is deleted and a new one opened. When set to 0, trace file rotation is disabled.
`maxfilesize`	**`10000000`** (or 9.5 megabytes)	Specifies the maximum size of an individual trace file in bytes. When the maximum number of files is greater than 1 and the maximum file size is reached, a new trace file is opened. When the maximum number of files is 1, the maximum file size is greater than zero, and the maximum file size is reached, the trace file is deleted and a new one opened.

Tips For Performance

The expensive statements trace may incur an overhead on system performance.

You may be able to adjust the settings of the configuration parameters described above to reduce the volume of trace files and optimize the performance. For example:

- You can increase the value of the threshold parameters to reduce the number of expensive statements
- Turn off `trace_parameter_values` to exclude statements with parameters ('?') in the where clause.
- Make sure that memory caching is in use: `use_in_memory_tracing` is active by default.

You may change these parameter values permanently or temporarily to handle a specific situation. If necessary you can disable expensive tracing by setting `enable` to off.

`use_in_memory_tracing` and M_EXPENSIVE_STATEMENTS

If the `use_in_memory_tracing` parameter is set to true:

- Records are cached in an in-memory buffer (ring buffer). Otherwise, the data is written directly to the file. The `in_memory_tracing_records` parameter specifies the maximum number of trace records (per service) stored in the in-memory buffer.
- M_EXPENSIVE_STATEMENTS is filled only with an in-memory buffer so that the number of trace records in M_EXPENSIVE_STATEMENTS is, at most, the value of `in_memory_tracing_records`.
- Performance is faster because it only reads the limited number of trace records in the in-memory buffer.

When switching `use_in_memory_tracing` from false to true, the in-memory buffer is cleared and filled with the most recent records of the trace files.

When `use_in_memory_tracing` is set to false, M_EXPENSIVE_STATEMENTS is filled with all the trace files and with all the records of the trace files as well.

Related Information

Setting a Memory Limit for SQL Statements [page 431]
Monitoring and Analyzing Expensive Statements (SAP HANA Cockpit)
Expensive Statements Monitoring (SAP HANA Studio)
M_EXPENSIVE_STATEMENTS System View
SAP Note 2180165

8.13.2.5 Deadlock Detection Using SQL

To help with diagnosis of system issues you can create a `waitgraph` file that contains thread information about deadlocks.

From the SQL command line you can use the ALTER SYSTEM CREATE WAITGRAPH command to create a waitgraph file that contains thread information about deadlocks. Full details of the syntax are given in the SAP HANA SQL Reference Guide; here, the usage and optional configuration steps are explained with some examples to illustrate the essential functionality. RESOURCE_ADMIN privileges are required to run this command. You can monitor the progress of the job in the monitoring view M_JOB_PROGRESS.

The following command will write the current waitgraph on service myhost:30003 into the trace file named my_waitgraph.trc.

```
ALTER SYSTEM CREATE WAITGRAPH AT LOCATION 'myhost:30003' INTO FILE
'my_waitgraph.trc'
```

If a file name is not specified then the following default file name structure is used:

```
<servicename>_<hostname>.<port>.waitgraph.<YYYYMMDD-HHMMSS>.<pid>.trc
```

In the above example only threads that caused a deadlock are part of the waitgraph. To get more detail you can use the ALL THREADS syntax to include all threads even if they do not participate in the deadlock:

```
ALTER SYSTEM CREATE WAITGRAPH ALL THREADS INTO FILE 'my_waitgraph.trc'
```

8.13.2.6 Kernel Profiler

The kernel profiler is a sampling profiler built into the SAP HANA database. It can be used to analyze performance issues with systems on which third-party software cannot be installed, or parts of the database that are not accessible by the performance trace. It is inactive by default.

The kernel profile collects, for example, information about frequent and/or expensive execution paths during query processing.

It is recommended that you start kernel profiler tracing immediately before you execute the statements you want to analyze and stop it immediately after they have finished. This avoids the unnecessary recording of irrelevant statements. It is also advisable as this kind of tracing can negatively impact performance.

Enabling and Configuring the Kernel Profiler

You can enable and configure the kernel profiler in the SAP HANA Database Explorer (Trace Configuration) or you can manage the kernel profiler from the SQL command line using the ALTER SYSTEM command. In both cases RESOURCE ADMIN or TRACE ADMIN privileges are required.

Enabling Kernel Profiler using SQL

The kernel profiler statement supports basic functions such as start, stop, save and clear, but also a number of options such as to run the profile for a specific location, for a specific user or within certain memory limits.

The following examples illustrate the basic usage of the ALTER SYSTEM command:

```
ALTER SYSTEM START KERNEL PROFILER
```

```
ALTER SYSTEM SAVE KERNEL PROFILER
```

The SAVE keyword stops the profiler, saves the data (to the trace directory) and clears allocated memory. The following statement starts profiling at host:port ab1234:30003 and filters for the user specific trace (database trace) profile MYTRACEPROFILE

```
ALTER SYSTEM START KERNEL PROFILER AT 'ab1234:30003' TRACEPROFILE
'MYTRACEPROFILE';
```

Refer to the SAP HANA SQL Reference Guide for further options and details of these commands.

Configuration Options in Database Explorer

Option	Description
Service(s) to profile	The service(s) that you want to profile.
Sampling interval	The amount of time the kernel profiler is to wait between call stack retrievals. When you activate the kernel profiler, it retrieves the call stacks of relevant threads several times. It waits between each sample for the length of time specified here minus the time the previous retrieval took.
Memory limit	Memory limit that will stop tracing. The kernel profiler can potentially use a lot a memory. To prevent the SAP HANA database from running out of memory due to profiling, you can specify a memory limit that cannot be exceeded.
Optional filter	The specific database user or application user you want to profile.

Kernel Profiler Traces

Profiling results are saved to two trace files. When started from SQL:

- kernel_profiler_cpu.dot
- kernel_profiler_wait.dot

When started from Database Explorer:

- `CPU_<service>_<host>_<port>_<timestamp>.dot`
- `WAIT_<service>_<host>_<port>_<timestamp>.dot`

To analyze these trace files you need a tool capable of reading the .dot output format, or you may be asked to send the files to SAP Support.

View M_KERNEL_PROFILER

The view M_KERNEL_PROFILER displays the status of the profiler (started or stopped) and provides information about current kernel profilers. The view can also be accessed by users with RESOURCE ADMIN and/or TRACE ADMIN privileges.

Further Information

An SAP Community blog *Kernel profiler filtered by connection ID and statement hash* gives further details of this feature.

Related Information

Diagnostic Files and Logs [page 461]
M_KERNEL_PROFILER System View
ALTER SYSTEM {START | STOP | SAVE | CLEAR} KERNEL PROFILER Statement (System Management)
SAP Community Blog: Kernel profiler filtered by connection ID and statement hash

8.13.2.7 Plan Decision Recorder

To identify performance fluctuations in recompiled execution plans, use a `sqlplan` trace file that explains why the cached execution plan was not used and a recompilation was required.

The plan decision recorder logs SQL optimizer details for recompiled execution plans into a dedicated `sqlplan` trace file. This trace file contains details like the plan ID, recompilation status, the execution engine used, the reason why the HEX execution engine was bypassed, and additional relevant data. This data can be used for diagnosing performance fluctuations caused by parameter changes, providing insights into why the cached plan was bypassed and a recompilation was necessary to optimize for the new parameter values. Additionally, this information is useful when working with the SQL Plan Stability app.

The plan decision recorder is inactive by default.

Configuring the Plan Decision Recorder

You can manage the plan decision recorder in SAP HANA database explorer or SQL command line using the ALTER SYSTEM ALTER CONFIGURATION statement. The TRACE ADMIN privilege is required.

The configuration parameters for the plan decision recorder are in the `indexserver.ini` configuration file. These parameters are visible once set.

- enable: false. Activates or deactivates the trace (the plan decision recorder).
- enable_non_parameterized_query: false. Controls whether non-parameterized queries are recorded in the trace.
- maxfiles. Defines the maximum number of trace files to retain. The default is 10.
- maxfilesize. Defines the maximum size of each individual trace file in megabytes. The default is 10. Note: A 10-megabyte trace file can store optimizer information for approximately 20,000 SQL statements.

For more information on configuration parameters, refer to the *SAP HANA Configuration Parameter Reference Guide, Configuration Parameters SPS 08*.

Example

The following example demonstrates how to manage the size of the plan decision recorder trace file by setting a maximum file size limit:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'system')
SET ('sql_plan_decision_recorder','maxfilesize') = '20'
WITH RECONFIGURE
COMMENT 'Limit the size of the plan decision recorder trace file to 20
megabytes';
```

Viewing the Plan Decision Recorder Traces

The information is saved in `sqlplan` trace files using the naming format `indexserver_<host>.<port>.sqlplan.<counter>.trc`.

To locate the `sqlplan` trace file in the SAP HANA database explorer, execute one of the following queries:

```
SELECT host, file_name
FROM M_TRACEFILES
WHERE file_name LIKE '%sqlplan%';
```

or

```
SELECT *
FROM M_TRACEFILES
WHERE FILE_NAME LIKE '%sqlplan%'
ORDER BY FILE_MTIME;
```

To view the contents of a `sqlplan` trace file, use the M_TRACEFILE_CONTENTS system view with the following query:

```
SELECT CONTENT
FROM M_TRACEFILE_CONTENTS
WHERE HOST = '<host>'
```

```
  AND FILE_NAME = 'indexserver_<host>.<port>.sqlplan.<counter>.trc';
```

To view the contents of a `sqlplan` trace file in a tabular form, use the following SQL procedure. This SQL procedure collects trace data related to SQL execution plans from `sqlplan` trace files and returns it as a table.

```
CREATE FUNCTION collect_record() RETURNS TABLE(result NVARCHAR(5000))
AS
BEGIN
    USING sqlscript_string AS lib;
    DECLARE single_file TABLE(result NVARCHAR(5000));
    DECLARE result_tab TABLE(result NVARCHAR(5000));
    DECLARE content_agg NCLOB;
    DECLARE cursor file_cursor FOR SELECT host, file_name FROM m_tracefiles
WHERE file_name LIKE '%sqlplan%';
    FOR r AS file_cursor DO
        SELECT string_agg(content) INTO content_agg FROM m_tracefile_contents
WHERE host = r.host AND file_name = r.file_name;
        single_file = lib:split_to_table(:content_agg, CHAR(10));
        INSERT INTO :result_tab SELECT * FROM :single_file;
    END FOR;
    RETURN(:result_tab);
END;
CREATE TABLE plan_recorder AS (
    SELECT * FROM (SELECT
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 1), ';') AS TIMESTAMP,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 2), ';') AS THREAD_ID,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 3), ';') AS
CONNECTION_ID,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 4), ';') AS
TRANSACTION_ID,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 5), ';') AS
UPDATE_TRANSACTION_ID,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 6), ';') AS DB_USER,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 7), ';') AS
SCHEMA_NAME,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 8), ';') AS PLAN_ID,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 9), ';') AS
STATEMENT_HASH,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 10), ';') AS
RECOMPILE_STATE,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 11), ';') AS
EXECUTION_ENGINE,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 12), ';') AS
HEX_REJECTION_REASON,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 13), ';') AS
LAST_INVALIDATED_REASON,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 13), ';') AS
PARAMETER_VALUES,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 13), ';') AS
STATEMENT_STRING,
        RTRIM(SUBSTR_REGEXPR('[^;]*;' IN result OCCURRENCE 14), ';') AS
EXPLAIN_PLAN
        FROM collect_record())
    WHERE TIMESTAMP != 'TIMESTAMP');
SELECT *
FROM plan_recorder;
```

Clearing the Plan Decision Recorder Traces

To clear the `sqlplan` trace files, execute the ALTER SYSTEM CLEAR TRACES statement.

Related Information

Traces [page 462]
SQL Plan Stability
Links to other documents
M_TRACEFILES System View (SAP HANA SQL Reference Guide for SAP HANA Platform)
M_TRACEFILE_CONTENTS System View (SAP HANA SQL Reference Guide for SAP HANA Platform)
M_SQL_PLAN_CACHE System View (SAP HANA SQL Reference Guide for SAP HANA Platform)
ALTER SYSTEM CLEAR TRACES Statement (SAP HANA SQL Reference Guide for SAP HANA Platform)
ALTER SYSTEM ALTER CONFIGURATION Statement (SAP HANA SQL Reference Guide for SAP HANA Platform)
Configuration Parameters SPS 08 (SAP HANA Configuration Parameter Reference Guide)

8.13.2.8 Traces and Trace Configuration for Internal Web Dispatcher

Several traces and trace configuration options are available for the internal Web Dispatcher, which runs as a native SAP HANA service (`webdispatcher`).

The HANA internal web dispatcher is an implementation of the NetWeaver web dispatcher. The support component ID for the web dispatcher is: BC-CST-WDP. For more detailed information you can refer to the SAP Web Dispatcher documentation and the NetWeaver documentation using the links given below under Related Information.

Developer Trace

The developer trace is the main trace for the Web Dispatcher and contains technical information for troubleshooting problems.

The developer trace file is `webdispatcher_<host>.<port>_dev_webdisp`.

You can configure the developer trace in the following ways:

- Changing the database trace level for the `dev_webdisp` component of the `webdispatcher` service
  The default trace level is ERROR.
- Changing (or adding) the property `rdisp/trace` in the `[profile]` section of the `webdispatcher.ini` configuration file.
  Possible values are 0, 1, 2, and 3.

Database Trace

The database trace files for the Web Dispatcher contain secondary information related to the Web Dispatcher's integration into the SAP HANA integration system (start/stop, configuration changes, and so on).

The database trace files are:

- `webdispatcher_<host>.<port>.<3_digit_file_counter>.trc`
- `webdispatcher_alert.<host>.trc`

You can configure the database trace by changing the trace level for the `webdispatcher` component of the `webdispatcher` service.

Header Trace

The header trace allows you to analyze HTTP requests and responses efficiently since it contains only the request data and no information about the internal workings of Web Dispatcher.

You can activate the header trace by adding the property `icm/http/trace_info` in the `[profile]` section of the `webdispatcher.ini` configuration file and setting the value to **`true`**. The trace level is `false` by default.

Header trace information is written to the `dev_webdisp` trace file.

HTTP Access Log

To monitor all HTTP(s) requests processed in an SAP HANA system, you can set up the internal Web Dispatcher to write a standardized HTTP log for each request.

To configure the Web Dispatcher to log all HTTP(s) requests, you add the property `icm/http/logging_0` to the `[profile]` section of the `webdispatcher.ini` configuration file, specifying the following value:

```
PREFIX=/, LOGFILE=$(DIR_INSTANCE)/trace/access_log-%y-%m-%d, MAXSIZEKB=10000,
SWITCHTF=day, LOGFORMAT=SAP
```

The access log file is `access_log-<timestamp>`.

> **Example**
>
> Sample log file entry: `[26/Nov/2014:13:42:04 +0200] 10.18.209.126 BOB - "GET /sap/xse/test/InsertComment.xsjs HTTP/1.1" 200 5 245`

The last three numbers are the HTTP response code, the size in bytes, and the response time in milliseconds. For more information about logging and alternative log formats, see the Internet Communication Manager (ICM) documentation on SAP Help Portal.

Related Information

SAP Web Dispatcher
rdisp/TRACE* Parameters
icm/HTTP/trace_infoicm/HTTP/trace_info
icm/HTTP/logging_<xx>

Logging in the ICM and SAP Web Dispatcher

8.13.3 Troubleshooting an Inaccessible or Unresponsive SAP HANA System

If a system cannot be reached by SQL or if performance is poor, tools are available to help to diagnose the cause of the problem.

For situations when a system cannot be reached by SQL or is experiencing performance problems, tools are available so that you or an SAP support engineer can access diagnosis information and perform emergency operations to resolve the situation.

For more information refer to:

- Troubleshoot an Unresponsive Database in SAP HANA Cockpit
- Troubleshoot Unresponsive System in SAP HANA Studio

Related Information

Troubleshoot an Unresponsive Database (SAP HANA Cockpit)
Troubleshoot an Unresponsive System (SAP HANA Studio)

8.13.4 Collecting Diagnosis Information for SAP Support

To help SAP Support analyze any problems with your system, you can collect a range of diagnosis information.

To help SAP Support analyze and diagnose problems with your system, you can collect a range of diagnosis information from your system into a zip file which you can download and attach to a support message.

You can trigger the collection of diagnosis information from the SAP HANA cockpit, the SAP HANA studio, or the command line. Refer to the following topics for more information:

- Collect and Download Diagnosis Information in SAP HANA Cockpit
- Collect and Download Diagnosis Information in SAP HANA Studio
- Collect Diagnosis Information from the Command Line

In addition to the following topic on the `fullSystemInfoDump` script a detailed SAP note is available on all the options available for running and triggering runtime dumps: 2400007 *FAQ: SAP HANA Runtime Dumps*.

Related Information

Collect and Download Diagnosis Information (SAP HANA Cockpit)

Collect and Download Diagnosis Information (SAP HANA Studio)
Diagnosis Information Collected [page 490]
SAP Note 2400007

8.13.4.1 Collect Diagnosis Information from the Command Line

The `fullSystemInfoDump.py` script allows you to collect information from your system, even when it is not accessible by SQL. You can then add this information to a support message, for example. The script is part of the SAP HANA server installation and can be executed directly from the command line.

Prerequisites

You are logged on as the operating system user, `<sid>`adm.

Context

The `fullSystemInfoDump.py` script is part of the server installation and can be run from the command line. It is located in the directory `$DIR_INSTANCE/exe/python_support`.

> **Note**
>
> In a multiple-container system, only the system administrator can collect diagnosis information from the command line since tenant database administrators do not have operating system access. Tenant database administrators must use the SAP HANA cockpit or studio to collect diagnosis information from their database.

Procedure

Start the script from its location with the command:

```
python fullSystemInfoDump.py
```

You can modify the command with several command line options. To see the available options, specify the option --help - see following table.

If the system can be reached by SQL (and you have not specified the option --nosql), the script starts collecting diagnosis information. If the system cannot be reached by SQL, the script starts collecting support information but does not export data from system views.

Option	Description
--version	Displays script version number
--help	Shows help
--nosql	Excludes the collection of system views **Note** If you are connected to the system database, only information from the system views of the system database will be collected. Information from the system views of tenant databases will **not** be collected regardless of this option.
--file `<filename>`	Zips the specified file in its source directory **Note** This option only zips the file; it does not trigger the collection of any other information.
--days `<no. of days>`	Collects information from the specified number of past days The default value is 7. **Note** You cannot use this option with the options --fromDate and --toDate.
--fromDate `<YYYY-MM-DD>`	Collects information starting from the specified date
--toDate `<YYYY-MM-DD>`	Collects information up to the specified date
--rtedump	Restricts the information collected to an RTE dump file or files You can configure the creation and collection of RTE dump files further with the remaining options.
--indexservers `<comma-separated list of index servers>`	Specifies the index server(s) from which RTE dump files are to be collected By default, dump files are created and collected for all index servers
--interval `<interval in minutes>`	Specifies the interval at which RTE dump files are to be collected Possible values are 1, 5, 10, 15, and 30. The default value is 1.
--sets `<no. of RTE dump file sets>`	Specifies the number of RTE dump file sets to be collected. Possible values are 1, 2, 3, 4, and 5.
--tenant `<tenant database name>`	Specifies which tenant database information is to be collected from You must specify a database name. To collect information from the system database, specify **SYSTEMDB**.

Results

The script creates a zip file containing the collected information and saves it to the directory `DIR_GLOBAL/sapcontrol/snapshots`. `DIR_GLOBAL` typically points to `/usr/sap/<sid>/SYS/global`.

The name of the zip file is structured as follows:
`fullsysteminfodump_<SID>_<DBNAME>_<HOST>_<timestamp>.zip`

The timestamp in the file name is UTC. The host and SID are taken from the `sapprofile.ini` file.

The output directory for the zip file is shown as console output when the script is running.

Related Information

Collect and Download Diagnosis Information
Collect and Download Diagnosis Information in SAP HANA Studio
Diagnosis Information Collected [page 490]

8.13.4.2 System Information Dump Files in SQL

You can obtain a run-time environment (RTE) dump file from the SQL command line.

In addition to the ways already described you can also obtain a run-time dump file using SQL. You can use the ALTER SYSTEM CREATE RUNTIMEDUMP command to create either a complete or a selective dump file about system activity. Full details of the syntax are given in the SAP HANA SQL Reference Guide; here, the usage and optional configuration steps are explained with some examples to illustrate the essential functionality. RESOURCE_ADMIN privileges are required to run this command. You can monitor the progress of the job in the monitoring view M_JOB_PROGRESS.

The basic syntax allows you to specify a host and an output file name. By default all sections of the dump file are written:

```
ALTER SYSTEM CREATE RUNTIMEDUMP AT LOCATION 'myhost:30003' INTO FILE
'my_rte_dump.trc'
```

If a file name is not specified then the following default file name structure is used:

```
<servicename>_<hostname>.<port>.rtedump.<YYYYMMDD-HHMMSS>.<pid>.trc
```

For example:

```
indexserver_myhost0001.30203.rtedump.20180627-104305.012220.trc
```

You can also name specific sections of the run time dump file to be included; for example, the following command prints only the STACK_SHORT section:

```
ALTER SYSTEM CREATE RUNTIMEDUMP AT LOCATION 'myhost:30003' SECTIONS
('STACK_SHORT') INTO FILE 'my_rte_dump.trc'
```

Alternatively, you can omit the SECTIONS argument and run the command based on a pre-defined profile which selects sections of the dump file to include. Profiles are defined as configuration file values in the

global.ini file. The following command, for example, will write a runtime dump that contains all sections defined by profile myRTEProfile:

```
ALTER SYSTEM CREATE RUNTIMEDUMP PROFILE 'myRTEProfile' INTO FILE
'my_rte_dump.trc'
```

The profile myRTEProfile may have the following defined as default sections, which would print all sections except STACK_FULL and MEMMAP.

```
!STACK_FULL,!MEMMAP,*
```

The profile is defined in the [runtimedump] section of the global.ini file with a name in the following format: `<profileName>_sections`; the value for the parameter is a comma-separated list of section names to include in the run-time dump output if the SECTIONS argument is not specified in the SQL statement. You can use the asterisk as a wildcard character at the end of the string and the exclamation mark before a section name to explicitly exclude a section which would otherwise be selected, as shown in the following examples:

- Default_sections = !STACK_FULL,!MEMMAP,*
- my_sections = *

Changes made to this configuration parameter are immediately effective, a restart is not necessary.

8.13.4.3 Diagnosis Information Collected

The Python support script `fullSystemInfoDump.py` script collects a range of information from your system for diagnosis purposes. It can be triggered from the SAP HANA cockpit, the SAP HANA studio, or directly from the command line.

> **Note**
>
> All of the following file types are collected unless the option --rtedump is specified, in which case only runtime environment (RTE) dump files are created and collected.

Log File

All information about what has been collected is shown as console output and is written to a file named `log.txt` that is stored in the zip file.

Trace Files

Each of the following trace files is put into a file with the same name as the trace file. For storage reasons, only the trace files from the last 7 days are collected unabridged. Older trace files are not collected. This behavior can be changed by using option --days or with the options --fromDate and --toDate.

Crashdump files and runtime dump files are always collected unabridged.

- $DIR_INSTANCE/<SAPLOCALHOST>/trace/compileserver_alert_<SAPLOCALHOST>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/compileserver_<SAPLOCALHOST>.<...>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/daemon_<SAPLOCALHOST>.<...>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/indexserver_alert_<SAPLOCALHOST>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/indexserver_<SAPLOCALHOST>.<...>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/nameserver_alert_<SAPLOCALHOST>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/nameserver_history.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/nameserver_<SAPLOCALHOST>.<...>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/preprocessor_alert_<SAPLOCALHOST>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/preprocessor_<SAPLOCALHOST>.<...>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/statisticsserver_alert_<SAPLOCALHOST>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/statisticsserver_<SAPLOCALHOST>.<...>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/xsengine_alert_<SAPLOCALHOST>.trc
- $DIR_INSTANCE/<SAPLOCALHOST>/trace/xsengine_<SAPLOCALHOST>.<...>.trc

Configuration Files

All configuration files are collected unabridged and stored in a file with the same name as the .ini file:

- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/attributes.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/compileserver.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/daemon.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/executor.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/extensions.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/filter.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/global.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/indexserver.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/inifiles.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/localclient.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/mimetypemapping.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/nameserver.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/preprocessor.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/scriptserver.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/statisticsserver.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/validmimetypes.ini
- $DIR_INSTANCE/<SAPLOCALHOST>/exe/config/xsengine.ini

Database System Log Files

The following backup files are collected unabridged:

- `$DIR_INSTANCE/<SAPLOCALHOST>/trace/backup.log`
- `$DIR_INSTANCE/<SAPLOCALHOST>/trace/backint.log`

.

RTE Dump Files

For each index server, an RTE dump file containing information about threads, stack contexts, and so on is created and stored in the file `indexserver_<SAPLOCALHOST>_<PORT>_runtimedump.trc`. These files are stored unabridged.

Crashdump Information

Crashdump files for services are collected unabridged.

Performance Trace Files

Performance trace files with the suffix *.tpt are collected unabridged.

Kerberos Files

The following Kerberos files are collected:

- `/etc/krb5.conf`
- `/etc/krb5.keytab`

System Views

If the collection of system views is not excluded (option --nosql specified), all rows of the following system views (with the exceptions mentioned below) are exported into a CSV file with the name of the table.

> **Note**
>
> If you are connected to the system database of a multiple-container system, only information from the system views of the system database will be collected. Information from the system views of tenant databases will **not** be collected regardless of this option.

- `SYS.M_CE_CALCSCENARIOS WHERE SCENARIO_NAME LIKE '%_SYS_PLE%'`

- SYS.M_CONNECTIONS with CONNECTION_ID > 0
- SYS.M_DATABASE_HISTORY
- SYS.M_DEV_ALL_LICENSES
- SYS.M_DEV_PLE_SESSIONS_
- SYS.M_DEV_PLE_RUNTIME_OBJECTS_
- SYS.M_EPM_SESSIONS
- SYS.M_INIFILE_CONTENTS
- SYS.M_LANDSCAPE_HOST_CONFIGURATION
- SYS.M_RECORD_LOCKS
- SYS.M_SERVICE_STATISTICS
- SYS.M_SERVICE_THREADS
- SYS.M_SYSTEM_OVERVIEW
- SYS.M_TABLE_LOCATIONS
- SYS.M_TABLE_LOCKS
- SYS.M_TABLE_TRANSACTIONS
- _SYS_EPM.VERSIONS
- _SYS_EPM.TEMPORARY_CONTAINERS
- _SYS_EPM.SAVED_CONTAINERS
- _SYS_STATISTICS.STATISTICS_ALERT_INFORMATION
- _SYS_STATISTICS.STATISTICS_ALERT_LAST_CHECK_INFORMATION

  > **Note**
  >
  > Only the first 2,000 rows are exported.

- _SYS_STATISTICS.STATISTICS_ALERTS

  > **Note**
  >
  > Only the first 2,000 rows are exported.

- _SYS_STATISTICS.STATISTICS_INTERVAL_INFORMATION
- _SYS_STATISTICS.STATISTICS_LASTVALUES
- _SYS_STATISTICS.STATISTICS_STATE
- _SYS_STATISTICS.STATISTICS_VERSION

The first 2,000 rows of all remaining tables in schema _SYS_STATISTICS are exported ordered by column SNAPSHOT_ID.

Additional Information Collected If SQL Connection Is Not Available

All available topology information is exported to a file named `topology.txt`. It contains information about the host topology in a tree-like structure. The keys are grouped using brackets while the corresponding values are referenced by the symbol ==>. For example:

```
[]
```

```
  ['host']
    ['host', 'ld8521']
      ['host', 'ld8521', 'role']
        ==> worker
      ['host', 'ld8521', 'group']
        ==> default
      ['host', 'ld8521', 'nameserver']
        ['host', 'ld8521', 'nameserver', '30501']
          ['host', 'ld8521', 'nameserver', '30501', 'activated_at']
            ==> 2011-08-09 16:44:02.684
          ['host', 'ld8521', 'nameserver', '30501', 'active']
            ==> no
          ['host', 'ld8521', 'nameserver', '30501', 'info']
            ['host', 'ld8521', 'nameserver', '30501', 'info', 'cpu_manufacturer']
              ==> GenuineIntel
            ['host', 'ld8521', 'nameserver', '30501', 'info',
'topology_mem_type']
              ==> shared
            ['host', 'ld8521', 'nameserver', '30501', 'info',
'sap_retrieval_path_devid']
              ==> 29
            ['host', 'ld8521', 'nameserver', '30501', 'info', 'build_time']
              ==> 2011-07-26 17:15:05
            ['host', 'ld8521', 'nameserver', '30501', 'info', 'net_realhostname']
              ==> -
            ['host', 'ld8521', 'nameserver', '30501', 'info', 'build_branch']
              ==> orange_COR
            ['host', 'ld8521', 'nameserver', '30501', 'info', 'mem_swap']
              ==> 34359730176
            ['host', 'ld8521', 'nameserver', '30501', 'info', 'mem_phys']
```

8.13.5 Problem Analysis Using hdbcons

`hdbcons` is a command line tool with which commands can be executed against running processes using a separate communication channel. It is intended for problem analysis by SAP HANA development support.

> ⚠ Caution
>
> Technical expertise is required to use `hdbcons`. To avoid incorrect usage, use `hdbcons` only with the guidance of SAP HANA development support.

`hdbcons` commands can be executed directly in the Administration editor on the *Console* tab. However, it is not visible by default. You can enable the display of the *Console* tab in the preferences of the SAP HANA studio under *SAP HANA* > *Global Settings*.

To see a list of available commands and display the help for a command, enter the command `help`.

Each command is subject to an individual authorization check. Operating system user (<sid>adm) access is not required.

8.13.6 Open a Support Connection

In some support situations, it may be necessary to allow an SAP support engineer to log into your system to analyze the situation.

Procedure

1. To enable a support user to log on to your system, complete the following tasks:
   a. Install the SAProuter as described on SAP Support Portal.
   b. Set up a support connection as described in SAP Note 1634848 (*SAP HANA database service connections*).
   c. Configure a Telnet connection as described in SAP Note 37001 (*Telnet link to customer systems*)
   d. Configure an SAP HANA database connection as described in SAP Note 1592925 (*SAP HANA studio service connection*).
   e. Configure a TREX/BIA/HANA service connection as described in SAP Note 1058533 (*TREX/BIA/HANA service connection to customer systems*).
2. Create a database user and grant the MONITORING role.

   The MONITORING role allows a database user to open the SAP HANA Administration Console perspective of the SAP HANA studio with read-only access to the system, system views, statistics views, trace files, and so on. However, this role does not provide any privileges for accessing application data. With the MONITORING role, it is also not possible to change the configuration of or start and stop a system. You can grant the MONITORING role to a support engineer if SAP support needs to connect to the system. Depending on the issue to be analyzed, further privileges may be needed to allow sufficient analysis (for example, to access application data or data models).

Related Information

SAProuter
SAP Note 1634848
SAP Note 37001
SAP Note 1592925
SAP Note 1058533

8.14 Scheduling Administrative Tasks

Frequently occurring tasks can be automated using the built-in scheduler.

Using the built-in scheduler, you can execute tasks repeatedly by creating scheduled jobs that execute procedures. For example, you can schedule data backups or delta backups to run without supervision at

specific intervals. The owner of a job can grant or revoke rights for it to allow other users to execute, alter, or drop it.

> **Note**
>
> Depending on the complexity of the scheduled SQL procedures and statements, a scheduled task can have a negative impact on the performance of the database system.

You can create a scheduled job in the current or specified schema using the CREATE SCHEDULER JOB statement which requires the CREATE ANY privilege in the schema where the scheduled job was created and the EXECUTE privilege on the procedure referenced by the scheduled job:

```
CREATE SCHEDULER JOB [ <schema_name>.] <job_name>
   CRON <cron> [ <job_recurrence_range> ]
   [ ENABLE | DISABLE ] PROCEDURE [ <schema_name>.]<procedure_name> [ PARAMETERS
<parameter_list> ]<NAME>=<CONST_EXPR> [, <NAME>=<CONST_EXPR>[, ...] ]
```

The value for the cron variable specifies the frequency of the job, a number of options and variables are available for specifying recurring times. For details and an example see the *SAP HANA SQL Reference Guide*.

> **Note**
>
> A scheduled job becomes invalid if permissions or objects are missing, in this case the STATUS value in the M_SCHEDULER_JOBS view is shown as ERROR. Restoring missing permissions or objects re-enables a scheduled job.

Use ALTER SCHEDULER JOB to enable a disabled scheduled job (see *ALTER SCHEDULER JOB* for details).

The status and the history of the scheduled jobs is stored in the M_SCHEDULER_JOBS system view. Users see jobs that they created or on which they've been granted the ALTER or DROP object privilege. The SYSTEM user or users with the CATALOG READ system privilege see status and history for all jobs (see *M_SCHEDULER_JOBS* for details).

A set of configuration parameters in the `indexserver.ini` file is available to control the history of the scheduled jobs. The configuration parameters are in the `job_scheduler` section of the configuration file.

[job_scheduler]

Parameter	history_retention_time
Short Description	Maximum retention time of finished jobs in M_SCHEDULER_JOBS.
Full Description	This parameter controls how long an executed job is stored in the M_SCHEDULER_JOBS system view. The default value of 40320 minutes refers to a maximum duration of 4 weeks. If the parameter value is set to 0, no retention time is set.
Type	Integer
Change	Online
Default	40320

Parameter	history_record_limit
Short Description	Maximum number of finished jobs in M_SCHEDULER_JOBS.
Full Description	This parameter controls how many executed jobs are stored in the M_SCHEDULER_JOBS system view. If the parameter value is set to 0, no limit is set.
Type	Integer
Change	Online
Default	1000

Related Information

Links to SAP HANA SQL Reference Guide

CREATE SCHEDULER JOB Statement (Data Definition)
ALTER SCHEDULER JOB Statement (Data Definition)
DROP SCHEDULER JOB Statement (Data Definition)
COMMENT ON Statement (Data Definition)
SCHEDULER_JOB_PARAMETERS System View
M_SCHEDULER_JOBS System View
SCHEDULER_JOBS System View
SAP HANA SQLScript Reference

9 Security Administration and User Management

Security administration, including user management, represents a category of database administration usually handled separately from general administration tasks.

This section provides an overview of the tasks related to security administration and user management. You can perform many of these tasks using the SAP HANA cockpit and we recommend that you do so where possible.

> **Note**
>
> The *SAP HANA Security Guide* is the entry point for all information relating to the secure operation and configuration of SAP HANA Cloud. Please also refer to the *SAP HANA Security Checklists and Recommendations*.

Related Information

SAP HANA Administration with SAP HANA Cockpit
SAP HANA Security Guide
SAP HANA Security Checklists and Recommendations

9.1 Monitoring Critical Security Settings

SAP HANA has many configuration settings that allow you to customize your system for your implementation scenario and system environment. Some of these settings are important for the security of your system. Misconfiguration could leave your system vulnerable.

Use the SAP HANA cockpit to monitor critical security-related settings at a glance. At the more detailed level, the *Security Checklist* app allows you to see which settings deviate from the recommended configuration and if possible, you can navigate to the cockpit app that allows you change the configuration.

In addition, please refer to the document *SAP HANA Security Checklists and Recommendations*. This document provides more detailed information as well as recommendations for many settings.

Related Information

Configuring SAP HANA Securely
SAP HANA Security Checklists and Recommendations

9.2 Auditing Activity in the SAP HANA Database

Auditing allows you to monitor and record selected actions performed in the SAP HANA database, providing you with visibility on who did what in the database (or tried to do what) and when.

The following actions are typically audited, for example:

- Changes to user authorization
- Creation or deletion of database objects
- Authentication of users
- Changes to system configuration
- Access to or changing of sensitive information

For a list of recommended audit policies and other best practices, see *Best Practices and Recommendations for Creating Audit Policies* in the *SAP HANA Security Guide*.

The main tasks related to auditing database activity can be performed using the SAP HANA cockpit. These include:

1. Enabling auditing in the database
2. Configuring audit trails
3. Creating and managing the required audit policies

   > **Note**
   >
   > A template configuration is available in the SAP HANA cockpit to help you apply SAP's recommended settings for creating audit policies.

4. Reviewing and managing the audit log.

For more information about performing these tasks, see the SAP HANA cockpit documentation. For more detailed information about how auditing works, see the *SAP HANA Security Guide*.

Related Information

Auditing Activity in SAP HANA
Best Practices and Recommendations for Creating Audit Policies
Configuring Database Auditing

9.3 Managing Data Encryption in SAP HANA

SAP HANA provides comprehensive encryption capabilities both for data at rest. SAP HANA encryption features use SAP's standard cryptographic library (CommonCryptoLib).

This section explains how to perform administration tasks related to data-at-rest encryption, for example, configuring encryption and managing encryption keys. You can perform many of these tasks using the SAP HANA cockpit and we recommend that you do so where possible.

Please also refer to the data storage security section in the *SAP HANA Security Guide*.

Related Information

Data Storage Security in SAP HANA
Managing Server-Side Data Encryption
SAP HANA Network and Communication Security
SAP Note 1848999

9.3.1 Server-Side Data Encryption Services

SAP HANA features encryption services for encrypting data at rest, as well as an internal encryption service available to applications with data encryption requirements.

Passwords

On the SAP HANA database server, all passwords are stored securely:

- Operating system user passwords are protected by the standard operating system mechanism, `/etc/shadow` file.
- All database user passwords are stored in salted hash form using PBKDF2 (Password-Based Key Derivation Function 2) and, for downward compatibility, secure hash algorithm SHA-256.
  The SAP HANA implementation of PBKDF2 uses the SHA-256 secure hash algorithm and 15,000 iterations.

  > **Note**
  >
  > The hash method SHA-256 can be disabled by setting the parameter `[authentication] password_hash_methods` in the `global.ini` configuration file to **`pbkdf2`**. The default value is **`pbkdf2,sha256`**.

- Credentials required by SAP HANA applications for outbound connections are securely stored in a database-internal credential store. This internal credential store is in turn secured using the internal application encryption service.

Data-at-Rest Encryption

To protect data saved to disk from unauthorized access at operating system level, the SAP HANA database supports data encryption in the persistence layer for the following types of data:

- Data volumes

- Redo log volumes
- Data and log backups

Security-Relevant Application Data

An internal encryption service is used to encrypt sensitive application data. This includes credentials required by SAP HANA for outbound connections, private keys of the SAP HANA server stored in the database, and data in secure stores defined by developers of SAP HANA XS applications (classic or advanced) or other applications (through SQL).

Secure Stores

SAP HANA uses either the instance SSFS (secure store in the file system) or the local secure store (LSS), to protect the root keys used for all data-at-rest encryption services and the internal application encryption service. It uses the system PKI SSFS to protect the system-internal root certificates required for secure internal communication.

Related Information

Data and Log Volume Encryption
Backup Encryption
Internal Application Encryption Service
Server-Side Secure Stores [page 501]
Local Secure Store (LSS)
Encryption Key Management

9.3.1.1 Server-Side Secure Stores

SAP HANA employs a secure storage system to safeguard the root keys essential for both data-at-rest encryption services and the internal application encryption service.

This protection can be configured using either the default Local Secure Store (LSS) (**recommended**) or the Instance Secure Store in the File System (Instance SSFS)(**legacy and deprecated secure store**).

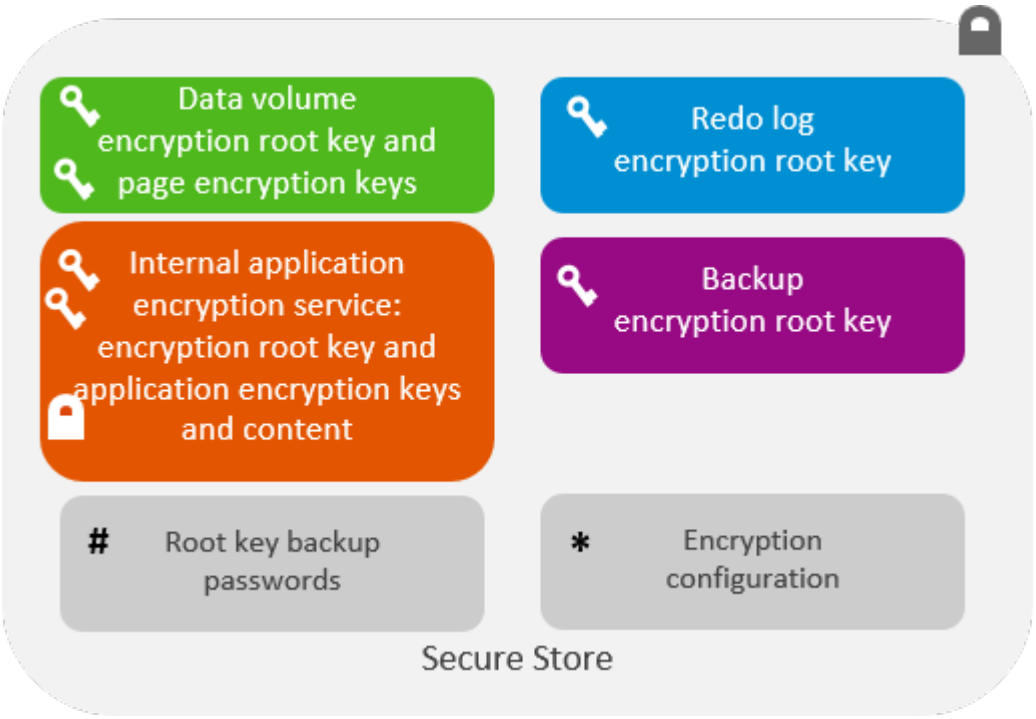
SAP HANA employs the System Secure Store in the File System with Public Key Infrastructure (System PKI SSFS) to safeguard system-internal root certificates for secure internal communication.

Secure Store for Encryption Root Keys and Internal Application Encryption Service (Instance SSFS and LSS)

SAP HANA uses the secure store to protect the following individually for each database:

- The root keys used for:
  - Data volume encryption (PERSISTENCE)
  - Redo log encryption (LOG)
  - Data and log backup encryption (BACKUP)
  - Internal Application Encryption Service (DPAPI)
- Content of Internal Application Encryption Service (DPAPI)
- Data page encryption keys (PERSISTENCE)
- The password of the root key backup
- The encryption configuration

The root keys protect all encryption keys used in the SAP HANA database from unauthorized access.



Secure store content of one database (system database and tenant)

Secure Store Content of a Database

There are two variations of the encryption root key secure store:

- **Local Secure Store (LSS)(Recommended)**: A separate lightweight utility that runs as a separate service on the SAP HANA server under a different operating system user. It uses tenant-specific files. It is the default secure store. For more information about the LSS, see *Local Secure Store*. For additional details, see SAP Note 3048191.

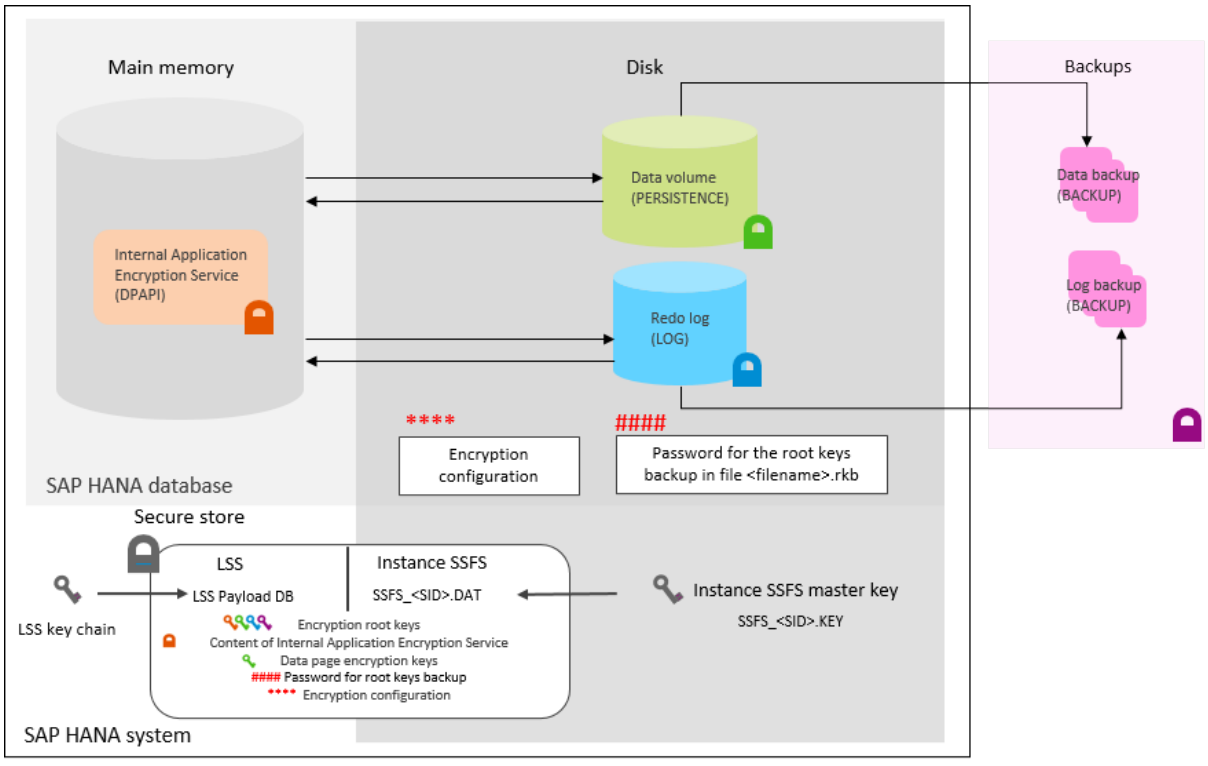
- **Instance Secure Store in the File System (Instance SSFS)(legacy and deprecated key store)**: A single file in the local file system that hosts the encryption keys for the system database and all tenants. For more information, refer to SAP Note 3544974.

> **Note**
>
> To prevent data encrypted in the SAP HANA database from becoming inaccessible, the content of the Instance SSFS and key information in the database must remain consistent. If the SSFS becomes inconsistent, the database detects this and issues alert 57. In this case, it is recommended to contact SAP Support to resolve the issue (SAP Note 2097613).

Encryption Services and Keys

The following diagram provides an overview of which data in SAP HANA can be encrypted using a dedicated encryption service, and how all associated encryption root keys are stored in the configured secure store.

> **Note**
>
> The following diagram shows only one database. However, a system always has a system database and any number of tenant databases. Every database in the system has its own encryption root keys for each of the available encryption services. The root keys of all databases are stored in the configured secure store.



Encryption Services and Keys of one database

The contents of the Instance SSFS in file SSFS_.DAT are protected by an individual master key in file SSFS_.KEY. Both files belong together.

Master Key Change

Unique master keys are generated for the instance secure store during installation or update.

If you received your system pre-installed from a hardware or hosting partner, it is recommended to change them immediately after handover to ensure they are not known outside of your organization. You can also change the master keys at any time later.

> ⓘ Note
>
> The default path of the key file of the Instance SSFS is:
>
> /usr/sap/<sid>/SYS/global/hdb/security/ssfs/
>
> If you change the default path, you must reconfigure it after a system rename.

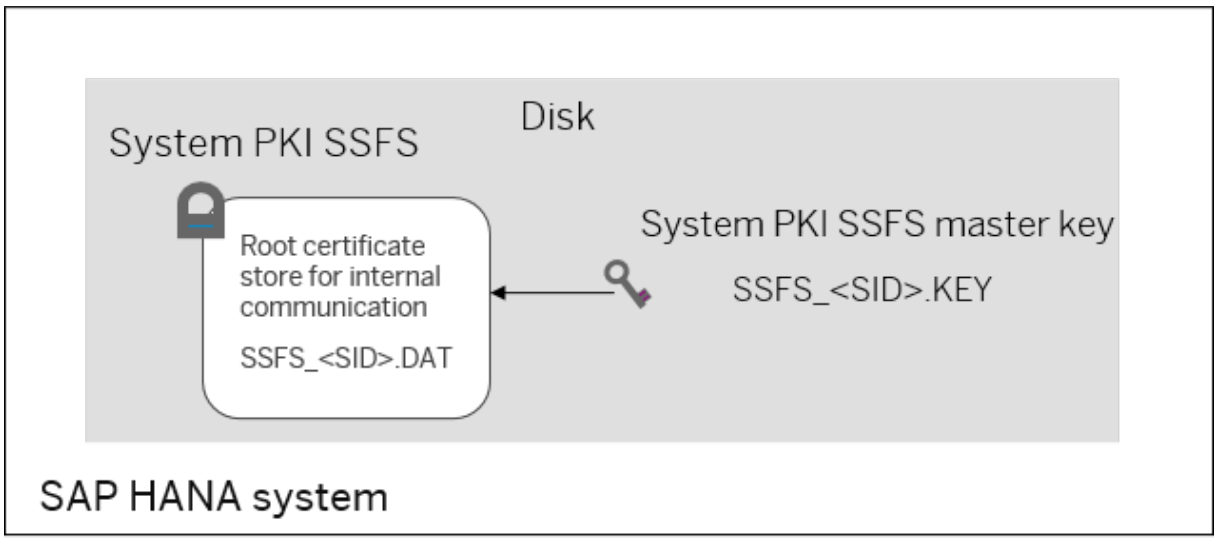
System Secure Store in the File System with Public Key Infrastructure (System PKI SSFS)

The System PKI SSFS protects the X.509 certificate infrastructure used to secure internal SSL/TLS-based communication between hosts in a multiple-host system or between processes of the individual databases in a system.

The master key of the System PKI SSFS encrypts all system-internal root certificates required for secure internal communication.

The contents of System PKI SSFS in file SSFS_.DAT are protected by an individual master key in file SSFS_.KEY. Both files belong together.

> ⓘ Note
>
> The default path of the key file of the System PKI SSFS is:
>
> /usr/sap/<sid>/SYS/global/security/rsecssfs/data/
>
> /usr/sap/<sid>/SYS/global/security/rsecssfs/key/
>
> If you change the default path, you must reconfigure it after a system rename.



Master Key Change

Unique master keys are generated for the System PKI SSFS during installation or update. If you received your system pre-installed from a hardware or hosting partner, it is recommended to change them immediately after handover to ensure they are not known outside of your organization. You can also change the master keys at any time later.

Related Information

Local Secure Store (LSS)
Secure Internal Communication
Change the SSFS Master Keys [page 505]
Internal Application Encryption Service
SAP Note 3048191
SAP Note 3544974
SAP Note 2097613

9.3.1.1.1 Change the SSFS Master Keys

The secure stores in the file system (SSFS) used by SAP HANA are protected by unique master keys, generated during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change these master keys immediately after handover to ensure that they are not known outside your organization.

> **Note**
>
> If you have replaced the instance SSFS with the local secure store (LSS), you do not have an instance SSFS master key, but you do still have a master key for the system PKI SSFS.

Prerequisites

- You have shut down the SAP HANA system.
- You have the credentials of the operating system user (`<sid>adm`) that was created when the system was installed.

Context

You change the SSFS master keys using the command line tool `rsecssfx`, which is installed with SAP HANA and available at `/usr/sap/<SID>/HDB<instance>/exe`.

Before changing the SSFS master keys, note the following:

- In a distributed SAP HANA system, every host must be able to access the file location of the instance SSFS master key.
- The SSFS master keys only have to be changed once for the whole instance and not per tenant database.
- In a system-replication setup you can change the instance SSFS master key on the primary system and the new key will be replicated to the secondary. To trigger replication you must subsequently restart the secondary system. In this case, however, you will need to copy the system PKI SSFS key and data file from the primary system to the secondary manually before registering and restarting the secondary system. Refer to the steps in the topic *Configure SAP HANA System Replication with hdbnsutil* and SAP Note 2369981 *Required configuration steps for authentication with HANA System Replication*.

> **→ Remember**
>
> In multi-tier system replication scenarios involving three systems, restart the tier-2 secondary system first, then the tier-3 secondary system. If a secondary system takes over from its replication source before the new master key has been replicated, all systems registered will use the old key from the former secondary system instead.

Procedure

1. Log on to the SAP HANA system host as the operating system user, `<sid>adm`.
2. Change the master key of the **instance SSFS**, if used, as follows:
   a. Re-encrypt the instance SSFS with a new key with the command:

```
export RSEC_SSFS_DATAPATH=/usr/sap/<SID>/SYS/global/hdb/security/ssfs
export RSEC_SSFS_KEYPATH=<path to current key file>
```

```
rsecssfx changekey $(rsecssfx generatekey -getPlainValueToConsole)
```

For script languages bash and csh the syntax is:

```
rsecssfx changekey `rsecssfx generatekey -getPlainValueToConsole`
```

> **Note**
>
> The command uses the backtick character not single quotes.

b. Configure the specified key file location in the `global.ini` configuration file at `/usr/sap/<SID>/SYS/global/hdb/custom/config/global.ini`.

If the file does not exist, create it. Add the following lines:

```
[cryptography]
ssfs_key_file_path = <path to key file>
```

> **Note**
>
> The default path of the key file is `/usr/sap/<sid>/SYS/global/hdb/security/ssfs`. If you change the default path, you may need to reconfigure it in the event of a system rename.

3. Re-encrypt the **system PKI SSFS** with a new key with the following command:

```
export RSEC_SSFS_DATAPATH=/usr/sap/<SID>/SYS/global/security/rsecssfs/data
export RSEC_SSFS_KEYPATH=<path to current key file>
rsecssfx changekey $(rsecssfx generatekey -getPlainValueToConsole)
```

> **Note**
>
> The default path of the key file is `/usr/sap/<sid>/SYS/global/security/rsecssfs/key`. If you change the default path, you may need to reconfigure it in the event of a system rename.

For script languages bash and csh the syntax is:

```
rsecssfx changekey `rsecssfx generatekey -getPlainValueToConsole`
```

> **Note**
>
> The command uses the backtick character not single quotes.

4. Restart the SAP HANA system.

Next Steps

Additional Information for Backup and Recovery

For file system-based copies of SAP HANA database installations, you must manually save and restore the instance SSFS master key file. Otherwise data loss can occur.

In regular backup and recovery scenarios, the SSFS must always be restored from a recent root key backup before a database recovery, unless:

- You have never changed the redo log encryption key.
- You are performing a recovery into the same database from which the backup was taken, and the database's secure store is intact and contains the latest root key changes.

> **Note**
>
> It is not necessary to save the system PKI SSFS key file. The system will generate a new system PKI SSFS automatically if required.

Related Information

Starting and Stopping SAP HANA Systems [page 63]
Configuring SAP HANA System Properties (INI Files) [page 126]
Server-Side Data Encryption Services [page 500]
Import Backed-Up Root Keys Before Database Recovery [page 537]
Configure SAP HANA System Replication with hdbnsutil [page 793]
SAP Note 2183624
SAP Note 2369981

9.3.1.1.2 Using the LSS with an External Key Management System

The LSS payload database (DB) contains the SAP HANA encryption root keys used for encrypting SAP HANA data, log volumes, and backups.

The body of the LSS payload DB is encrypted with a AES-256-bit key. With every save-operation, a new key for this encryption is generated and stored in a header of the same LSS payload DB. This header is encrypted. By default, the header encryption key is done with a local key chain of LSS (customer-supplied encryption keys [CSEK] scenario).

If desired, the SAP HANA/LSS can be reconfigured to connect LSS to a key in Data Custodian (DC) KMS. If this is done, then an RSA key pair managed by DC KMS is used as the header encryption key. LSS can then only open its payload DB by sending the encrypted header for decryption to DC KMS. This offers the capability to disrupt the key chain by disabling or deleting the key in KMS (customer-controlled encryption keys [CCEK] scenario). However, this makes the SAP HANA/LSS strictly dependent on the key in KMS.

> **Note**
>
> Only the SAP Data Custodian Key Management Service is supported.

When SAP DC KMS is used with the LSS, the LSS payload DB is encrypted with the public key of an asymmetric key pair whose private key is stored in DC KMS. Consequently, to read and decrypt a payload DB, access to the private key of the key pair is required.

DC KMS unreachable

If DC KMS is unreachable for some reason (for example, network issues, or a downtime of KMS), SAP HANA will in most cases continue to work, thanks to the filled caches in LSS.

Only if the LSS cache is empty (usually after a full restart of SAP HANA and LSS), the SAP HANA database can hang or even crash if KMS is not reachable. The detailed error traces of LSS should help to analyze the situation.

To verify that DC KMS is available and that there is an active key, SAP HANA regularly polls into KMS. This allows LSS to respond to the following changes:

- **Key revocation**
  If the DC KMS reports that the key has been revoked (disabled or deleted), the database is shut down and cannot be restarted or recovered from a backup (until the key is available again).

  > ⚠ Caution
  >
  > Key revocation is not trivial and should never be used to temporarily prevent access to applications. While a disabled key can be re-enabled, the cost (in terms of effort and time) of resuming application operation after a temporary key disablement may be high. Deleting a key is irreversible.

- **Key rotation**
  If DC KMS reports that the key has been rotated (for example, a new primary version of the key has been created), the currently active key management configuration is updated and LSS automatically re-encrypts the payload DB with the new key. This can be confirmed by querying the `KEY_MANAGEMENT_CONFIGURATIONS` system view: the active configuration has the new key version configured.

You can configure the polling interval using the `[cryptography] key_revocation_check_interval` in the `nameserver.ini`. The value must be between 1 and 3600 seconds. The default value is 60 seconds. If you set a value outside this range, it defaults to 60 seconds. Furthermore, if you set a value under 60 seconds, a warning is displayed that the polling rate might be too high and could lead to unnecessarily high network traffic.

Second Access Key Pair

The LSS-internal caching makes HANA/LSS partly resilient against temporary KMS outages. However, a permanent loss of the key in KMS, e.g., due to a disaster in KMS, would lead to a complete loss of the HANA database.

To also eliminate this risk, you can configure a second access key pair. You generate this second access key pair yourself.

> ⓘ Note
>
> Keep the second access key pair in a safe place in case of an emergency.

Only the public key of the second access key pair is provided to LSS, as a self-signed certificate within the external key management configuration passed to SAP HANA. SAP HANA forwards this certificate to the LSS, which then uses it to add a second header to the payload DB. This allows the Payload DB to be decrypted either regularly with the help of DC KMS (using the first header), or, in an emergency case, with the private key of the second access key pair (using the second header).

To bring SAP HANA/LSS back into an operable state after a DC KMS emergency, you need to take special action in which you provide the private key of the second access key pair to LSS, by:

- Restoring the LSS component using the `lssbackup` utility (`restoreWithSecondAccessKey`)
- Recovering an `lssbackup` as the first step in recovering an encrypted SAP HANA database (`RECOVER ENCRYPTION ROOT KEYS AND SETTINGS … SECURE STORE SECOND ACCESS`)

Related Information

Create and Manage a Key Management Configuration [page 510]
Create a Second Access Key [page 513]
Add Second Access Certificates to the Key Management Configuration [page 514]
Monitoring Key Management Configurations [page 515]
Restore a Payload Database File Backup with the Second Access Key [page 516]
LSS Backup Utility (lssbackup)
Import Backed-Up Root Keys Before Database Recovery [page 537]
RECOVER ENCRYPTION ROOT KEYS Statement (Backup and Recovery)
LSS Trace and Dump Files
SAP Note 2917358
SAP Note 2911896

9.3.1.1.2.1 Create and Manage a Key Management Configuration

A key management configuration includes a JSON document containing the configuration information required to connect to the chosen external key management service (KMS) or hardware security module (HSM).

Prerequisites

You can log on to the system database and have the system privileges `ENCRYPTION ROOT KEY ADMIN` and `DATABASE ADMIN`.

Context

You create and manage the tenant-specific configurations required to use a KMS or HSM in the system database using the SQL statements described below. Every database has a `DEFAULT` configuration, which is active by default and corresponds to no connection to an external KMS.

> ⚠ Caution
>
> After a key management configuration has been added, changed or activated, the LSS must be backed up. If automatic LSS backups are enabled (default), the required backup is always written and available for recovery. If automatic LSS backups are disabled (not recommended), this must be considered in your backup strategy. A missing backup could result in your database being unrecoverable. For more information, see *Automatic Content Backups* in the *SAP HANA Security Guide*.

For more information about the settings required by a specific KMS or HSM, see SAP Note 2917358.

Procedure

- To **add a key management configuration** for a specific database (tenant database or system database), execute the following statement:

```
ALTER DATABASE <database_name>
    ADD KEY MANAGEMENT CONFIGURATION <config_name> PROPERTIES '<settings>'
```

Parameter	Description
`<database_name>`	Name of the database
`<config_name>`	A unique name consisting of uppercase letters only, for example, 'AWS_HSM'. The name 'DEFAULT' is forbidden.
`<settings>`	A JSON document with key-value settings. The list of keys and their supported values depends on the chosen KMS or HSM. The command will fail if the specified settings do not work.

  The new configuration for the specified database is stored in the LSS tenant-specific configuration database file (`lssconfig.db`).

- To **alter a key management configuration** for a specific tenant database, execute the following statement:

```
ALTER DATABASE <database_name>
    ALTER KEY MANAGEMENT CONFIGURATION <config_name> PROPERTIES '<updates>'
```

Parameter	Description
`<database_name>`	Name of the tenant database
`<config_name>`	The name of the configuration to be modified, in upper-case letters
`<updates>`	A JSON document with the new key-value settings. The list of keys and their supported values depends on the chosen HSM or KMS. All settings not described in the update remain unchanged.

  This command will fail if the changed settings do not work.

- To **activate a key management configuration** for a specific tenant database, execute the following statement.

By activating a configuration, you can switch between different configurations.

```
ALTER DATABASE <database_name>
    ACTIVATE KEY MANAGEMENT CONFIGURATION <config_name>
```

Parameter	Description
`<database_name>`	Name of the database
`<config_name>`	The name of the configuration to be activated, in upper-case letters

The command will fail (leaving everything unchanged) if the newly activated configuration does not work.

- To **remove a key management configuration** for a specific tenant database, execute the following statement:

```
ALTER DATABASE <database_name>
    DROP KEY MANAGEMENT CONFIGURATION <config_name>
```

Parameter	Description
`<database_name>`	Name of the database
`<config_name>`	The name of the configuration to be deleted, in uppercase letters

The command will fail if the specified configuration is still active. You deactivate a configuration by activating another one.

> **Note**
>
> If you no longer want to use the LSS with a key management system at all, activate the `DEFAULT` key management configuration.

Related Information

SAP Note 2917358

ALTER SYSTEM {ADD | ACTIVATE | UPDATE | DROP} KEY MANAGEMENT CONFIGURATION Statement (System Management)

ALTER DATABASE Statement (Tenant Database Management)

9.3.1.1.2.2 Create a Second Access Key

The second access key is needed for recovering from disasters in the key management service (KMS) or hardware security module (HSM). You create a second access key by using the SAPGENPSE tool.

Procedure

> **Note**
>
> For security reasons, we recommend that you do not enter the passwords (`-x` and `-z`) on the command line. By omitting these parameters, you will be interactively prompted to enter the passwords. In this way, you avoid unintentionally leaving the passwords in the command history.

1. Create an asymmetric key pair as follows:

```
sapgenpse get_pse -a RSA:4096:SHA256 -x <second_access_key_passphrase> -p
<second_access_key>.pse -noreq CN=<host_FQDN>">
```

Parameter	Description
`<second_access_key_passphrase>`	Password for the PSE file
`<second_access_key>`	Name of the PSE file

2. Export the certificate with the public key to the file `<second_access_certificate>.pem`:

```
sapgenpse export_own_cert -p <second_access_key>.pse -x
<second_access_key_passphrase> -o <second_access_certificate>.pem
```

Parameter	Description
`<second_access_key>`	Name of the PSE file
`<second_access_key_passphrase>`	Password for the PSE file
`<second_access_certificate>`	File name of the exported certificate

3. Export the private key file as follows and then store it in a safe place.

   You can optionally add a passphrase for the exported private key interactively or by using the `-z` option. If you add a passphrase, it must be inserted whenever the private key is used. Make sure that you do not forget the passphrase, or store it in a safe place.

```
sapgenpse export_p8 -x <second_access_key_passphrase>
-z <private_key_passphrase> -p <second_access_key>
<second_access_private_key>.pem
```

Parameter	Description
`<second_access_key_passphrase>`	Password for the PSE file

Parameter	Description
`<second_access_key>`	Name of the PSE file
`<second_access_private_key>`	File name of the exported private key
`<private_key_passphrase>`	Password for the exported private key file

Once the private key and passphrase have been stored in a safe place, the PSE file can be deleted.

Related Information

Encryption Key Chain

9.3.1.1.2.3 Add Second Access Certificates to the Key Management Configuration

The public key of the second access key is provided as a self-signed certificate that is part of the external key management configuration passed to SAP HANA.

Prerequisites

You can log on to the system database and have the system privileges `ENCRYPTION ROOT KEY ADMIN` and `DATABASE ADMIN`.

Procedure

1. In the JSON document with key-value settings, enhance an already existing external key management configuration by adding the second access certificate to a JSON object with the `SET` action. For example:

```
{
"setup": {
    "configure_second_access": {
        "action": "SET",
        "certificates": [
            "-----BEGIN CERTIFICATE-----\n<certificate content base64
encoded>\n-----END CERTIFICATE-----"
            ]
        }
    },
<KMS-specific configuration goes here>
    }
}
```

- If you have already configured second access certificates for the active external key management configuration and want to add them to a new key management configuration, apply the `KEEP` action in the JSON document with key-value settings:

```
{
    "setup": {
        "configure_second_access": {
            "action": "KEEP"
        }
    }
    <KMS-specific configuration goes here>
}
```

- If you already have second access certificates for the active external key management configuration but do not want to use them in a new key management configuration, apply the `NONE` action in the JSON document with key-value settings:

```
{
    "setup": {
        "configure_second_access": {
            "action": "NONE"
        }
    }
    <KMS-specific configuration goes here>
}
```

2. Add, alter, or activate the key management configuration using the corresponding SQL statements. See *Create a Key Management Configuration*.

Results

The LSS now encrypts the body encryption key of the payload DB file twice: first with the certificate obtained from the KMS or HSM, and second with the certificate of the second access key pair. Both results are stored in the header of the payload database file. In normal operations, the LSS opens the payload DB file with the help of the KMS or HSM. Use the second access key only to recover from disasters in the KMS or HSM.

Related Information

Create and Manage a Key Management Configuration [page 510]

9.3.1.1.2.4 Monitoring Key Management Configurations

The monitoring view KEY_MANAGEMENT_CONFIGURATIONS provides information about the existing key management configurations.

It gives the following details:

Column	Description
DATABASE_NAME	Database name
CONFIGURATION_NAME	Name of the key management configuration
IS_ACTIVE	Indicates whether this configuration is used by the LSS to define access to SAP HANA's sensitive information
TYPE	Type of external key management
CLIENT_VERSION	Version of the driver software
PROPERTIES	Reduced version of the `properties` section of the `<settings>` JSON (the "credentials" object is omitted)
CONFIGURATION_ID	ID that uniquely identifies a key management configuration

The certificates configured for the second access key are shown as an additional object, "second_access", of the `<settings>` JSON, as shown below:

```
{
    "second_access": {
        "certificates": [
            "-----BEGIN CERTIFICATE-----
<certificate content base64 encoded>
-----END CERTIFICATE-----"
        ]
    }
    <KMS-specific configuration goes here>
}
```

Related Information

KEY_MANAGEMENT_CONFIGURATIONS System View

9.3.1.1.2.5 Restore a Payload Database File Backup with the Second Access Key

Use the private key file of the second access key to restore the payload database (DB) file of a tenant in an emergency. This is necessary in rare cases only.

Prerequisites

- You have operating system-level access to the SAP HANA system as the `<sid>crypt` user.
- When the LSS backup was created, the active key management configuration contained a second access key. See *Create a Second Access Key* and *Add Second Access Certificates to the Key Management Configuration*.

Procedure

1. Copy the private key file from its safe place to a directory that is readable and writable by `<sid>crypt`.
2. Restore the tenant's payload DB file using the `lssbackup` utility:

```
lssbackup -t <tenant> restoreWithSecondAccessKey <backupfile> payload
<configbackup> <privatekeyfile>
```

Parameter	Description
`<tenant>`	Name of the tenant **Note** The system database and tenant databases are both tenants in the LSS context.
`<backupfile>`	Backup file of the payload DB file
`<configbackup>`	Backup file of the `config` DB file
`<privatekeyfile>`	The file name of the PEM-encoded private key file of the second access key that was used in the LSS backup

Note

The payload DB file cannot be restored from a single backup file that contains both the contents of the payload DB file and the config DB file.

3. Enter the LSS backup password.
4. Enter the passphrase for the private key if one has been set.

Results

The restore operation uses the second access key provided as a private key file and the corresponding passphrase to decrypt and restore the contents of the payload DB file.

Related Information

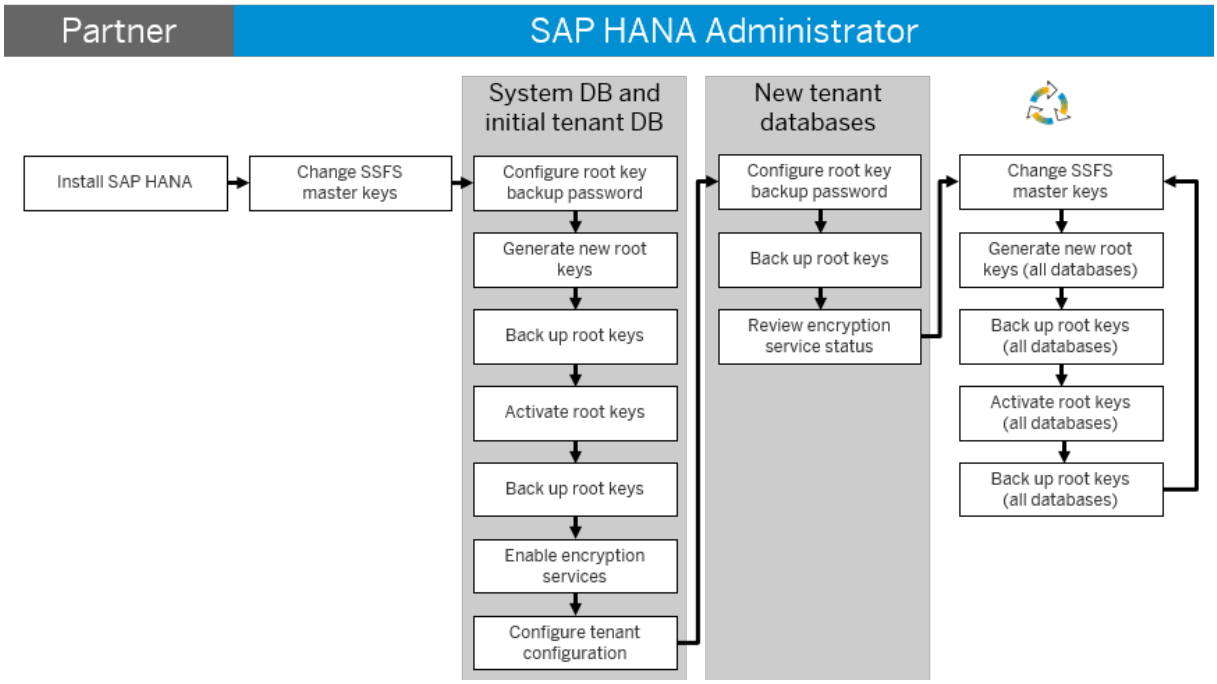
LSS Backup Utility (lssbackup)

9.3.1.2 Encryption Configuration

We recommend that you configure data encryption immediately after handover of your system from a hardware or hosting partner.

First-Time Configuration

The following figure shows the recommended process for configuring encryption in your SAP HANA system for the first time.



Immediately after system handover from your hardware or hosting partner, perform the following high-level steps.

Location	Steps	More Information
On the SAP HANA server	Change the master keys of the instance SSFS, if used, and the system PKI SSFS.(1)	Unique master keys are generated for the instance SSFS, if used, and the system PKI SSFS during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change the master keys any time later.

Location	Steps	More Information
In the system database	1. Set the password for the root key backup for the system database. This password is required to securely back up root keys and subsequently restore backed-up root keys during data recovery.(2)	**⚠ Caution** The password is stored in the secure store along with the other root keys and used whenever you create a backup of the encryption root keys. The password should also be stored in a separate safe location. You will need to enter it to restore the secure store content before a database recovery. Losing this password may result in the database being unrecoverable.
	2. Change the encryption root keys for all encryption services (data volume encryption, redo log encryption, data and log backup encryption, and internal application encryption) in the system database.(3) 1. Generate new root keys. 2. Back up all root keys to a root key backup file in a secure location. 3. Activate the new root keys. 4. Back up all root keys.	Unique root keys are generated during installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change root keys any time later. You must back up all keys after you generate or activate a key of any type. This ensures that you always have an up-to-date backup of your root keys available for recovery. **⚠ Caution** Store the root key backup file in a safe location. Losing this file may result in the database being unrecoverable.
	3. Enable the required encryption services: Data volume encryption, redo log encryption, and data and log backup encryption in the system database.(4)	Although SAP HANA provides you with the flexibility to encrypt data volumes, redo logs, and backups independently of each other, if you require full protection in the persistence layer, we recommend that you enable all services. It is not necessary to enable the internal application encryption service explicitly. It is available automatically to requesting applications.

Location	Steps	More Information
	4. Configure how you want encryption to be handled in new tenant databases. You can do so with the following parameters in the `database_initial_encryption` section of the `global.ini` configuration file: • `persistence_encryption` (default: off) • `log_encryption` (default: off) • `backup_encryption` (default: off) • `encryption_config_control` (default: local_database)	By default, all encryption services are disabled in new tenant databases and only tenant database administrators can enable them. See *Encryption Configuration Control*.
In the first tenant database (if automatically created during installation)	1. Set the password for the root key backup for the first tenant database.(2)	The password is required to securely back up root keys and subsequently restore backed-up root keys during data recovery.
	2. Change the encryption root keys for all encryption services in the first tenant database.(3)	See step 2 above (system database).
	3. Enable the required encryption services in the first tenant database.(4)	By default, only the tenant database administrator can do this in the tenant database. See *Encryption Configuration Control*.
In subsequent tenant databases	1, Set the password for the root key backup for the tenant database.(2)	The password is required to securely back up root keys and subsequently restore backed-up root keys during data recovery.
	2. Back up all root keys to a root key backup file in a secure location.	It is not necessary to change the root keys in new tenant databases. Unique root keys are generated on database creation and cannot be known outside of your organization. **⚠ Caution** Store the root key backup file in a safe location. Losing this file may result in the database being unrecoverable.

Location	Steps	More Information
	3. Change the status of encryption services in the tenant database if required.[4] Encryption services are configured in line with the parameters in the `database_initial_encryption` section of the `global.ini` configuration file as described above.	Who can enable or disable encryption services depends on how the parameter `encryption_config_control` is configured. See *Encryption Configuration Control*.
During operation	• Periodically change the master keys of the instance SSFS, if used, and the system PKI SSFS in line with your security policy. • Periodically change the encryption root keys in all databases in line with your security policy.	

[1] If the instance SSFS is used in a system-replication configuration, you change the instance SSFS master key on the primary system. To trigger replication of the new key to the secondary system, you must subsequently restart the secondary system. In multi-tier system replication scenarios involving three systems, restart the tier-2 secondary system first, then the tier-3 secondary system. If a secondary system takes over from its replication source before the new master key has been replicated, all systems registered will use the old key from the former secondary system instead.

[2] In a system-replication configuration, set the root key backup password in the primary system only. The password will be propagated to all secondary systems. The secondary systems must be running and replicating.

[3] In a system-replication configuration, change root keys in the primary system only. New keys will be propagated to all secondary systems. The secondary systems must be running and replicating.

[4] In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Configuration After Update from a Single-Container System

If you updated from a single-container system, your system has a system database and one tenant database. The existing data encryption configuration is retained. Note the following:

- The SSFS master keys for the system remain unchanged and the instance SSFS is still used.
- Existing encryption root keys are the encryption root keys of the tenant database. The update process generates new unique root keys for the system database.
- If a root key backup password existed before update, it is the root key backup password of the tenant database. The system database will not have a root key backup password set. You must set this password manually after the update.
- Encryption services that were enabled before update are enabled in both the system database and the tenant database.

Related Information

Change the SSFS Master Keys [page 505]
Set the Root Key Backup Password
Changing Encryption Root Keys [page 525]
Enable Encryption [page 533]
Encryption Configuration Control [page 522]
Encryption Key Management

9.3.1.2.1 Encryption Configuration Control

You can enable or disable the encryption of data and log volumes, and data and log backups in a new SAP HANA database or in an existing operational database. For a tenant database, you need to know whether encryption configuration is controlled by the tenant database or the system database.

Ownership of Encryption Control

By default, encryption configuration is controlled by the tenant database, but the control can be switched to the system database, or the system database can switch control back to the tenant database.

To see which database is controlling encryption configuration for a tenant database, you can query the system view SYS.M_ENCRYPTION_OVERVIEW. From the system database, you can query the system view SYS_DATABASES.M_ENCRYPTION_OVERVIEW.

Encryption Control in New Tenant Databases

When a new tenant database is created, the `encryption_config_control` parameter in the `database_initial_encryption` section of the `global.ini` configuration file in the system database determines whether encryption configuration is controlled by the tenant database or the system database. You can use this parameter to configure encryption control for new tenant databases:

- If the value of this parameter is `local_database` (default), then only the tenant database administrator can enable or disable encryption from the tenant database.
- If the value is `system_database`, then only the system database administrator can enable or disable encryption from the system database.

Switching Encryption Control in Existing Tenant Databases

If the tenant database controls encryption configuration, the tenant database administrator can hand over this control to the system administrator by executing the following ALTER SYSTEM statement:

```
ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY SYSTEM DATABASE
```

If the system database controls encryption configuration, the system database administrator can hand it over to the tenant database administrator by executing the following ALTER DATABASE statement:

```
ALTER DATABASE <database_name> ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL
DATABASE
```

For simplicity, the system database administrator can hand over control to all tenants at once by executing the following statement:

```
ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASES
```

Controlling Data Volume Re-encryption Performance

When enabling or disabling data volume encryption, the SAP HANA database re-encrypts or decrypts data volumes through background savepoint cycles. You can control and monitor the performance of this process using configuration parameters.

Configuration Parameters

The following parameters control the re-encryption process and can be set online without requiring a system restart:

- `data_volume_encryption_max_log_space_used_before_throttle_percent` (Default: 5%)
  - Controls when the re-encryption job throttles based on log space usage
  - When log space usage exceeds this percentage, the job slows down to prevent log overflow
- - `data_volume_encryption_max_pages_per_savepoint` (Default: varies by system)
    - Controls how many pages are processed per savepoint cycle
    - Higher values speed up re-encryption but may impact system performance

Setting Parameters

Sample Code

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM')
SET ('persistence',
'data_volume_encryption_max_log_space_used_before_throttle_percent') =
'<value>'
WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM')
SET ('persistence', 'data_volume_encryption_max_pages_per_savepoint') =
'<value>'
WITH RECONFIGURE;
```

Monitoring Re-encryption Progress

Monitor the progress using the following statement:

Sample Code

```
SELECT * FROM M_DATA_VOLUME_PAGE_STATISTICS_RESET;
```

Key columns to monitor:

- `ENCRYPTED_SIZE` - Size of encrypted data
- `NOT_ENCRYPTED_SIZE` - Size remaining to be encrypted
- Progress can be calculated as: (ENCRYPTED_SIZE / (ENCRYPTED_SIZE + NOT_ENCRYPTED_SIZE)) * 100

Related Information

ALTER SYSTEM ENCRYPTION CONFIGURATION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)
M_ENCRYPTION_OVERVIEW System View
SAP Note 3659679
SAP Note 3555251

9.3.1.3 Set the Root Key Backup Password

The root key backup password is required to securely back up the root keys of the database and subsequently to restore the backed-up root keys during data recovery.

Prerequisites

You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Procedure

Set the root key backup password with the following SQL statement.

```
ALTER SYSTEM SET ENCRYPTION ROOT KEYS BACKUP PASSWORD <passphrase>
```

The length and layout of the password must be in line with the database's password policy.

> ⚠ Caution
>
> If the root key backup already has a password, it will be overwritten.

> ⓘ Note
>
> In a system-replication configuration, set the root key backup password in the primary system only. The password will be propagated to all secondary systems. The secondary systems must be running and replicating.

Results

The password is set and stored in the secure store together with the SAP HANA encryption root keys and encryption-related configuration. You must provide this password to import and export root keys from the backup into the database before starting a database recovery. All root key backups taken after the password is set use this password to protect the backup files.

> ⓘ Note
>
> If you are using the local secure store (LSS), the password set is used to fully backup and recover the SAP HANA database, including the secure store.

For more information about root key backups, see the *SAP HANA Security Guide*. For more information about setting the root key backup password using the SAP HAN cockpit, see the SAP HANA cockpit documentation.

> ⚠ Caution
>
> The password should also be stored in a separate safe location. You will need to enter it to restore the secure store content before a database recovery. Losing this password may result in the database being unrecoverable.

> → Tip
>
> To verify that the password you have is the same as the one that the system uses when creating new root key backups, use the statement `ALTER SYSTEM VALIDATE ENCRYPTION ROOT KEYS BACKUP PASSWORD <passphrase>`.

Related Information

Password Policy Configuration Options
Root Key Backup
ALTER SYSTEM VALIDATE ENCRYPTION ROOT KEYS BACKUP PASSWORD Statement (System Management)
Set the Root Key Backup Password

9.3.1.4 Changing Encryption Root Keys

Unique root keys are generated during installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change root keys any time later.

Change the root keys for the following encryption services immediately after handover of your system and periodically during operation:

- Data volume encryption
- Redo log encryption

- Data and log backup encryption
- Internal application encryption

It is important to always change encryption root keys as follows:

1. Generate new root keys.
2. Activate new root keys.

> ⚠ Caution
>
> You do not need to back up the root keys after they haven been generated or activated since they are included in the SAP HANA database self-contained backups.

> ⓘ Note
>
> In a system-replication configuration, change root keys in the primary system only. New keys will be propagated to all secondary systems. The secondary systems must be running and replicating.

9.3.1.4.1 Generate New Root Keys

The first step in changing encryption root keys is to generate new root keys.

Prerequisites

- You are connected to the tenant database requiring the root key change.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Procedure

Generate new root keys for all encryption services using the following SQL statements:

Encryption Service	Statement
Data volume encryption	`ALTER SYSTEM PERSISTENCE ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`
Redo log encryption	`ALTER SYSTEM LOG ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`
Data and log backup encryption	`ALTER SYSTEM BACKUP ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`

Encryption Service	Statement
Internal application encryption	`ALTER SYSTEM APPLICATION ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`

New root keys for all encryption services are generated. To verify the creation of new root keys, query the system view `ENCRYPTION_ROOT_KEYS` and check the status of the latest entries for each root key type.

Related Information

ENCRYPTION_ROOT_KEYS System View
ALTER SYSTEM PERSISTENCE ENCRYPTION Statement (System Management)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
ALTER SYSTEM APPLICATION ENCRYPTION Statement (System Management)

9.3.1.4.2 Back Up Root Keys

After you have generated or activated new encryption root keys or created a new tenant database with new root keys, you must back up all root keys.

Prerequisites

- The external location to which you plan to save the backup is accessible.
- You have set the root key backup password. See *Set the Root Key Backup Password*.
- If using the SQL statement `BACKUP ENCRYPTION ROOT KEYS`:
  - The system database has encryption configuration control. See *Encryption Configuration Control*.
  - You have the system privilege `ENCRYPTION ROOT KEY ADMIN`
  - You have the system privilege `BACKUP ADMIN` or `BACKUP OPERATOR` if backing up the system database root keys, or the system privilege `DATABASE BACKUP ADMIN` or `DATABASE BACKUP OPERATOR` if backing up a tenant database's root keys.
- If using the SQL function `ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS` or `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`, you have the system privilege `ENCRYPTION ROOT KEY ADMIN`.
- If using `hdbnsutil`, you have the credentials of the operating system user (`<sid>adm`).
- If using `hdbnsutil`, you know the ID of the database whose root keys you want to back up. You can determine the IDs of all tenant databases by executing the following SQL command in the system database:

```
SELECT DATABASE_NAME,
    CASE WHEN (DBID = '' AND
```

```
    DATABASE_NAME = 'SYSTEMDB')
    THEN 1
    WHEN (DBID = '' AND
    DATABASE_NAME <> 'SYSTEMDB')
    THEN 3
    ELSE TO_INT(DBID)
    END DATABASE_ID
FROM (SELECT DISTINCT DATABASE_NAME, SUBSTR_AFTER (SUBPATH,'.') AS DBID FROM
SYS_DATABASES.M_VOLUMES)
```

Context

A backup of encryption root keys must be available at an external location to ensure recovery is possible in certain scenarios.

> ⚠ Caution
>
> Store both the root key backup and the password required to read it in a secure location. Losing the backup or the password may result in the database being unrecoverable.

You can back up root keys in a number of ways.

Option	Execution Location	Database Mode	Backup File Location
SQL statement `BACKUP ENCRYPTION ROOT KEYS`	System database	Tenant database can be online or offline; system database must be online	File system on the database server. The file needs to be manually copied to a secure external location.
SQL extraction function `ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS`[1]	Applicable database	Database must be online	CLOB result. The result needs to be manually copied and saved to a file at a secure external location.
SQL extraction function `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`	Applicable database or system database	If executed from the tenant database: Database must be online If executed from the system database: Database can be online or offline; system database must be online	CLOB result. The result needs to be manually copied and saved to a file at a secure external location.
SAP HANA cockpit[2]	Applicable database	Database must be online	Local file system
hdbnsutil tool[3]	SAP HANA server	Database can be online or offline	File system on the database server. The file needs to be manually copied to a secure external location.

> ⓘ Note
>
> - [1]If the local secure store (LSS) is being used in conjunction with an external key management system, then you cannot back up the root keys using

`ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS` since it does extract the keys in the required format. You must use `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`.

- (2)The SAP HANA cockpit can be used to back up root keys for both the local secure store (LSS) and the secure store in the file system (SSFS).
- (3)If the local secure store (LSS) is being used in conjunction with an external key management system, then you cannot back up the root keys using `hdbnsutil` since a separation of duties between operating system administrator and key administrator is not possible.

Procedure

1. Back up the root keys of a tenant database using one of the following methods:

Option	Description
SQL statement `BACKUP ENCRYPTION ROOT KEYS`	In the system database, execute the SQL statement: `BACKUP ENCRYPTION ROOT KEYS [ <root_keytype_list> ] [ FOR <database_name> ] USING FILE ('<root_key_backup_definition_file>')` • `<database_name>` is the name of the tenant database. • The `<root_keytype_list>` option lists the root key types and accepts the values `PERSISTENCE`, `LOG`, `BACKUP`, and `APPLICATION`. If you do not specify any value for `<root_keytype_list>`, all root key types are backed up. **Note** If the LSS is being used in conjunction with an external key management system, then you cannot back up individual root key types. A full LSS backup is required. Therefore, in this case you must not use the `<root_keytype_list>` option. • `<root_key_backup_definition_file>` specifies the backup file. The name of the backup file is constructed from the following elements: <prefix>_<database_name>_rootkeys (for example: '/backup/HDB/rootkeys/backup001') – see BACKUP ENCRYPTION ROOT KEYS Statement. The file is written to the file system on the database server.
SQL extraction function `ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS`	1. In the tenant database whose keys are being extracted, execute the following SQL statement: `SELECT ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS`

Option	Description
	```('PERSISTENCE, APPLICATION, BACKUP, LOG') FROM DUMMY``` The result of this command is a character large object (CLOB). 2. Copy the CLOB result and save it to a file at a secure external location. We recommend the file extension `.rkb`.
SQL extraction function `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`	1. Depending on which database has encryption configuration control, execute the following SQL statement in either the tenant database or the system database: ```SELECT ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE ('<database_name>') FROM DUMMY``` The result of this command is a character large object (CLOB). 2. Copy the CLOB result and save it to a file at a secure external location. We recommend the file extension `.rkb`.
SAP HANA cockpit	You can back up root keys using the *Data Encryption* app of the SAP HANA cockpit. For more information, see the SAP HANA cockpit documentation.
`hdbnsutil` **tool**	**Note** If the LSS is being used in conjunction with an external key management system, then you cannot back up the root keys using `hdbnsutil`. 1. Log on to the SAP HANA server as operating system user `<sid>`adm. 2. Back up the new keys with the following command: ```hdbnsutil -backupRootKeys <path_filename> --dbid=<dbid> --type='ALL'``` • `<dbid>` is the tenant database ID. • The `<type>` option is the root key type and accepts the values `PERSISTENCE`, `LOG`, `BACKUP`, and `APPLICATION`. The value `ALL` specifies that root keys of all types are backed up. If you do not specify any value for `<type>`, all root key types are backed up. • `<path_filename>` specifies the path where the backup should be stored plus the name of the file (using the 'rkb' file extension is optional). The file is written to the file system on the database server.

2. Save the root key backup file to a secure location.

> ⚠ Caution
>
> Store the root key backup file in a safe location. If this file is lost, it may not be possible to recover the database.

3. Optional: To ensure that the backup file can be recovered, validate the password for the root key backup file.

> ⓘ Note
>
> Each root key backup file created is unique. You must validate it as described here and not through comparison with other files (if multiple backups have been done).

To validate the backup file log on to the SAP HANA server as operating system user `<sid>adm`, and use the following command in the `hdbnsutil` tool:

```
hdbnsutil -validateRootKeysBackup <path_filename> [--password=<passphrase>]
```

> → Recommendation
>
> We recommend that you do not enter the password on the command line. You will be interactively prompted to enter it. In this way, you avoid unintentionally leaving the password in the command history and making it visible in process monitoring tools provided by the operating system.

Related Information

Encryption Configuration Control [page 522]
Root Key Backup
ALTER SYSTEM SET ENCRYPTION ROOT KEYS BACKUP PASSWORD Statement (System Management)
BACKUP ENCRYPTION ROOT KEYS Statement (Backup and Recovery)
ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS Function (Security)
Back Up Root Keys (SAP HANA Cockpit)
Set the Root Key Backup Password [page 524]

9.3.1.4.3 Activate Root Keys

Activate new encryption root keys so that they can be used to encrypt new data.

Prerequisites

- You are connected to the tenant database requiring the root key change.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

- You have backed up the new encryption root keys. You can verify whether or not root keys are backed up by querying the system view `ENCRYPTION_ROOT_KEYS`.

Procedure

Activate the new root keys by executing the following SQL statements:

Encryption Service	Statement
Data volume encryption	`ALTER SYSTEM PERSISTENCE ENCRYPTION ACTIVATE NEW ROOT KEY`
Redo log encryption	`ALTER SYSTEM LOG ENCRYPTION ACTIVATE NEW ROOT KEY`
Data and log backup encryption	`ALTER SYSTEM BACKUP ENCRYPTION ACTIVATE NEW ROOT KEY`
Internal application encryption	`ALTER SYSTEM APPLICATION ENCRYPTION ACTIVATE NEW ROOT KEY`

If encryption is enabled, new data is encrypted with the new root keys.

> **Note**
>
> It is not necessary to enable the internal application encryption service explicitly. It is available automatically to requesting applications.

Related Information

Generate New Root Keys [page 526]
Back Up Root Keys [page 527]
ENCRYPTION_ROOT_KEYS System View
ALTER SYSTEM PERSISTENCE ENCRYPTION Statement (System Management)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
ALTER SYSTEM APPLICATION ENCRYPTION Statement (System Management)

9.3.1.5 Enable Encryption

You can enable data volume encryption, redo log encryption, and encryption of data and log backups in a new SAP HANA database or in an existing operational database.

Prerequisites

- To enable encryption for a tenant database, you know whether encryption configuration is controlled by the tenant database or the system database:
  - If the tenant database controls encryption configuration, encryption can only be enabled or disabled directly in the tenant database and not from the system database.
  - If the system database controls encryption configuration, encryption can only be enabled or disabled using SQL from the system database, with a user that has the system privilege `DATABASE ADMIN`.

  See *Encryption Configuration Control*.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.
- If necessary, you have changed and backed up the encryption root keys. See *Changing Encryption Root Keys*.

Context

It is recommended that you enable encryption in the system database and the tenant databases when they are created. In this way, you ensure that all the pages are encrypted. If you received SAP HANA from a hardware or hosting partner, you should enable encryption after handover and before importing your sensitive data.

If you enable encryption in an operational database, only the pages in use in the data volumes are encrypted. Pages in data volumes that are not in use may still contain old content, and are only overwritten and encrypted over time. This means that your data in data volumes will only be fully encrypted after some delay. In addition, only redo log entries that are created after encryption is enabled are encrypted. Redo log files that were created before encryption was enabled are not encrypted. Although encryption can be switched on at any point in time, unencrypted data can remain on disk. If this is not wanted, you will need to install a new database on a fresh hard drive, activate encryption, import a backup, and low-level erase the old disks.

You can enable encryption of full data backups, delta data backups, and log backups in the database at any time.

> → Recommendation
>
> Although SAP HANA provides you with the flexibility to encrypt data volumes, redo logs, and backups independently of each other, if you require full protection in the persistence layer, we recommend that you enable all services.

Procedure

> **Note**
>
> In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Enable the required encryption service using the SAP HANA cockpit (tenant database control only) or the following SQL statements:

If the tenant database has control	• Data volume encryption
	`ALTER SYSTEM PERSISTENCE ENCRYPTION ON`
	• Redo log encryption
	`ALTER SYSTEM LOG ENCRYPTION ON`
	• Backup encryption
	`ALTER SYSTEM BACKUP ENCRYPTION ON`

If the system database has control	• Data volume encryption
	`ALTER DATABASE <database_name> PERSISTENCE ENCRYPTION ON`
	• Redo log encryption
	`ALTER DATABASE <database_name> LOG ENCRYPTION ON`
	• Backup encryption
	`ALTER DATABASE <database_name> BACKUP ENCRYPTION ON`

Results

Data volume encryption and redo log encryption

All data persisted to data volumes is encrypted and all future redo log entries persisted to log volumes are encrypted.

Backup encryption

Backup encryption is enabled. Subsequent log backups, as well as full backups and delta data backups will be encrypted.

> **Note**
>
> If backup encryption is active, a data snapshot is **not automatically encrypted**. For more information, see *Backup Encryption*.

Related Information

Encryption Configuration [page 518]
Encryption Configuration Control [page 522]
Changing Encryption Root Keys [page 525]
Data and Log Volume Encryption
Backup Encryption
Enable Encryption (SP HANA Cockpit)
M_ENCRYPTION_OVERVIEW System View
ALTER SYSTEM PERSISTENCE ENCRYPTION Statement (System Management)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
SAP Note 2159014

9.3.1.6 Disable Encryption

Disabling data volume encryption triggers the decryption of all encrypted data. Newly persisted data is not encrypted. Disabling redo log encryption makes sure that future redo log entries are not encrypted when they are written to disk.

Prerequisites

- To disable encryption for a tenant database, you know whether encryption configuration is controlled by the tenant database or the system database:
  - If the tenant database controls encryption configuration, encryption can only be enabled or disabled directly in the tenant database and not from the system database.
  - If the system database controls encryption configuration, encryption can only be enabled or disabled using SQL from the system database, with a user that has the system privilege `DATABASE ADMIN`.

  See *Encryption Configuration Control*.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Procedure

> **Note**
>
> In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Disable the required encryption service using the SAP HANA cockpit (tenant database control only) or the following SQL statements:

If the tenant database has control	• Data volume encryption
	`ALTER SYSTEM PERSISTENCE ENCRYPTION OFF`
	• Redo log encryption
	`ALTER SYSTEM LOG ENCRYPTION OFF`
	• Backup encryption
	`ALTER SYSTEM BACKUP ENCRYPTION OFF`

If the system database has control	• Data volume encryption
	`ALTER DATABASE <database_name> PERSISTENCE ENCRYPTION OFF`
	• Redo log encryption
	`ALTER DATABASE <database_name> LOG ENCRYPTION OFF`
	• Backup encryption
	`ALTER DATABASE <database_name> BACKUP ENCRYPTION OFF`

Results

Data volume encryption

Data starts being decrypted in the background. Depending on the size of the SAP HANA database, this process can be very time consuming. Only after this process has completed is all your data decrypted. Newly persisted data is not encrypted.

Redo log encryption

New redo log entries are not encrypted. Existing redo log entries are not decrypted. Log entries will only be fully unencrypted when all encrypted entries have been overwritten.

Backup encryption

New data backups, delta backups, and log backups are not encrypted. On an unencrypted data volume, data snapshots are also unencrypted.

Related Information

Encryption Configuration [page 518]
Encryption Configuration Control [page 522]
Disable Encryption (SP HANA Cockpit)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)

9.3.1.7 Import Backed-Up Root Keys Before Database Recovery

Before performing a recovery from encrypted data and log backups, you must import the backed-up root keys.

Prerequisites

- You have the credentials of the operating system user (`<sid>adm`).
- You can log on to the system database and have the system privilege `DATABASE STOP`.
- The location of the root key backup file is accessible.
- If using SQL to import keys: You have the system privilege `DATABASE RECOVERY OPERATOR` or `DATABASE ADMIN` and `ENCRYPTION ROOT KEY ADMIN` in the system database.
- If using `hdbnsutil` to import keys: You know the ID of the database whose root keys you want to import. You can determine the IDs of all tenant databases by executing the following SQL command in the system database:

```
SELECT DATABASE_NAME,
    CASE WHEN (DBID = '' AND
    DATABASE_NAME = 'SYSTEMDB')
    THEN 1
    WHEN (DBID = '' AND
    DATABASE_NAME <> 'SYSTEMDB')
    THEN 3
    ELSE TO_INT(DBID)
    END DATABASE_ID
FROM (SELECT DISTINCT DATABASE_NAME, SUBSTR_AFTER (SUBPATH,'.') AS DBID FROM
SYS_DATABASES.M_VOLUMES);
```

Context

Before you recover an encrypted database, you must first import backed-up root keys to initialize the secure store. When the instance SSFS (secure store in the file system) is used as the secure store, the set of keys from the root key backup file replaces the set of keys in the instance SSFS.

If the LSS is used with an external KMS (key management system), backed-up root keys can also be recovered using a second access key pair in an emergency situation. This is only necessary if the private key stored in the KMS has been lost, damaged, or becomes unusable for some reason.

You can import root keys in the following ways:

Option	Execution Location	Database Mode	Scope
SQL statement `RECOVER ENCRYPTION ROOT KEYS`	System database	The tenant database must be offline.	Encryption root keys can only be recovered for tenant databases.
hdbnsutil tool*	SAP HANA server	The tenant database must be offline.	Encryption root keys can be recovered for tenant databases and system databases.

> **Note**
>
> *You cannot recover root keys using `hdbnsutil` if LSS is used in conjunction with an external KMS.

Procedure

1. Log on to the SAP HANA server as operating system user `<sid>`adm.
2. Validate that you have the password for the root key backup file.

   To validate the SSFS format root key backup file format, execute:

   ```
   cd /usr/sap/<sid>/<HDBinstance_no>/exe
   ./hdbnsutil -validateRootKeysBackup <filename> [--password=<password>]
   ```

3. In the system database, stop the tenant database to be recovered.

   You can do this in the SAP HANA cockpit or by executing the statement `ALTER SYSTEM STOP DATABASE <database_name>`.
4. Import backed-up root keys using one of the following methods:
   - Using the SQL statement `RECOVER ENCRYPTION ROOT KEYS`. The input backup file format can be SSFS format root keys backup.
     - To recover keys from the SSFS (not connected to an external KMS), execute:

       ```
       RECOVER ENCRYPTION ROOT KEYS (<root_keytype_list>) FOR <database_name>
       USING FILE <root_key_backup_definition_file> PASSWORD <password>
       ```

       - `<database_name>` is the name of the tenant database.

- `<root_keytype_list>` (optional) specifies the root key types and accepts the values `PERSISTENCE`, `LOG`, `BACKUP`, and `APPLICATION`. If you do not specify any value for `<root_keytype_list>`, all root key types are imported.
- `<root_key_backup_definition_file>` specifies the name of the SSFS root key backup file in the file system (without the trailing _rootkeys).

  > **Note**
  >
  > If an LSS format backup file is provided the statement will throw an exception.

- `<password>` is the root key backup password.

- Using the `hdbnsutil` program

  > **Note**
  >
  > You cannot recover a root key backup using `hdbnsutil` if the LSS is used in conjunction with an external KMS.

  Execute the following command to recover the SSFS format root key backup:

  ```
  cd /usr/sap/<sid>/<HDBinstance_no>/exe
  ./hdbnsutil -recoverRootKeys <filename> --dbid=<dbid> --
  password=<password> --type=ALL
  ```

  - `<dbid>` is the tenant database ID.
  - `<type>` import the backed-up root keys using either the option is the root key type and also accepts the values `PERSISTENCE`, `LOG`, `BACKUP`, and `APPLICATION`. The value `ALL` specifies that root keys of all types are imported. If you do not specify any value for `<type>`, all key types are imported.
  - `<password>` is the root key backup password.

  > **→ Recommendation**
  >
  > We recommend that you do not enter the password on the command line (`--password`). By omitting this parameter, you will be interactively prompted to enter it.

  > **Note**
  >
  > If you have backed-up root keys to different files, for example according to root key type, you need to execute the command several times.

Related Information

Stop a Tenant Database
Prerequisites: Recovering an Encrypted SAP HANA Database
Using the LSS with an External Key Management System [page 508]
Recovering an SAP HANA Database
RECOVER ENCRYPTION ROOT KEYS Statement (Backup and Recovery)

9.3.2 Client-Side Data Encryption

Client-side data encryption is a column-level data encryption capability managed by the client driver.

It provides a separation between those who own the data (and can view it) and those who manage the data (but should have no access), and delivers a built-in protection of sensitive data from other third-party database administrators and cloud administrators. With client-side encryption, table columns containing sensitive data (credit card numbers, for instance) are encrypted using an encryption key that is accessible only to the client. Column data is encrypted and decrypted only on the client-driver, allowing the applications to read and write data in cleartext form.

Client-side encryption uses both symmetric and asymmetric encryption. Sensitive column data is encrypted with a symmetric column encryption key (CEK) which is encrypted using an asymmetric client key pair (CKP). CEKs are encrypted and stored on the SAP HANA server. The public key of the CKP is stored on the SAP HANA server and in the `hdbkeystore` (a secure key store) on the client's local machine. The private key of the CKP is stored only in the `hdbkeystore` on the client's local machine. Key generation and data encryption and decryption happens on the client driver only; SAP HANA server only stores the encrypted keys and encrypted data.

To access the encrypted data, an application must use a client driver that supports client-side encryption and the client must have access to the CEK that encrypts the column. When writing or reading encrypted data to or from the server, the application must use a prepared statement.

Client-side data encryption supports two types of encryption – non-deterministic (or randomized) and deterministic. Choose the encryption algorithm based on the intended use of the data.

Client-side data encryption also supports key rotation for column encryption keys (CEKs) and client key pairs (CKPs). Keys rotation decreases the risk of keys being breached and ensures users' data confidentiality and security. SAP Common Crypto Library (`sapcrypto.dll`) provides data encryption capabilities for client-side encryption. Client-side encryption is supported on JDBC and ODBC (SQLDBC) client drivers available with the SAP HANA client.

> **Note**
>
> Applications developed using SAP HANA Extended Applications Services, classic model (XS classic) do not use the SAP HANA client.

For more conceptual and procedural information about how to configure client-side data encryption, see the *SAP HANA Client-Side Data Encryption Guide*.

Related Information

SAP HANA Client-Side Data Encryption Guide

9.3.3 Use FIPS 140-2 Certified Cryptographic Kernel in CommonCryptoLib

The SAP Cryptographic Library, CommonCryptoLib, supports a FIPS 140-2 compliant cryptographic kernel module, which must be enabled if required.

Prerequisites

CommonCryptoLib patch level 8.4.37 or higher is in use. You can check the version with the following statement: `SELECT * FROM "SYS"."M_HOST_INFORMATION" WHERE KEY LIKE 'crypt%';`

> **Note**
>
> This statement also shows current version information of your FIPS-compliant crypto kernel if already enabled. If FIPS mode is disabled, the version number is `none`.

Procedure

1. In the system database, set the value of the parameter `[cryptography] ccl_fips_enabled` in the `global.ini` configuration file to `true`.
2. Restart the database.

Results

The FIPS 140-2 certified crypto kernel, `libslcryptokernel`, is used instead of the built-in crypto kernel, libsapcrypto.so.

If `libslcryptokernel` is not a FIPS 140-2 certified one, the initialization of the library will fail. This means that SAP HANA server processes will not start because of dependent errors in other security functions, for example license errors, SSL errors, and so on.

Related Information

Modify a System Property in SAP HANA Cockpit
Start a Tenant Database [page 87]
Stop a Tenant Database
Cryptographic Service Provider
SAP Note 2093286
SAP Note 2117112

9.4 Managing Client Certificates

SAP HANA uses X.509 client certificates as the basis for securing internal and external communication channels, as well as for several user authentication mechanisms. Certificates can – or in some case must – be stored and managed in the SAP HANA database. For some purposes, files stored in the file system are possible.

We recommend storing and managing the certificates required for trust validation where possible in the database and using the SAP HANA cockpit to manage the full workflow. For more information, see the SAP HANA cockpit documentation.

In some cases, certificates must be stored and managed in trust and key stores located in the file system, in so-called personal security environments or PSEs, for example the certificates required to secure internal communication channels using the system public key infrastructure (system PKI). For more information, see the section on certificate management in the *SAP HANA Security Guide*.

Replacing Default Server Certificates with Trusted Certificates

SAP HANA automatically creates self-signed server certificates during installation using an internal certificate authority. For production use, you should replace these default certificates with certificates signed by your organization's trusted Certificate Authority.

Certificate Generation Requirements

Before you can replace the default certificates, you must generate appropriate X.509 certificates using third-party certificate management tools. The certificate generation process varies depending on whether you're working with database-stored certificates (managed through the SAP HANA cockpit) or file system PSEs (stored in personal security environments).

For more information on generating X.509 certificates for PSEs, see SAP Note 1718944.

Certificate Management with SAP HANA Cockpit

Once you have obtained the appropriate certificates from your trusted Certificate Authority, you can use the SAP HANA cockpit to import and configure these certificates to replace the default self-signed certificates. The cockpit provides a complete workflow for managing certificates stored in the database.

For detailed instructions on certificate management procedures, see *Managing Client Certificates and Public Keys*.

Related Information

Certificate Management in SAP HANA
Managing Client Certificates and Public Keys
SAP Note 2175664
SAP Note 1718944

9.5 Managing SAP HANA Users

Every user who wants to work with the SAP HANA database must have a database user. As a user administrator, you create and provision the required users, as well as perform other tasks related to user administration.

Managing users in SAP HANA includes the following tasks:

- Configuring SAP HANA for the required user authentication mechanisms
- Provisioning users:
  1. Defining and creating roles
  2. Defining and creating user groups (optional)
  3. Creating users
  4. Granting roles to users
- Other user administration tasks, for example:
  - Deleting users when they leave the organization
  - Reactivating users after too many failed logon attempts
  - Deactivating users if a security violation has been detected
  - Resetting user passwords

> **Note**
>
> Users of SAP HANA SAP HANA Extended Services (SAP HANA XS) advanced applications are managed independently of the SAP HANA database. Dedicated administration tools are available for managing application users and roles. For more information, see the section on maintaining the SAP HANA XS advanced model runtime in the *SAP HANA Administration Guide*.

> **Note**
>
> If you are using an Identity Management (IDM) system for user provisioning, it is highly recommended that you create a dedicated technical user for that system that has the system privileges USER ADMIN and ROLE ADMIN and object privilege EXECUTE on the procedure GRANT_ACTIVATED_ROLE. This database user should then be used exclusively by the IDM system for its user provisioning tasks.

Related Information

User and Role Management
SAP HANA User Management
Maintaining the SAP HANA XS Advanced Model Run Time [page 1256]

9.5.1 Database Users

Every user who wants to work with the SAP HANA database must have a database user.

Database users are created with either the `CREATE USER` statement, or using the SAP HANA cockpit.

For more information about the types of database users, see the *SAP HANA Security Guide*.

Related Information

User Types
CREATE USER Statement (Access Control)
Create a Database User

9.5.1.1 The SYSTEM User

The `SYSTEM` database user is the initial user that is created during the creation of the SAP HANA database.

`SYSTEM` is the database superuser. It has irrevocable system privileges, such as the ability to create other database users, access system tables, and so on.

In the system database, the `SYSTEM` user has additional privileges for managing tenant databases, for example, creating and dropping databases, changing configuration (*.ini) files of databases, and performing database-specific data backups.

It is highly recommended that you do not use `SYSTEM` for day-to-day activities in production environments. Instead, use it to create database users with the minimum privilege set required for their duties (for example, user administration, system administration). Then deactivate `SYSTEM`. You may temporarily reactivate the `SYSTEM` user for emergency or bootstrapping tasks.

> ⓘ Note
>
> The `SYSTEM` user is not required to update the SAP HANA database system; a lesser-privileged user can be created for this purpose. However, to upgrade SAP support package stacks, SAP enhancement packages and SAP systems using the Software Update Manager (SUM) and to install, migrate, and provision SAP systems using the Software Provisioning Manager (SWPM), the `SYSTEM` user **is required** and needs to be temporarily reactivated for the duration of the upgrade, installation, migration or provisioning.

If the password of `SYSTEM` user of the system database is lost, it can be reset using the operating system user (`<sid>`adm user). The system administrator can reset the `SYSTEM` user password of a tenant database from the system database.

Related Information

Deactivate the SYSTEM User [page 545]

Create a Lesser-Privileged Database User for Update
Resetting the SYSTEM User Password [page 546]

9.5.1.2 Deactivate the SYSTEM User

As the most powerful database user, `SYSTEM` is not intended for use in production systems. Use it to create lesser privileged users for particular purposes and then deactivate it.

Prerequisites

You have the system privilege `USER ADMIN`.

Context

It is highly recommended that you do not use `SYSTEM` for day-to-day activities in production environments. Instead, use it to create database users with the minimum privilege set required for their duties (for example, user administration, system administration). Then deactivate `SYSTEM`. You may temporarily reactivate the `SYSTEM` user for emergency or bootstrapping tasks.

> ⓘ Note
>
> The `SYSTEM` user is not required to update the SAP HANA database system; a lesser-privileged user can be created for this purpose. However, to upgrade SAP support package stacks, SAP enhancement packages and SAP systems using the Software Update Manager (SUM) and to install, migrate, and provision SAP systems using the Software Provisioning Manager (SWPM), the `SYSTEM` user **is required** and needs to be temporarily reactivated for the duration of the upgrade, installation, migration or provisioning.

Procedure

Execute the following statement:

> {·} Code Syntax

```
ALTER USER SYSTEM DEACTIVATE USER NOW;
```

Results

The `SYSTEM` user is deactivated and can no longer **connect** to the SAP HANA database. However, it may appear as though `SYSTEM` is still active in the system (for example when a procedure that was created by `SYSTEM` with `DEFINER MODE` is called).

You can verify that user `SYSTEM` is in fact deactivated in the `USERS` system view. For user `SYSTEM`, check the values in the columns `USER_DEACTIVATED`, `DEACTIVATION_TIME`, and `LAST_SUCCESSFUL_CONNECT`.

> **Note**
>
> You can still use the `SYSTEM` user as an emergency user even if it has been deactivated. Any user with the system privilege `USER ADMIN` can reactivate `SYSTEM` with the statement `ALTER USER SYSTEM ACTIVATE USER NOW`. To ensure that an administrator does not do this surreptitiously, it is recommended that you create an audit policy monitoring `ALTER USER` statements. Also change the password of the `SYSTEM` user after reactivating it.

Related Information

ALTER USER Statement (Access Control)
Create a Lesser-Privileged Database User for Update
Create an Audit Policy

9.5.1.3 Resetting the SYSTEM User Password

The system database and all tenant databases each have their own SYSTEM user. The system administrator can reset the password of any SYSTEM user if it has been irretrievably lost.

Related Information

Reset the SYSTEM User Password in the System Database [page 547]
Reset the SYSTEM User Password in a Tenant Database [page 550]

9.5.1.3.1 Reset the SYSTEM User Password in the System Database

A database administrator has a number of options to reset the password for the SYSTEM user in the system database.

Prerequisites

Situation	Prerequisite
If you **can** log on as the SYSTEM user	Logon credentials for the SYSTEM user in the system database
If you **cannot** log on as the SYSTEM user	• The system privilege USER ADMIN for a different user • To reset the SYSTEM user password in emergency mode, you need the credentials of the operating system administrator (`<sid>adm`).

Context

If you **can** log on as the SYSTEM user	• Change the password using SAP HANA cockpit. Log onto the system database, and choose *User & Role Management* > *Manage Users*. • Log onto the system database, and execute the ALTER USER SQL statement: `ALTER USER SYSTEM PASSWORD <new_password>`
If you **cannot** log on as the SYSTEM user	If the password of the SYSTEM user has been irretrievably lost, you cannot log on to the database as the SYSTEM user. A user with the system privilege USER ADMIN can set a new password for the SYSTEM user. Alternatively, you can reset the password as the operating system administrator by starting the name server in emergency mode. The steps to do this are described below.

Procedure

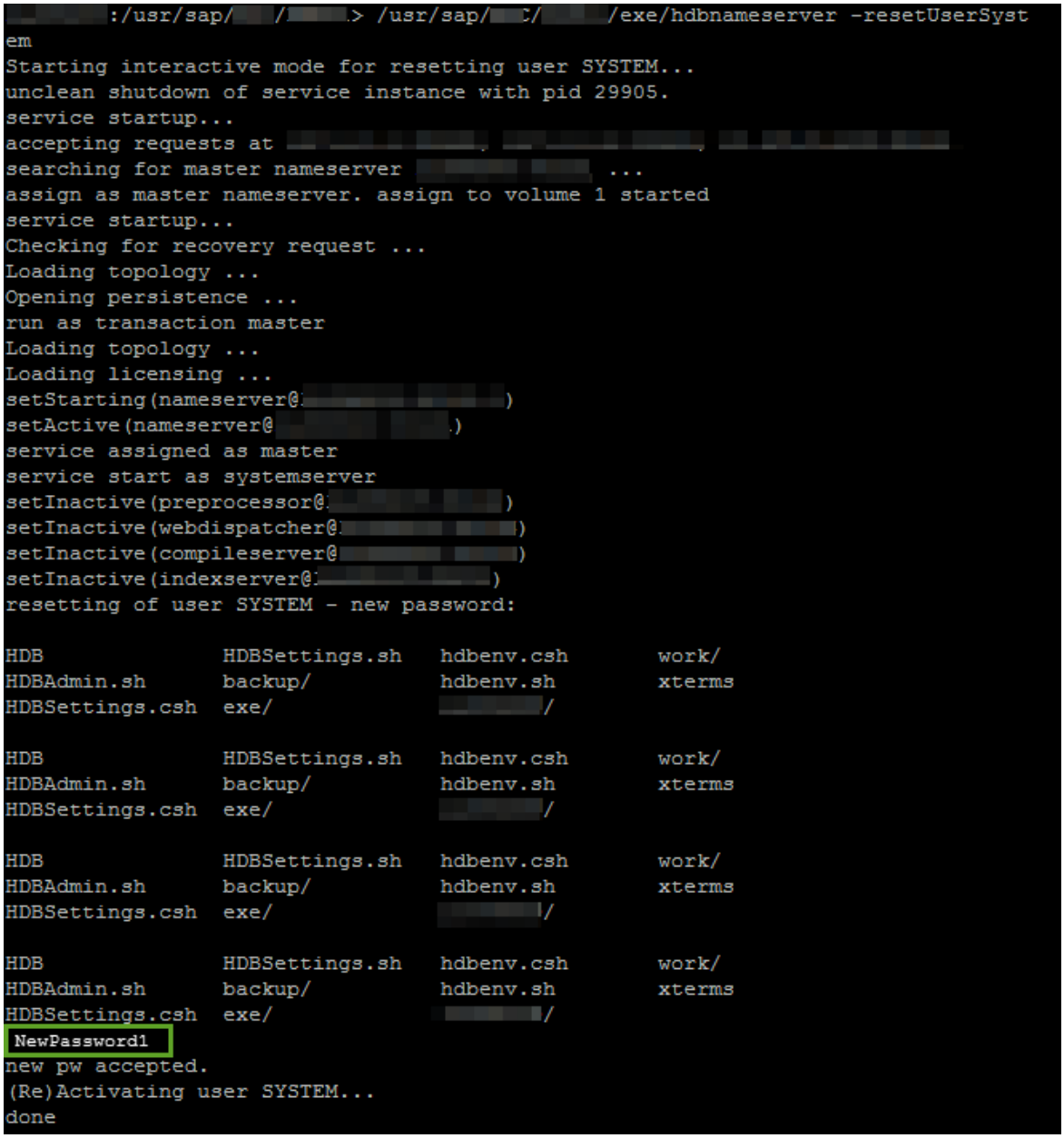
To reset the SYSTEM user password in emergency mode:

1. Log on to the server on which the name server of the system database is running as the operating system user (that is, `<sid>adm` user).
2. Open a command line interface.
3. Shut down the instance by executing the following command:

   ```
   /usr/sap/<SID>/HDB<instance>/exe/sapcontrol -nr <instance> -function StopSystem HDB
   ```

4. In a new session, start the name server of the system database by executing the following commands:
   - `/usr/sap/<SID>/HDB<instance>/hdbenv.sh`
   - `/usr/sap/<SID>/HDB<instance>/exe/hdbnameserver -resetUserSystem`

   After some start-up notifications, the prompt `resetting of user SYSTEM - new password` appears, followed by additional notifications:



Reset SYSTEM User Password (System Database)

5. After the last notification appears, enter a new password for the SYSTEM user.

   You must enter a password that complies with the password policy configured for the system.

   The password for the SYSTEM user of the system database is reset and the name server stops.

6. In a new session, start the instance by executing the following command:

   ```
   /usr/sap/<SID>/HDB<instance>/exe/sapcontrol -nr <instance> -function StartSystem
   HDB
   ```

Results

- The password of the SYSTEM user in the system database is reset.
  The next time you log on with this user, you have to change the password in accordance with the password policy of the system database.
- If you previously deactivated the SYSTEM user, it is now also reactivated. This means you will need to deactivate it again.

9.5.1.3.2 Reset the SYSTEM User Password in a Tenant Database

A database administrator has a number of options to reset the password for the SYSTEM user in a tenant database.

Prerequisites

Situation	Prerequisite
If you **can** log on as the SYSTEM user	Logon credentials for the SYSTEM user in the tenant database
If you **cannot** log on to the tenant database as the SYSTEM user	• The system privilege USER ADMIN for a different user in the tenant database • You are connected to the system database and have the system privilege DATABASE ADMIN.

Context

If you **can** log on as the SYSTEM user	• Change the password using SAP HANA cockpit. Log onto the tenant database, and choose *User & Role Management* > *Manage Users*. • Log onto the tenant database, and execute the SQL statement ALTER USER: `ALTER USER SYSTEM PASSWORD <new_password>`

If you **cannot** log on as the SYSTEM user	If the password of the SYSTEM user has been irretrievably lost, you cannot log on to the tenant database as the SYSTEM user. A user with the system privilege USER ADMIN can set a new password for the SYSTEM user from the system database. Alternatively, you can reset the password using the SQL statement ALTER DATABASE from the system database. The steps to do this are described below.

Procedure

To reset the SYSTEM user password for a tenant database:

1. Stop the tenant database.

   Execute the following statement:

   ```
   ALTER SYSTEM STOP DATABASE <database_name>
   ```

2. Create a new password for the SYSTEM user by executing the following statement:

   ```
   ALTER DATABASE <database_name> SYSTEM USER PASSWORD <new_password>
   ```

   > ⓘ Note
   >
   > The password must adhere to the password policy of the system database.

   The password for the SYSTEM user is reset and the tenant database is started.

Results

- The password of the SYSTEM user in the tenant database is reset.
  The next time you log on with this user, you have to change the password in accordance with the password policy of the tenant database.
- If the SYSTEM user was previously deactivated, locked, or expired, it is now activated again. We recommend that you deactivate it.
- If auditing is enabled, the password change is automatically logged in both the system and tenant database audit trails.

Related Information

Deactivate the SYSTEM User [page 545]
Auditing Activity in the SAP HANA Database [page 499]
Change a Database User

Reset the SYSTEM Password of a Tenant Using the Cockpit [page 109]

9.5.2 Operating System User <sid>adm

The <sid>adm user is not a database user but a user at the operating system level. Also referred to as the operating system administrator, this user has unlimited access to all local resources related to SAP systems.

In addition to the SAP HANA database user SYSTEM, the installation process also creates an external operating system user (<sid>adm, for example, sp1adm or xyzadm).

This operating system user, also referred to as the operating system administrator, simply exists to provide an operating system context. From the operating system perspective, the operating system administrator is the user that owns all SAP HANA files and all related operating system processes. The operating system user's credentials are required, for example, to start or stop database processes or to execute a recovery.

The operating system user is not an SAP HANA database user.

9.5.3 Configuring User Authentication and Single-Sign On

SAP HANA supports several authentication mechanisms, several of which can be used for the integration of SAP HANA into single sign-on environments (SSO). Depending on which mechanisms you are implementing, you must configure SAP HANA accordingly.

Users accessing SAP HANA can be authenticated using:

- User name and password
- Security assertion markup language (SAML)
- X.509 client certificates
- JSON Web Tokens (JWT)
- Kerberos, SPNEGO
- Logon and assertion tickets
- LDAP directory server

This section provides an overview of the steps necessary in SAP HANA to configure the various mechanisms for JDBC/ODBC access. For HTTP access via SAP HANA Extended Application Services, advanced and classic model, see the section on application run-time services.

For more information about the authentication mechanisms themselves, see the *SAP HANA Security Guide*.

Related Information

User Authentication Mechanisms
Maintaining Single Sign-On for XS Advanced Applications [page 1509]
Maintaining Single Sign-On for XS Classic Applications [page 1191]
Configuring LDAP Integration [page 569]

9.5.3.1 Configure Password Policies

The passwords of database users are subject to certain rules, which are defined in password policies. You can change the default password policy of the database and maintain user group-specific password policies in line with your organization's security requirements.

Context

The password policy of the database is defined by parameters in the `password policy` section of the `indexserver.ini` configuration file for tenant databases and the `nameserver.ini` configuration file for the system database. The database password policy is valid for all database users unless the user is in a user group with its own dedicated password policy

> → Tip
>
> To determine which password policy a user is currently subject to, query the system view M_EFFECTIVE_PASSWORD_POLICY.

In addition to configuring the password policy parameters, you can also add words or partial words to the password exclude list. The password exclude list is implemented with the database table `_SYS_PASSWORD_BLACKLIST` in the schema `_SYS_SECURITY`. This table is empty when the database is first created.

Procedure

Configure password policies as follows:

- To change the database password policy, use the *Authentication* app of the SAP HANA cockpit.
- To maintain a password policy for a user group, use the *User Group Management* app of the SAP HANA cockpit.
- To maintain the password exclude list, use the *Authentication* app of the SAP HANA cockpit.

Related Information

Password Policy
User Groups
Password Policy Configuration Options
Configure a Password Policy for a User Group
Configure the Database Password Policy and Password Exclude List
M_EFFECTIVE_PASSWORD_POLICY System View

9.5.3.2 Configure SAML Authentication for ODBC/JDBC Access

Configure SAP HANA to use SAML bearer assertions to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients.

Prerequisites

- You have the system privilege CREATE SAML PROVIDER.
- If you are using certificate collections and certificates stored directly in the database (recommended), you have all the necessary privileges. See *SQL Statements and Authorization for In-Database Certificate Management (Reference)* in the *SAP HANA Security Guide*.
- You have the system privilege USER ADMIN or for users in user groups configured for exclusive administration, the object privilege USERGROUP OPERATOR on the relevant user group(s).

Context

Configuring SAML authentication involves a number of logical steps. These are outlined below. For more information on how to perform the individual steps, see the SAP HANA cockpit documentation or the *SAP HANA SQL Reference*.

Procedure

1. Create the identity providers in SAP HANA that will be used to validate incoming SAML assertions.

   You can create SAML identity providers in SAP HANA using the SAP HANA cockpit or the CREATE SAML PROVIDER statement.
2. Configure database users for SAML authentication and map them to their external identities.

   You can either map database users explicitly to their external identities (SAP HANA-based user mappings) or allow the identity provider to map its users to database users (identity provider-based user mappings). You do this in the user definition using either the SAP HANA cockpit or the CREATE | ALTER USER statement.
3. Create and configure the trust store(s) used to validate incoming tokens against certificates signed by a trusted Certification Authority (CA).

   We recommend creating a certificate collection with the purpose `SAML` that contains the required certificates directly in the database. If necessary, you can create separate collections for different identity providers.

   You can import certificates, and create and configure certificate collections using the SAP HANA cockpit or the relevant SQL statements (CREATE CERTIFICATE, CREATE PSE, ALTER PSE and SET PSE).

It is also possible to use a trust store located on the file system.

> ⚠ Caution
>
> We recommend creating certificate collections for individual purposes in the database directly, rather than using trust stores (PSE) in the file system. By default, the same PSE in the file system is shared by all databases for all external communication channels (including HTTP) and SAML authentication. Different PSEs must be explicitly configured for tenant databases.

Related Information

Single Sign-On Using SAML 2.0
Add a SAML Identity Provider
CREATE SAML PROVIDER Statement (Access Control)
Create a Database User
CREATE USER Statement (Access Control)
SQL Statements and Authorization for In-Database Certificate Management (Reference)
Managing Client Certificates and Public Keys
CREATE CERTIFICATE Statement (System Management)
CREATE PSE Statement (System Management)
ALTER PSE Statement (System Management)
SET PSE Statement (System Management)

9.5.3.3 Configure JWT Authentication for ODBC/JDBC Access

Configure SAP HANA to use JSON Web Tokens (JWT) to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients.

Prerequisites

- You have the system privilege CREATE JWT PROVIDER.
- You have all the privileges necessary for managing certificates and certificate collections. See *SQL Statements and Authorization for In-Database Certificate Management (Reference)* in the *SAP HANA Security Guide*.
- You have the system privilege USER ADMIN or for users in user groups configured for exclusive administration, the object privilege USERGROUP OPERATOR on the relevant user group(s).

Context

Configuring JWT authentication involves a number of logical steps. These are outlined below. For more information on how to perform the individual steps, see the SAP HANA cockpit documentation or the *SAP HANA SQL Reference*.

Procedure

1. Create the identity providers in SAP HANA that will be used to validate incoming tokens.

   You can create JWT identity providers in SAP HANA using the SAP HANA cockpit or the CREATE JWT PROVIDER statement.
2. Configure database users for JWT authentication and map them to their external identities.

   You can either map database users explicitly to their external identities (SAP HANA-based user mappings) or allow the identity provider to map its users to database users (identity provider-based user mappings). You do this in the user definition using either the SAP HANA cockpit or the CREATE | ALTER USER statement.
3. Create and configure the trust store(s) used to validate incoming tokens against certificates or public keys signed by a trusted Certification Authority (CA).

   To do this, you create a certificate collection with the purpose `JWT` that contains the required certificates or public keys. If necessary, you can create separate collections for different identity providers.

   You can import certificates and public keys, and create and configure certificate collections using the SAP HANA cockpit or the relevant SQL statements (CREATE CERTIFICATE, CREATE PUBLIC KEY, CREATE PSE, ALTER PSE and SET PSE).

Related Information

Single Sign-On Using JSON Web Tokens
Add a JWT Identity Provider
CREATE JWT PROVIDER Statement (Access Control)
Create a Database User
CREATE USER Statement (Access Control)
SQL Statements and Authorization for In-Database Certificate Management (Reference)
Managing Client Certificates and Public Keys
CREATE CERTIFICATE Statement (System Management)
CREATE PUBLIC KEY Statement (System Management)
CREATE PSE Statement (System Management)
ALTER PSE Statement (System Management)
SET PSE Statement (System Management)

9.5.3.4 Configure X.509 Certificate Authentication for ODBC/JDBC Access

Configure SAP HANA to use X.509 certificates to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients.

Prerequisites

- You have the system privilege CREATE X509 PROVIDER.
- You have all the privileges necessary for managing certificates and certificate collections. See *SQL Statements and Authorization for In-Database Certificate Management (Reference)* in the *SAP HANA Security Guide*.
- You have the system privilege USER ADMIN or for users in user groups configured for exclusive administration, the object privilege USERGROUP OPERATOR on the relevant user group(s).

Context

Configuring X.509 certificate-based authentication involves a number of logical steps. These are outlined below. For more information on how to perform the individual steps, see the SAP HANA cockpit documentation or the *SAP HANA SQL Reference*.

Procedure

1. Create the identity providers in SAP HANA that will be used to validate incoming certificates.

   You can create X.509 identity providers in SAP HANA using the SAP HANA cockpit or the CREATE X509 PROVIDER statement.
2. Configure database users for X.509 certificate-based authentication and map them to their external identities.

   You can either map database users explicitly to their external identities (SAP HANA-based user mappings) or allow the identity provider to map its users to database users (identity provider-based user mappings). In the latter case, the identity provider maps its users to database users using matching rules that you define in the identity provider (step 1).

   You configure user mapping using either the SAP HANA cockpit or the CREATE | ALTER USER statement.
3. Create and configure the trust store used to validate incoming certificates against certificates signed by a trusted Certification Authority (CA).

   To do this, you create a certificate collection with the purpose `X509` that contains the required certificates. If necessary, you can create separate collections for different identity providers.

   You can import certificates, and create and configure certificate collections using the SAP HANA cockpit or the relevant SQL statements (CREATE CERTIFICATE, CREATE PSE, ALTER PSE and SET PSE).

4. Configure the client connection if necessary.

   If the connection between SAP HANA and the client is TLS/SSL secured, by default, the client's TLS certificate is used for authentication if a user is enabled for X.509 authentication and no other credentials have been specified. You can specify a different X.509 certificate for authentication as part of the connection string using the `authenticationX509` connection parameter. You can also explicitly deactivate X.509 authentication for a client connection by using the `authenticationMethods` parameter in the connection string to specify the authentication mechanisms that you do want to use. For more information, see the *SAP HANA Client Interface Programming Reference*.

Related Information

X.509 Certificate-Based User Authentication
Add an X.509 Identity Provider
CREATE X509 PROVIDER (Access Control)
Create a Database User
CREATE USER Statement (Access Control)
SQL Statements and Authorization for In-Database Certificate Management (Reference)
Managing Client Certificates and Public Keys
CREATE CERTIFICATE Statement (System Management)
CREATE PSE Statement (System Management)
ALTER PSE Statement (System Management)
SET PSE Statement (System Management)
JDBC Connection Properties
ODBC Connection Properties

9.5.3.5 Configure Kerberos for SAP HANA Database Hosts

If you are implementing Kerberos-based user authentication, you must configure Kerberos on the authentication server.

Prerequisites

To allow users to log on to the SAP HANA database using Kerberos authentication, you have installed MIT Kerberos client libraries on the host(s) of the SAP HANA database.

Context

SAP HANA supports Kerberos version 5 for single sign-on based on Active Directory (Microsoft Windows Server) or Kerberos authentication servers. For HTTP access via SAP HANA Extended Services (SAP HANA

XS), advanced or classic model, Kerberos authentication is enabled with Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO).

Once Kerberos client libraries have been installed, you must configure Kerberos on the authentication server by performing the following logical steps:

1. Register service principal names (SPN) for each host in the SAP HANA system using the following syntax: `<service>/<host domain name>@<Kerberos realm name>`, where
   - `<service>` is either **`hdb`** (for Kerberos via ODBC and JDBC) or **`HTTP`** (for SPNEGO via HTTP/SAP HANA XS)
   - `<host domain name>` is the fully qualified domain name of the host
     If the service is HTTP, you must register one SPN for each alias of the host name as well.
   - `<Kerberos realm name>` (Kerberos terminology) is identical to domain name in Active Directory terminology

   This results in the generation of a service key table (keytab) for each host. This keytab contains the encrypted key for the host in question.
2. Export the keytab(s) to files.
3. Import each keytab file into the Kerberos installation on the respective host.

Procedure

The concrete steps to be performed on the authentication server depend on whether you are using Kerberos or Active Directory as follows:

1. Register the SPNs.

   > ⓘ Note
   >
   > In Active Directory, before a SPN can be registered, you must create a plain user account that acts as the server principal on the domain controller. Afterward, you must map the SPN to the user account using a separate command.

2. Export the keytab(s) to files using a command line tool shipped with the authentication server.

   This is applicable for both Kerberos and Active Directory.
3. Import the keytab files.
   The files are transported to the file system path on the SAP HANA database hosts in line with how the Kerberos client is configured.

Results

You can now map the users stored in the Kerberos Key Distribution Center (KDC) to database users in SAP HANA database. You can do this when you create database users. Alternatively, if database users already exist, you can change their authentication details.

> → Remember
>
> A per-database configuration is not possible – databases users in all databases must be mapped to users in the same KDC.

For more information about how to set up SSO with SAP HANA using Kerberos and Microsoft Active Directory, see SAP Note 1837331.

Related Information

Single Sign-On Using Kerberos
Maintaining the SAP HANA XS Classic Model Run Time [page 1134]
Create a Database User
SAP Note 1837331

9.5.3.6 Configure Authentication Using SAP Logon Tickets and Assertions

Configure SAP HANA to use SAP logon or assertion tickets to authenticate users accessing SAP HANA directly from ODBC/JDBC database clients.

Prerequisites

- If you are using certificate collections and certificates stored directly in the database (recommended), you have all the necessary privileges. See *SQL Statements and Authorization for In-Database Certificate Management (Reference)* in the *SAP HANA Security Guide*.
- If you are using a trust store located in the file system, you have the system privilege INIFILE ADMIN.
- You have the system privilege USER ADMIN or for users in user groups configured for exclusive administration, the object privilege USERGROUP OPERATOR on the relevant user group(s).

Context

SAP HANA validates incoming logon/assertion tickets against certificates signed by a trusted Certification Authority (CA) stored in a dedicated trust store. This trust store must contain all root certificate(s) used to validate logon/assertion tickets. We recommend creating a certificate collection with the purpose **SAP LOGON** and the required certificates directly in the database.

It is also possible to use a trust store located in the file system. The default location of the trust store in the file system depends on the cryptographic library configured for SSL:

- `$SECUDIR/saplogon.pse` (CommonCryptoLib)

  > **Note**
  >
  > The saplogon.pse trust store is available automatically.

- `$HOME/.ssl/saplogon.pem` (OpenSSL)

  > **Note**
  >
  > **Deprecated:** OpenSSL is deprecated. You must migrate to CommonCryptoLib. For more information, see SAP Note 2093286.

If necessary, you can change the location of this trust store in the `indexserver.ini` system properties file.

> **Caution**
>
> By default, the same trust store in the file system is shared by all databases. Different PSEs must be explicitly configured for tenant databases.

Procedure

1. Configure the trust store:

Option	Description
In-database certificate collection	Create a certificate collection with the purpose `SAP LOGON` that contains the required certificates. You can import certificates, and create and configure certificate collections using the SAP HANA cockpit or the relevant SQL statements (CREATE CERTIFICATE, CREATE PSE, ALTER PSE and SET PSE).
File-system based	1. In the configuration file `indexserver.ini` file, change the value of the `[authentication] saplogontickettruststore` parameter. 2. Restart the system. You can modify configuration files in the SAP HANA cockpit.

2. Configure users for authentication using logon/assertion tickets.

   You do this in the user definition using either the SAP HANA cockpit or the CREATE | ALTER USER statement.

Related Information

Single Sign-On Using SAP Logon and Assertion Tickets
Managing Client Certificates and Public Keys
SQL Statements and Authorization for In-Database Certificate Management (Reference)
CREATE CERTIFICATE Statement (System Management)
CREATE PSE Statement (System Management)
ALTER PSE Statement (System Management)
SET PSE Statement (System Management)
Modify a System Property in SAP HANA Cockpit
CREATE USER Statement (Access Control)
Create a Database User

SAP Note 2093286

9.5.4 User Authorization

After they have been successful logged on, the user's authorization to perform the requested operations on the requested objects is verified.

To perform operations in the SAP HANA database, a database user must have the necessary privileges. Users must have both the privilege(s) to perform the operation and to access the resources (such as schemas and tables) to which the operation applies. Privileges can be granted to database users either directly, or indirectly through roles that they have been granted. In this case, the privileges are inherited. Roles are the standard mechanism of granting privileges to users.

For more information about how to grant roles and privileges to users, see the section *Managing User Authorization* in the SAP HANA cockpit documentation. For more information about the authorization concept in SAP HANA, see the *SAP HANA Security Guide*.

> ⓘ Note
>
> For some administration tasks, such as start-up, the credentials of the SAP operating system user (`<sid>`adm) are also required.

Related Information

SAP HANA Authorization
Managing User Authorization
Operating System User <sid>adm [page 552]
Logon Checks

9.5.4.1 System Views for Verifying Users' Authorization

You can query several system views to get detailed information about exactly which privileges and roles users have and how they come to have them. This can help you to understand why a user is authorized to perform particular actions, access particular data, or not.

> → Remember
>
> You must have the system privilege CATALOG READ to query the following views.

System View	Query	Result
ACCESSIBLE_VIEWS	SELECT * from "PUBLIC"."ACCESSIBLE_VIEWS" where USER_NAME = '<user_name>';	All views that the user is authorized to access are returned.
EFFECTIVE_APPLICATION_PRIVILEGES	select * from "SYS"."EFFECTIVE_APPLICATION_PRIVILEGES" where USER_NAME='<user_name>';	All application privileges granted to the specified user both directly and indirectly through roles are are returned separately.
EFFECTIVE_MASK_EXPRESSIONS	SELECT * FROM EFFECTIVE_MASK_EXPRESSIONS where ROOT_SCHEMA_NAME = '<schema_name>' and ROOT_OBJECT_NAME = '<object_name>' and ROOT_COLUMN_NAME = '<column_name>' and USER_NAME ='<user_name>';	All masked columns that the specified user can see in the specified view and the corresponding mask expressions

System View	Query	Result
EFFECTIVE_PRIVILEGE_GRANTEES	Object privilege: `SELECT * FROM EFFECTIVE_PRIVILEGE_GRANTEES WHERE OBJECT_TYPE = '<object_type>' AND SCHEMA_NAME = '<schema_name>' AND OBJECT_NAME = '<object_name>' AND PRIVILEGE = '<privilege>';` System privilege: `SELECT * FROM EFFECTIVE_PRIVILEGE_GRANTEES WHERE OBJECT_TYPE = 'SYSTEMPRIVILEGE' AND PRIVILEGE = '<privilege>';` XML-based analytic privilege: `SELECT * FROM EFFECTIVE_PRIVILEGE_GRANTEES WHERE OBJECT_TYPE = 'ANALYTICPRIVILEGE' AND SCHEMA_NAME = '<schema_name>' AND OBJECT_NAME = '<privilege_object_name>' AND PRIVILEGE = 'EXECUTE';` SQL-based analytic privilege: `SELECT * FROM EFFECTIVE_PRIVILEGE_GRANTEES WHERE OBJECT_TYPE = 'SQLANALYTICPRIVILEGE' AND SCHEMA_NAME = '<schema_name>' AND OBJECT_NAME = '<privilege_object_name>' AND PRIVILEGE = 'EXECUTE';` **Note** For analytic privileges, schema name is optional.	All users or roles who have the specified privilege are returned.
EFFECTIVE_PRIVILEGES	`SELECT * FROM "PUBLIC"."EFFECTIVE_PRIVILEGES" where USER_NAME = '<user_name>';`	All privileges granted to the specified user both directly and indirectly through roles are returned separately.
EFFECTIVE_ROLE_GRANTEES	`SELECT * FROM EFFECTIVE_ROLE_GRANTEES WHERE ROLE_NAME = '<role_name>';`	All users or roles who have the specified role are returned.

System View	Query	Result
EFFECTIVE_ROLES	SELECT * FROM "PUBLIC"."EFFECTIVE_ROLES" where USER_NAME = '<user_name>' AND ROLE_SCHEMA_NAME = '<schema_name of role>'; **Note** Schema name is optional.	All roles granted to the specified user both directly and indirectly through other roles are returned separately.
EFFECTIVE_STRUCTURED_PRIVILEGES	SELECT * from "PUBLIC"."EFFECTIVE_STRUCTURED_PRIVILEGES" where ROOT_SCHEMA_NAME = '<schema>' AND ROOT_OBJECT_NAME = '<object_name>' AND USER_NAME = '<user_name>'	The analytic privileges that are applicable to the specified view are returned, including dynamic filter conditions if relevant. It is also indicated whether or not the specified user is authorized to access the view.
GRANTED_PRIVILEGES	SELECT * FROM "PUBLIC"."GRANTED_PRIVILEGES" where GRANTEE = '<user_name>';	Privileges granted directly to the specified user (or role) are returned. Privileges contained within granted roles are not shown. **Note** It is possible to query the privileges directly granted to a role by replacing `where GRANTEE = '<USER>'` with `where GRANTEE = '<ROLE>'`
GRANTED_ROLES	SELECT * FROM "PUBLIC"."GRANTED_ROLES" where GRANTEE = '<user/role_name>';	All roles granted directly to the specified user (or role) are returned. Roles contained within granted roles are not shown. **Note** It is possible to query the roles directly granted to a role by replacing `where GRANTEE = '<USER>'` with `where GRANTEE = '<ROLE>'`

Related Information

ACCESSIBLE_VIEWS System View
EFFECTIVE_APPLICATION_PRIVILEGES System View
EFFECTIVE_MASK_EXPRESSIONS System View
EFFECTIVE_PRIVILEGE_GRANTEES System View
EFFECTIVE_PRIVILEGES System View
EFFECTIVE_ROLE_GRANTEES System View
EFFECTIVE_ROLES System View

EFFECTIVE_STRUCTURED_PRIVILEGES System View
GRANTED_PRIVILEGES System View
GRANTED_ROLES System View

9.5.4.2 Resolve Insufficient Privilege Errors

Use the built-in procedure `GET_INSUFFICIENT_PRIVILEGE_ERROR_DETAILS` to find out which privilege a user is missing when he or she gets an "insufficient privilege" error.

Prerequisites

- You have the EXECUTE object privilege on `GET_INSUFFICIENT_PRIVILEGE_ERROR_DETAILS`.
- You have the GUID of the error received by the end user experiencing the authorization issue.

  Example

  ```
  insufficient privilege: Detailed info for this error can be found with guid
  '3DFFF7D0CA291F4CA69B327067947BEE'
  ```

Context

When a user is not authorized to perform an operation and receives an "insufficient privilege" error, it is often difficult to know which privilege or privileges the user is in fact missing. Many "insufficient privilege" errors therefore also return a GUID that allows you as an administrator to identify the missing privilege. Then you can decide whether or not to grant the privilege to the user.

> Note
>
> Not all "insufficient privilege" errors can be resolved by granting the user a missing privilege, for example, some operations can only be performed by the SYS user, or by a user in the system database. In these cases, the error message provides the explanation.

If the user is missing several privileges, only the first missing privilege determined by the authorization check is returned. You can find out what other privileges are missing by successively running `GET_INSUFFICIENT_PRIVILEGE_ERROR_DETAILS` with the GUID returned with each subsequent error message.

The details of every "insufficient privilege" error due missing user privileges are stored in an internal table for 144 hours by default. The number of entries in this table is limited to 10,000. This configuration is controlled by the following parameters in the `authorization` section of the `global.ini` file:

- `insufficient_privilege_error_details_retain_duration`
- `insufficient_privilege_error_details_retain_records`

> **Note**
>
> It is possible to disable this feature by setting the parameter `enable_insufficient_privilege_error_details_procedure` to false. However, then it is only possible to troubleshoot authorization errors using the database trace, which may not be feasible in all scenarios.

Procedure

Execute the procedure `GET_INSUFFICIENT_PRIVILEGE_ERROR_DETAILS`:

```
call SYS.GET_INSUFFICIENT_PRIVILEGE_ERROR_DETAILS ('<GUID in "insufficient
privilege" error>', ?)
```

Results

The result provides you with information about the first authorization check performed for the particular user connection.

Next Steps

If you grant the end user the identified missing privilege and he or she gets another "insufficient privilege", then the user is missing another privilege. Call the procedure again with the GUID of the next error message to troubleshoot further.

Related Information

GET_INSUFFICIENT_PRIVILEGE_ERROR_DETAILS
Database Trace (Basic, User-Specific, and End-to-End) [page 464]
Display Information about an "Insufficient Privilege" Error

9.5.4.3 Restrict Use of the CLIENT User Parameter

Allow only authorized technical users to overwrite the value of the `CLIENT` parameter for a database connection or the value of the `$$client$$` parameter in an SQL query.

Context

The `CLIENT` user parameter can be used to authorize named users in SAP HANA database. Only a user with the USER ADMIN system privilege can change the value of the `CLIENT` parameter already assigned to other users. However, at runtime, any user can assign an arbitrary value to the `CLIENT` parameter either by setting the corresponding session variable or passing the parameter via placeholder in a query.

While this is the desired behavior for technical users that work with multiple clients such as SAP Business Warehouse, S/4 HANA, or SAP Business Suite, it is problematic in named user scenarios if the `CLIENT` parameter is used to authorize access to data and not only to perform data filtering.

Procedure

1. Grant the system privilege `CLIENT PARAMETER ADMIN` to database users or roles who are permitted to access to the CLIENT user parameter (for example, technical users).

   Sample Code

   ```
   GRANT CLIENT PARAMETER ADMIN TO <user or role>;
   ```

2. In the `global.ini` configuration file, see the value of the `[authorization] secure_client_parameter` to **`true`**.

Results

Only users with the system privilege CLIENT PARAMETER ADMIN can overwrite the value of the `CLIENT` parameter for a database connection or the value of the `$$client$$` parameter in an SQL query.

Related Information

SAP Note 2582162

9.5.5 Configuring LDAP Integration

The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing directory services. If you use an LDAP-compliant directory server to manage users and their access to resources, you can leverage LDAP-based authentication to access SAP HANA and LDAP group membership to authorize users.

Related Information

LDAP User Authentication
LDAP Group Authorization for Existing Users
Secure Communication Between SAP HANA and an LDAP Directory Server

9.5.5.1 Configure LDAP Authentication and Authorization

Set up a connection to an LDAP server by creating an LDAP provider in SAP HANA. Depending on your requirements, you can use the LDAP server to authenticate or authorize, or to authenticate and authorize users. For LDAP-authenticated users, you can also enable the automatic creation of users in SAP HANA.

Prerequisites

- An LDAP v3 compliant server
- Communication between SAP HANA and the LDAP server is secure.

  > ⚠ Caution
  >
  > If the LDAP server is being used for user authentication, communication between SAP HANA and the LDAP server must be secured to protect the transmission of user passwords.

  > ⓘ Note
  >
  > To protect data transmitted between the client and SAP HANA, as well as to ensure that the client is connecting to the expected SAP HANA (server authentication), you must secure communication between clients and SAP HANA.

- To create an LDAP provider in SAP HANA, you have the system privilege `LDAP ADMIN`.
- To configure SAP HANA users, system privilege `USER ADMIN` or users in user groups configured for exclusive administration, the object privilege `USERGROUP OPERATOR` on the relevant user group(s).
- To map LDAP groups to SAP HANA roles (authorization only), your have the system privilege `ROLE ADMIN`.

Procedure

1. Create and configure an LDAP provider in the SAP HANA database.

   You do this using the `CREATE LDAP PROVIDER` statement. For more information, see the section *LDAP Provider Configuration (Reference)* in this guide, as well as the documentation for `CREATE LDAP PROVIDER` in the *SAP HANA SQL Reference*.

   {·} Sample Code

   The following example creates an LDAP provider that can be used to authenticate users via a secure connection (LDAP with STARTTLS). The provider is enabled and set as the default LDAP provider.

   ```
   CREATE LDAP PROVIDER my_ldap_provider
   CREDENTIAL TYPE 'PASSWORD' USING
   'user=cn=LookupAccount,cn=Users,dc=largebank,dc=como;password=<password>'
   USER LOOKUP URL 'ldap://myhostname:389/cn=Users,dc=largebank,dc=com??sub?
   (&(objectClass=user)(sAMAccountName=*))'
   ATTRIBUTE DN 'distinguishedName'
   SSL ON
   DEFAULT ON
   ENABLE PROVIDER;
   ```

   {·} Sample Code

   The following example creates an LDAP provider that can be used for authenticating and authorizing users via a secure connection (ldaps). The provider is enabled, permitted to create users in SAP HANA, and set as the default LDAP provider.

   ```
   CREATE LDAP PROVIDER my_ldap_provider
   CREDENTIAL TYPE 'PASSWORD' USING
   'user=cn=LookupAccount,cn=Users,dc=largebank,dc=como;password=hUWe8ZTiQyG'
   USER LOOKUP URL 'ldaps://myhostname:389/cn=Users,dc=largebank,dc=com??sub?
   (&(objectClass=user)(sAMAccountName=*))'
   ATTRIBUTE DN 'distinguishedName'
   ATTRIBUTE MEMBER_OF 'memberOf'
   SSL OFF
   DEFAULT ON
   ENABLE PROVIDER
   ENABLE USER CREATION FOR LDAP;
   ```

2. For **LDAP authentication**, configure the SAP HANA users to be authenticated using their LDAP password.

   ⓘ Note

   This step is only necessary if you have **not** enabled the LDAP provider for automatic user creation

   You can configure users for LDAP authentication using either the SAP HANA cockpit or the `CREATE USER` or `ALTER USER` statements.

   {·} Sample Code

   ```
   CREATE USER julie WITH IDENTITY FOR LDAP PROVIDER;
   ```

   ⓘ Note

   Users configured for LDAP authentication cannot simultaneously be configured for local password authentication. Local password authentication must be disabled before an existing user can be

configured for LDAP authentication, for example, using the statement `ALTER USER <user_name> DISABLE PASSWORD`.

3. For **authorization based on LDAP group membership**, do the following:
   a. Map LDAP groups to roles in the SAP HANA database.

      This allows SAP HANA to determine which roles to assign to users based on their membership in one or more LDAP groups, either directly or indirectly through nested groups. Users' access to requested resources is then determined by the privileges defined in the SAP HANA roles.

      You map LDAP groups to roles in the role definition in SAP HANA using either the SAP HANA cockpit or the `CREATE ROLE` or `ALTER ROLE` statements. You must specify the unique distinguished name (DN) of an LDAP group.

      Sample Code

      ```
      CREATE ROLE Securities_DBA
      LDAP GROUP
      “cn=Securities_DBA,OU=Application,OU=Groups,ou=DatabaseAdmins,cn=Users,d
      c=largebank,dc=com’;
      ```

      Note

      It is not possible to include an LDAP group in the definition of design-time roles.

   b. Configure SAP HANA users for LDAP group authorization, if necessary.

      Note

      If you have configured LDAP authentication, this step is only necessary if you have not enabled the LDAP provider for automatic user creation

      You can configure users for LDAP group authorization by specifying `LDAP` as the authorization mode using either the SAP HANA cockpit or the CREATE USER and ALTER USER statements.

      Sample Code

      ```
      CREATE USER USER1 PASSWORD <password> AUTHORIZATION LDAP;
      ```

      Note

      If you change the authorization mode of an existing user, any roles and privileges already granted to the user are revoked. For more information, see the *SAP HANA Security Guide*.

      Tip

      To see which authorization mode is configured for a user, display the user in the SAP HANA cockpit or refer to the AUTHORIZATION_MODE column of the USERS system view.

4. Verify the configuration of the LDAP provider using the `VALIDATE LDAP PROVIDER` statement.

   Example

   Example 1:

This example verifies whether the specified SAP HANA user can be successfully authenticated using their LDAP password.

Sample Code

```
VALIDATE LDAP PROVIDER my_ldap_provider CHECK USER testuser1 PASSWORD
'<LDAP_password>';
```

Example 2:

This example verifies that based on the current LDAP configuration and role-to-group mappings in SAP HANA, the specified SAP HANA user will be granted SAP HANA roles:

Sample Code

```
VALIDATE LDAP PROVIDER my_ldap_provider CHECK USER 'john';
```

Related Information

LDAP User Authentication
LDAP Provider Configuration (Reference) [page 575]
Secure Communication Between SAP HANA and an LDAP Directory Server
Secure Communication Between SAP HANA and JDBC/ODBC Clients
CREATE LDAP PROVIDER Statement (Access Control)
ALTER LDAP PROVIDER Statement (Access Control)
VALIDATE LDAP PROVIDER Statement (Access Control)

9.5.5.2 Configure SAML or JWT Authentication with LDAP-Based User Provisioning

Configure SAML- or JWT-based user authentication with automatic user provisioning in SAP HANA based on LDAP group membership.

Prerequisites

To configure automatic user provisioning based on LDAP group membership for SAML- or JWT-authenticated users, you must set up LDAP group authorization. To do this you need:

- An LDAP v3 compliant server
- System privilege `LDAP ADMIN` (to create the LDAP provider)
- System privileges `CERTFICATE ADMIN` and `TRUST ADMIN` (to configure secure communication)

- System privilege `ROLE ADMIN` or if roles are organized using role groups, object privilege `OPERATOR` on the relevant role groups (to map LDAP groups to roles)

To configure SAML-/JWT-based authentication, you need:

- System privilege `CREATE SAML PROVIDER` or `CREATE JWT PROVIDER` (to create the identity provider)
- Object privilege `OPERATOR` on the user group in which automatically created users will be created (to enable automatic user creation)
- System privileges `CERTFICATE ADMIN` and `TRUST ADMIN`, and object privilege `ALTER` on the identity provider (to configure trust validation)

Procedure

1. Create and configure an LDAP provider.

   You do this using the `CREATE LDAP PROVIDER` statement. For more information, see the section *LDAP Provider Configuration (Reference)* in this guide, as well as the documentation for `CREATE LDAP PROVIDER` in the *SAP HANA Cloud, SAP HANA Database SQL Reference*.

   {·} Sample Code

   The following example creates an LDAP provider that can be used to provision users via a secure connection (LDAP with STARTTLS). The provider is enabled and set as the default LDAP provider.

   ```
   CREATE LDAP PROVIDER my_ldap_provider
   CREDENTIAL TYPE 'PASSWORD' USING
   'user=cn=LookupAccount,cn=Users,dc=largebank,dc=como;password=<password>'
   USER LOOKUP URL 'ldap://myhostname:389/cn=Users,dc=largebank,dc=com??sub?
   (&(objectClass=user)(sAMAccountName=*))'
   ATTRIBUTE DN 'distinguishedName'
   SSL ON
   DEFAULT ON
   ENABLE PROVIDER;
   ```

   {·} Sample Code

   The following example creates an LDAP provider that can be used to provision users via a secure connection (ldaps). The provider is enabled and set as the default LDAP provider.

   ```
   CREATE LDAP PROVIDER my_ldap_provider
   CREDENTIAL TYPE 'PASSWORD' USING
   'user=cn=LookupAccount,cn=Users,dc=largebank,dc=como;password=hUWe8ZTiQyG'
   USER LOOKUP URL 'ldaps://myhostname:389/cn=Users,dc=largebank,dc=com??sub?
   (&(objectClass=user)(sAMAccountName=*))'
   ATTRIBUTE DN 'distinguishedName'
   ATTRIBUTE MEMBER_OF 'memberOf'
   SSL OFF
   DEFAULT ON
   ENABLE PROVIDER;
   ```

2. Import the certificate of the Certificate Authority (CA) that signed the certificate used by the LDAP server and create a certificate collection with purpose `LDAP`.

   You can do this in the SAP HANA cockpit or using SQL as shown in the following example.

Sample Code

```
CREATE CERTIFICATE ldap_cert FROM
'-----BEGIN CERTIFICATE-----
<certificate content>
-----END CERTIFICATE-----';
CREATE PSE ldap_pse;
ALTER PSE ldap_pse ADD CERTIFICATE ldap_cert;
SET PSE ldap_pse PURPOSE LDAP;
```

3. Create a SAML or JWT provider with automatic user creation enabled.

   You can do this in the SAP HANA cockpit or using SQL as shown in the following example.

Sample Code

```
CREATE JWT PROVIDER jwt_prov
WITH ISSUER 'http://test.localhost:8080/uaa/oauth/token'
CLAIM 'user_name' AS EXTERNAL IDENTITY
ENABLE USER CREATION  USERGROUP jwt_users  LDAP AUTHORIZATION;
```

4. Import the public certificate required for user authentication and create a certificate collection with purpose `SAML` or `JWT` for the identity provider.

   You can do this in the SAP HANA cockpit or using SQL as shown in the following example.

Sample Code

```
CREATE CERTIFICATE jwt_cert FROM
'-----BEGIN CERTIFICATE-----
<certificate content>
-----END CERTIFICATE-----';
CREATE PSE jwt_pse;
ALTER PSE jwt_pse ADD CERTIFICATE jwt_cert;
SET PSE jwt_pse PURPOSE JWT FOR PROVIDER jwt_prov;
```

5. Map roles to LDAP groups.

   This allows SAP HANA to determine which roles to assign to users based on their membership in one or more LDAP groups, either directly or indirectly through nested groups. Users' access to requested resources is then determined by the privileges defined in the SAP HANA roles.

   You can map LDAP groups to roles in the SAP HANA cockpit or using SQL as shown in the following example. You must specify the unique distinguished name (DN) of an LDAP group.

Sample Code

```
CREATE ROLE Securities_DBA
LDAP GROUP
“cn=Securities_DBA,OU=Application,OU=Groups,ou=DatabaseAdmins,cn=Users,dc=l
argebank,dc=com’;
```

Note

It is not possible to include an LDAP group in the definition of design-time roles.

Results

When a user logs on from a client application configured for SAML or JWT authentication, SAP HANA authenticates the user and then requests the user's group memberships from the LDAP server. If the authenticated user exists in your LDAP server and this user is a member of at least one LDAP group mapped to at least one SAP HANA role, a database user is created and the relevant roles granted.

Related Information

LDAP Group Authorization for Existing Users
Single Sign-On Using SAML 2.0
Single Sign-On Using JSON Web Tokens
Configuring Trust (SAP HANA Cockpit)
Create a Role (SAP HANA Cockpit)

9.5.5.3 LDAP Provider Configuration (Reference)

To set up a connection to an LDAP server, you must create an LDAP provider in the SAP HANA database. Depending on your requirements, you can use the LDAP server to authenticate and/or authorize users. For LDAP-authenticated users, you can also enable the automatic creation of users in SAP HANA.

The following table describe the information needed to configure the LDAP provider. For the exact syntax of the element, see `CREATE LDAP PROVIDER` and `ALTER LDAP PROVIDER` statements *SAP HANA SQL and System Views Reference*.

Configuration Information	SQL Clause	Description/Example
Name	`<ldap-provider-name>`	Name of the LDAP server
How the LDAP server is accessed	`CREDENTIAL TYPE`	Access take places using an LDAP server user with permissions to perform searches as specified by the user look-up URL. You must specify the distinguished name (DN) and password of this user.

Configuration Information	SQL Clause	Description/Example

The LDAP URL used to locate a requested user entry on the LDAP server	`USER LOOKUP URL`	Returns a unique user entry on the LDAP server that corresponds to current SAP HANA user
		
		The user look-up URL has the following format:
		
		`ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?<filter>`
		
		Example
		`USER LOOKUP URL 'ldaps://myhostname:389/cn=Users,dc=largebank,dc=com??sub?(&(objectClass=user)(sAMAccountName=*))'`
		
		The search for a user entry is based on the SAP HANA user name of the current user. The search filter must include a filter condition in the format `'<attribute>=*'`, where `<attribute>` is an LDAP attribute whose value is matched against the name of the SAP HANA user.
		
		Note
		The search filter must contain a single asterisk (*). An error is returned if either no asterisk or more than one is present.
		
		The asterisk (*) is replaced by the SAP HANA user name before the LDAP search query is sent to LDAP server.
		
		In the above example `(&(objectClass=user)(sAMAccountName=*))` becomes `(&(objectClass=user)(sAMAccountName=<USER_NAME of the current HANA user>))`
		
		Note
		The user look-up search query must return a single entry, that is a unique user entry corresponding to the current SAP HANA user. An error occurs if more than one entry is returned.
		
		The `<attributes>` element must remain empty in the user look-up URL. Any `<attributes>` specified in the user look-up URL are ignored. SAP HANA internally constructs the `<attributes>` list before executing the user look-up URL as follows:
		
		• If a nested group URL is specified, `<attributes>` is replaced with the attribute specified by `ATTRIBUTE DN` clause.

Configuration Information	SQL Clause	Description/Example
		• If a nested group URL is not specified, `<attributes>` is replaced with the attributes specified by `ATTRIBUTE DN` and `ATTRIBUTE MEMBER_OF` clauses.

Configuration Information	SQL Clause	Description/Example

Optional: The LDAP URL used to obtain a user's group membership information, including nested groups, from the LDAP server	`NESTED GROUP LOOKUP URL`	The nested group look-up URL returns the complete list of groups of which the LDAP user identified by the user look-up URL is a member, including groups with indirect membership.
		
		The nested group look-up URL has the following format:
		
		`ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?<filter>`
		
		Example
		`NESTED GROUP LOOKUP URL 'ldaps://myhostname:389/ou=groupsOU,dc=x??sub?(member:1.2.840.113556.1.4.1941:=*)'`
		
		The asterisk (*) in the search filter is replaced by the user DN before the nested group look-up URL search query is executed.
		
		Note
		The search filter must contain a single asterisk (*). An error is returned if either no asterisk or more than one is present.
		
		SAP HANA obtains the list of groups from the NESTED GROUP LOOKUP URL query in the following manner:
		
		• If no attributes are specified in the nested group look-up URL, the attribute specified by `ATTRIBUTE DN` clause is used.
		
		Example
		'ldap://myhostname:389/ou=groupsOU,dc=x??sub?(member:1.2.840.113556.1.4.1941:=*)'
		
		In this case, before sending the above query to the LDAP server to obtain list of groups, SAP HANA replaces the empty attribute list with the attribute specified by the ATTRIBUTE DN clause. Also, '*' in the search filter is replaced with the user DN that is obtained by executing USER LOOKUP URL query.
		• If more than one attribute is specified in the nested group look-up URL, then SAP HANA only uses the first attribute from the `<attributes>` list. Other attributes are ignored.
		
		If the nested group look-up URL query returns more than one entry, then the value of the first attribute from each

Configuration Information	SQL Clause	Description/Example
		entry is combined to obtain the complete list of groups for the user. This attribute is optional if the LDAP provider is being used for user authentication only.
The LDAP attribute that provides the distinguished name (DN) of the LDAP user entry	`ATTRIBUTE DN`	**Example** `ATTRIBUTE DN 'distinguishedName'`
Optional: The LDAP attribute that provides a list of groups that a user is a member of	`ATTRIBUTE MEMBER_OF`	**Example** `ATTRIBUTE MEMBER_OF 'memberOf'` This attribute is not required if a nested group look-up URL is specified. If both `ATTRIBUTE MEMBER_OF` and `NESTED GROUP LOOKUP URL` are specified, `NESTED GROUP LOOKUP URL` takes precedence. This attribute is optional if the LDAP provider is being used for user authentication only.
Whether or not connections between SAP HANA and the LDAP server are secured using TLS/SSL	`SSL ON | SSL OFF`	This setting applies to both LDAP access authentication and queries, including user lookup and nested groups lookup. If TLS/SSL is enabled, the URL begins with "ldap://". **→ Remember** The root certificate of the LDAP server must be available in a certificate collection with purpose *LDAP*. **Note** TLS/SSL-secured communication uses the SAP cryptographic library, CommonCryptoLib. For more information, see SAP Note 1848999. Connections to the LDAP server can also be secured using the secure LDAP protocol. In this case, the URL begins with `ldaps://` and TLS/SSL must be switched off. **Caution** Whether the LDAP provider is to be used for LDAP group authorization and/or authentication, you must secure communication between SAP HANA and the LDAP server using the TLS/SSL protocol to protect the transmission of user passwords between SAP HANA and the LDAP server.

Configuration Information	SQL Clause	Description/Example
Whether the LDAP provider is to be used for LDAP group authorization and/or authentication	`DEFAULT ON | DEFAULT OFF`	You can create several LDAP providers but only one can be in use at any time.
Whether or not the LDAP provider is enabled	`DISABLE PROVIDER | ENABLE PROVIDER`	Whether or not the LDAP provider is enabled **⚠ Caution** If the default LDAP provider is disabled, users configured for LDAP authorization and/or authentication will not be able to log on after the configured role reuse duration has expired. For more information about the reuse duration, see the *SAP HANA Security Guide*.
Optional: Whether or not the LDAP provider can create a database user in SAP HANA if required	`ENABLE USER CREATION FOR LDAP [USER TYPE {STANDARD | RESTRICTED}]`	The LDAP server creates a user in SAP HANA if all of the following are true: • LDAP authentication is enabled both in SAP HANA and for the user logging on. • The user is a member of at least one LDAP group mapped to an SAP HANA role. • A database user with this name does not already exist. By default, the user created is a standard user. If you want restricted users to be created, add the option USER TYPE RESTRICTED. The new database user is automatically configured for LDAP authentication and LDAP group authorization (authorization mode `LDAP`).

Example

The following example creates an LDAP provider for obtaining the LDAP group membership of SAP HANA users via a secure connection (LDAP with STARTTLS). The provider is enabled and set as the default for LDAP group authorization.

Sample Code

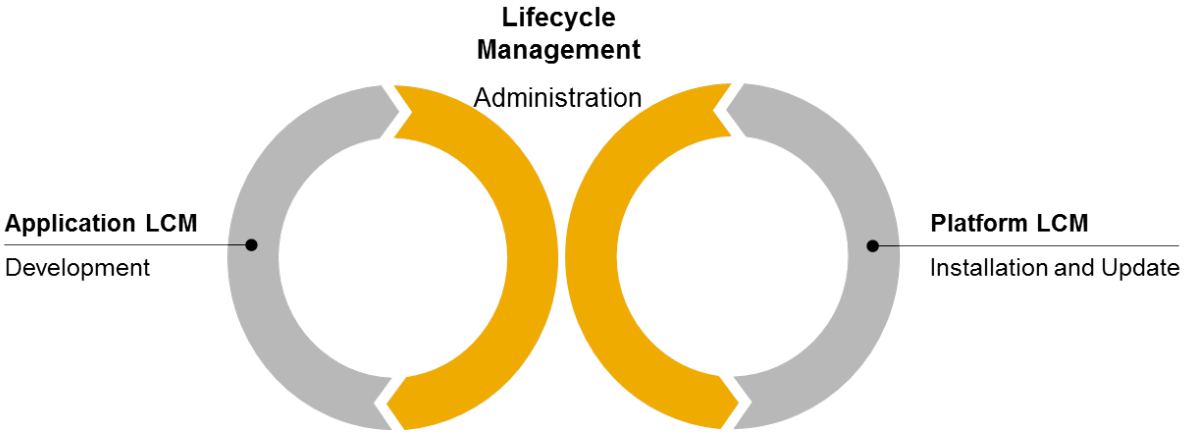
```
CREATE LDAP PROVIDER my_ldap_provider
CREDENTIAL TYPE 'PASSWORD' USING
'user=cn=LookupAccount,cn=Users,dc=largebank,dc=como;password=hUWe8ZTiQyG'
USER LOOKUP URL 'ldap://myhostname:389/cn=Users,dc=largebank,dc=com??sub?
(&(objectClass=user)(sAMAccountName=*))'
ATTRIBUTE DN 'distinguishedName'
ATTRIBUTE MEMBER_OF 'memberOf'
SSL ON
DEFAULT ON
ENABLE PROVIDER;
```

Related Information

CREATE LDAP PROVIDER Statement (Access Control)
ALTER LDAP PROVIDER Statement (Access Control)

10 SAP HANA Lifecycle Management

SAP HANA lifecycle management covers two aspects: platform lifecycle management for customizing and updating your SAP HANA platform and application lifecycle management for managing SAP HANA content products and transports.



Platform Lifecycle Management Aspects

You can customize platform lifecycle management aspects of your SAP HANA system by accessing the SAP HANA database lifecycle manager from three user interfaces: the graphical user interface, the command-line interface, or the Web user interface in a stand-alone Web browser, in the SAP HANA studio, or via the SAP HANA cockpit.

SAP HANA platform lifecycle management encompasses the installation and update of an SAP HANA server, mandatory components, and additional components, as well as the post-installation configuration. The concepts and procedures for SAP HANA platform installation and update are described in the *SAP HANA Server Installation and Update Guide* on SAP Help Portal.

A number of system configuration features are integrated into the SAP HANA database lifecycle manager, such as:

- The initial configuration of your SAP HANA platform to integrate it into your landscape. For example, by registering it in a system landscape directory, or configuring the inter-service communication.
- Adapting the topology of your SAP HANA platform by adding or removing additional SAP HANA hosts.
- Reconfiguring the system.

System configuration as it pertains to SAP HANA lifecycle management is described in the *SAP HANA Platform Lifecycle Management* section of this *SAP HANA Administration Guide*.

Application Lifecycle Management Aspects

SAP HANA application lifecycle management aspects can be accessed in different user interfaces: an interface that runs as an SAP HANA XS application in a web browser, a command-line tool hdbalm, integrated in SAP HANA studio, or via the SAP HANA cockpit.

SAP HANA application lifecycle management supports you in all phases of the lifecycle of an SAP HANA application or add-on product, from modelling your product structure, through application development, transport, assembly, to installing and updating products that you have downloaded from SAP Support Portal or which you have assembled yourself.

All application lifecycle management tasks are documented in the guide *SAP HANA Application Lifecycle Management* on SAP Help Portal.

System administrators use SAP HANA application lifecycle management mainly to install and update SAP HANA applications or add-on products. Therefore, these tasks are documented in this *SAP HANA Administration Guide*. Tasks related to SAP HANA development are documented in the *SAP HANA Developer Guide - For SAP HANA Studio* (on SAP Help Portal) under *SAP HANA Application Lifecycle Management*.

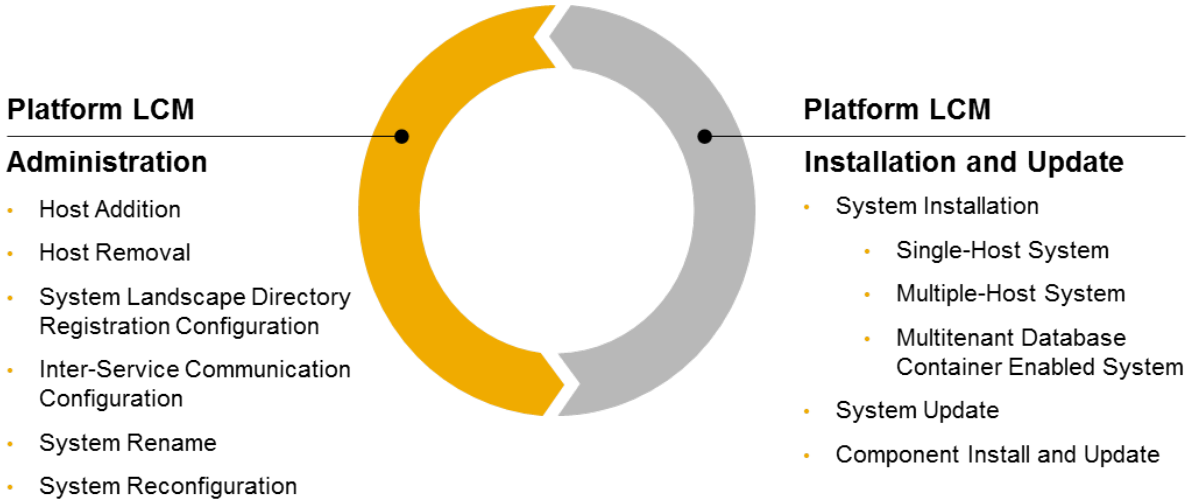
Related Information

SAP HANA Platform Lifecycle Management [page 583]
SAP HANA Application Lifecycle Management [page 617]

10.1 SAP HANA Platform Lifecycle Management

After the SAP HANA system is installed, it can be configured on the system level.

The SAP HANA platform lifecycle management (LCM) information in this *SAP HANA Administration Guide* details platform administration and configuration. For information about installing and updating the SAP HANA system, see the *SAP HANA Server Installation and Update Guide* on SAP Help Portal.



The SAP HANA database lifecycle manager provides flexibility to accommodate all types of administrators. Before performing administration tasks using the SAP HANA database lifecycle manager, consider reviewing the topic *Using the SAP HANA Platform LCM Tools* to understand the available user interfaces, interaction modes, and parameter entry methods.

You can use the SAP HANA database lifecycle manager to perform the following administration tasks:

- Configure the system
  - Configure a multiple-host system
    - Add one or more hosts to a system
    - Remove one or more hosts from a system
  - Configure a connection to the System Landscape Directory (SLD)
  - Configure inter-service communication
- Change the existing system
  - Change the system identifiers
    - Rename system hosts
    - Change the SID
    - Change the instance number
  - Reconfigure the system
    - Relocate the system to new hardware
    - Copy or clone the system
    - Convert the system to a multitenant database container enabled system

Related Information

Using the SAP HANA Platform LCM Tools [page 585]
Configuring an SAP HANA System to Connect to the System Landscape Directory (SLD) [page 46]
Configuring SAP HANA Inter-Service Communication [page 1116]
Rename an SAP HANA System Host [page 708]
Change the SID of an SAP HANA System [page 710]
Change the Instance Number of an SAP HANA System [page 712]
Relocate the SAP HANA System [page 666]
Copy or Clone an SAP HANA System [page 669]
Converting an SAP HANA System to Support Tenant Databases [page 66]

10.1.1 About the SAP HANA Database Lifecycle Manager (HDBLCM)

The SAP HANA database lifecycle manager (HDBLCM) is used to install, update, or configure an SAP HANA system. You can use the SAP HANA database lifecycle manager in graphical user, command-line, or Web user interface.

10.1.1.1 Using the SAP HANA Platform LCM Tools

The SAP HANA database lifecycle manager (HDBLCM) is used to perform SAP HANA platform lifecycle management (LCM) tasks, including installing, updating, and configuring an SAP HANA system. The SAP HANA database lifecycle manager is designed to accommodate hardware partners and administrators, and so it offers a variety of usage techniques.

The SAP HANA database lifecycle manager is used by means of program interface type, program interaction mode, and parameter entry mode. Before using the SAP HANA database lifecycle manager, you should choose which user interface you prefer to use and how you want to modify the platform LCM task to achieve your desired result. You modify the actions of the platform LCM tools using parameters. Parameters can be modified in a number of ways, for example, in the entry field of a graphical interface, as a call option with the program call, or in a configuration file. These options can be mixed and matched depending on the parameters you need to use and the program interaction mode you choose.

Platform LCM Tools and Program Interaction Modes

	Interactive Mode	Advanced Interactive Mode	Batch Mode
Graphical User Interface	■	■	◇
Command-Line Interface	■	■	■
Web User Interface	■	◇	◇

The first choice to make is which SAP HANA database lifecycle manager (HDBLCM) interface type you prefer to use. The SAP HANA HDBLCM program can be run as a graphical user interface, a command-line interface, or as Web user interface in a Web browser (the Web user interface is not available for all platform LCM tasks).

Once you've chosen the graphical user, command-line, or Web user interface, you can decide if you prefer to interactively enter parameter values, or give all required parameters with the call to the platform LCM tool, and let it run unattended to completion. Interactive mode is available for all user interfaces, and is the default mode for program interaction. To use interactive mode, you simply call the SAP HANA HDBLCM user interface, and enter parameter values as they are requested by the program. Advanced interactive mode involves entering some parameter values interactively and providing some parameter values as call options or in a configuration file. This is the recommended interaction mode if you'd like to modify parameter default values which are not requested in interactive mode. Batch mode is an advanced platform LCM interaction method because all required parameters must be provided with the call to the LCM program on the command line. Batch mode is designed for large-scale platform LCM tasks, which would be time consuming to perform interactively.

Platform LCM parameters can be entered interactively (only available for interactive mode or advanced interactive mode), as a call option on the command line, or via a configuration file. If you are performing platform LCM tasks in advanced interactive mode, you can choose any of the three parameter entry methods

(or use more than one). If you are using batch mode, you must enter parameter values either as call options to the SAP HANA database lifecycle manager or from a configuration file. The syntax for the parameters as call options can be found in the *Parameter Reference*. The configuration file is generated as a blank template, then edited, and called as a call option.

Related Information

Use Interactive Mode to Perform Platform LCM Tasks [page 594]
Use Advanced Interactive Mode to Perform Platform LCM Tasks [page 595]
Use Batch Mode to Perform Platform LCM Tasks [page 597]

10.1.1.1.1 Choosing the Correct SAP HANA HDBLCM for Your Task

It is important to distinguish between the version of the SAP HANA database lifecycle manager (HDBLCM) that is available on the installation medium and the version that is unpacked during installation, and subsequently used to perform administration and configuration tasks after the SAP HANA system has been installed.

The SAP HANA database lifecycle manager is available in two varieties - an installation medium version to perform installation and update, and a resident version for update and configuration that is unpacked on the SAP HANA host during installation or update. The SAP HANA resident HDBLCM has been designed to be version-compatible. That means, every time you install or update an SAP HANA system, you can be sure that any subsequent configuration tasks performed with the SAP HANA database lifecycle manager will work as expected because the installation or update tool and the configuration tool are of the same version and have been tested together. The SAP HANA resident HDBLCM is located at `<sapmnt>/<SID>/hdblcm`.

10.1.1.1.2 Performing LCM Tasks by Program Interface

SAP HANA platform lifecycle management tasks can be performed from a graphical, command-line and Web user interface.

Related Information

Use the Graphical User Interface to Perform Platform LCM Tasks [page 587]
Use the Command-Line Interface to Perform Platform LCM Tasks [page 588]
Using the Web User Interface [page 589]

10.1.1.1.2.1 Use the Graphical User Interface to Perform Platform LCM Tasks

SAP HANA platform lifecycle management tasks can be performed from a graphical interface.

Procedure

1. Change to the directory where the SAP HANA database lifecycle manager is located:

Option	Description
Installation Medium (Intel-Based Hardware Platforms)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64`
Installation Medium (IBM Power Systems)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_PPC64`
Installation Archive downloaded from SAP Support Portal (SAP Service Marketplace)	`cd SAP_HANA_DATABASE`
SAP HANA resident HDBLCM	`cd <sapmnt>/<SID>/hdblcm`

   In general, installation and update is carried out from the installation medium. Configuration tasks are performed using the SAP HANA resident HDBLCM. For more information about the two SAP HANA database lifecycle manager types, see Related Information.

2. Start the SAP HANA platform lifecycle management tool:

   ```
   ./hdblcmgui
   ```

3. Enter parameter values in the requested fields. In addition, you can specify parameter key-value pairs as call options or in the configuration file template.

   > ⓘ Note
   >
   > If parameter key-value pairs are specified as command-line options, they override the corresponding parameters in the configuration file. Parameters in the configuration file override default settings.
   >
   > **Order of parameter precedence:**
   >
   > Command Line > Configuration File > Default

   For more information about program interaction modes and parameter values entry methods, see Related Information.

Related Information

Choosing the Correct SAP HANA HDBLCM for Your Task [page 586]
Entering Platform LCM Parameters as Call Options from the Command Line [page 601]

10.1.1.1.2.2 Use the Command-Line Interface to Perform Platform LCM Tasks

SAP HANA platform lifecycle management tasks can be performed from the command line.

Procedure

1. Change to the directory where the SAP HANA database lifecycle manager is located:

Option	Description
Installation Medium (Intel-Based Hardware Platforms)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64`
Installation Medium (IBM Power Systems)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_PPC64`
Installation Archive downloaded from SAP Support Portal (SAP Service Marketplace)	`cd SAP_HANA_DATABASE`
SAP HANA resident HDBLCM	`cd <sapmnt>/<SID>/hdblcm`

   In general, installation and update is carried out from the installation medium. Configuration tasks are performed using the SAP HANA resident HDBLCM. For more information about the two SAP HANA database lifecycle manager types, see Related Information.

2. Start the SAP HANA platform lifecycle management tool:

```
./hdblcm
```

3. Enter parameter values in one of the following ways.
   - **Interactive parameter entry** - If you call the SAP HANA platform LCM tool only, the program runs in interactive mode. Parameter default values are suggested in brackets, and can be accepted with *Enter*. Otherwise, enter a non-default value, then select *Enter*.
   - **Command-line parameter entry as call options** - If you enter parameter key-value pairs as call options with the call to the SAP HANA platform LCM tool, the program runs in advanced interactive mode and requests values for any parameter values which you didn't specify in the original input. If you entered the batch mode call option, the program runs to completion without any further requests, unless a mandatory parameter was left out of the original input, in which case, the program fails to perform the platform LCM task.

- **Configuration file parameter entry** - If you enter parameter key-value pairs in the configuration file template, and enter the configuration file path as a call option with the call to the SAP HANA platform LCM tool, the program runs in advanced interactive mode and requests values for any parameter values which you didn't specify in the original input. If you entered the batch mode call option, the program runs to completion without any further requests, unless a mandatory parameter was left out of the original input, in which case, the program fails to perform the platform LCM task.

> **Note**
>
> If parameter key-value pairs are specified as command-line options, they override the corresponding parameters in the configuration file. Parameters in the configuration file override default settings.
>
> **Order of parameter precedence:**
>
> Command Line > Configuration File > Default

For more information about program interaction modes and parameter values entry methods, see Related Information.

Related Information

Choosing the Correct SAP HANA HDBLCM for Your Task [page 586]
Performing LCM Tasks by Parameter Entry Method [page 598]
Performing LCM Tasks by Program Interaction Mode [page 594]
Entering Platform LCM Parameters as Call Options from the Command Line [page 601]

10.1.1.1.2.3 Using the Web User Interface

SAP HANA platform lifecycle management tasks can be performed using the SAP HANA database lifecycle manager (HDBLCM) Web user interface.

10.1.1.1.2.3.1 About the Web User Interface

The SAP HANA database lifecycle manager (HDBLCM) Web user interface is hosted by the SAP Host Agent, which is installed on the SAP HANA host. When installing or updating the SAP HANA system, as part of the SAP HANA resident HDBLCM configuration, the SAP HANA system deploys its artifacts on the SAP Host Agent, thus enabling the Web user interface.

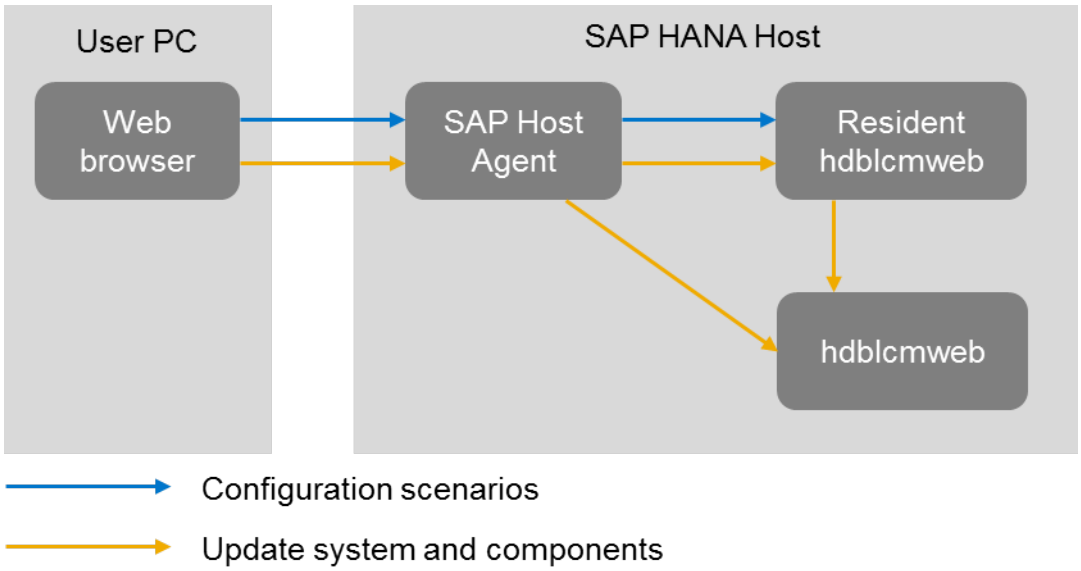
All Web user interface actions are always performed in the context of an already installed and registered SAP HANA system. In order to access the SAP HANA database lifecycle manager Web user interface you need to log on as the system administrator user `<sid>`adm.

The communication between the Web browser and the SAP Host Agent is always done over HTTPS, which requires that the SAP Host Agent has a secure sockets layer (SSL) certificate (PSE) in its security directory. For more information about SSL certificate handling, see Related Information.

The backend is provided by the special executable `hdblcmweb`, which is started automatically by the SAP Host Agent as soon as an action is triggered from the Web user interface and terminates after the action completes.

> **Note**
>
> You should never start `hdblcmweb` manually. For security reasons, `hdblcmweb` is always started with system administrator user `<sid>`adm privileges. If you require logging with individual users (to ensure personalized logging), use the SAP HANA database lifecycle manager graphical user or command-line interface.

> **Note**
>
> Make sure that the system administrator user `<sid>`adm has permissions to read the paths, passed as parameters in the Web user interface (for example, the SAP HANA database installation kit or locations with SAP HANA components).



One platform LCM task, which is worth special attention is the update of the SAP HANA system and components. The SAP HANA system updates are always performed by the installation kit SAP HANA database lifecycle manager in the graphical user and command-line interfaces, (and not the SAP HANA resident HDBLCM). This is because the SAP HANA database lifecycle manager, in the graphical user and command-line interfaces, is not forward compatible. Meaning that only the new version of the tool knows how to update an older system.

On the other hand, all scenarios in the Web user interface are handled by the SAP HANA resident HDBLCM, which is part of the system. For this reason, as a first step before even starting the update process, you are required to enter a location of an SAP HANA database installation kit. After detecting the kit, the update Web user interface is loaded from the installation kit and the installation kit SAP HANA database lifecycle manager starts serving as backend until the update process finishes. It is as if you start the SAP HANA database lifecycle manager directly from the installation kit in graphical user or command-line interface.

Related Information

Secure Sockets Layer (SSL) Certificate Handling [page 604]

10.1.1.1.2.3.2 Use the Web User Interface to Perform Platform LCM Tasks

The SAP HANA database lifecycle manager (HDBLCM) can be accessed as a Web user interface in either a standalone browser or in the SAP HANA cockpit.

Prerequisites

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

  > **Note**
  >
  > For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.
- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Context

The Web user interface supports only the following SAP HANA platform lifecycle management tasks:

- View system information
- Update system and components
- Install or update additional components
- Configure System Landscape Directory (SLD) registration
- Configure inter-service communication

When performing installation and update tasks, various parameters can be set in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Procedure

Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://<hostname>:1129/lmsl/HDBLCM/<SID>/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

Results

The SAP HANA database lifecycle manager is displayed as a Web user interface in either a standalone browser or in the SAP HANA cockpit.

Related Information

SAPUI5 Developer Guide
Add an SAP HANA System

10.1.1.1.2.3.3 Log Off From an SAP HANA System

In the SAP HANA database lifecycle manager (HDBLCM) Web user interface, you can log off from an SAP HANA system and close all connections to the system. To be able to connect to system again, you must log on.

Procedure

- To log off from a system click the *Log out* button.
  All open connections to the system are closed.

  > **Note**
  >
  > Currently, this feature is not available for browsers on mobile devices.

10.1.1.1.2.3.4 Troubleshooting the Web User Interface

If you have problems with the Web user interface, see SAP Note 2078425 for steps you can take to troubleshoot and resolve them.

> **Note**
>
> The Web browser used to render the platform lifecycle management Web user interface in the SAP HANA studio **cannot** be changed via *Windows* > *Preferences* > *General* > *Web Browser*.

Related Information

SAP Note 2078425

10.1.1.1.3 Performing LCM Tasks by Program Interaction Mode

SAP HANA platform lifecycle management tasks can be performed in interactive mode, advanced interactive mode and batch mode.

10.1.1.1.3.1 Use Interactive Mode to Perform Platform LCM Tasks

Interactive mode is a method for running SAP HANA platform lifecycle management (LCM) tools which starts the program and requires you to enter parameter values successively before the program is run. Interactive mode is the default mode for the SAP HANA platform LCM tools.

Context

In general, installation and update is carried out from the installation medium. Configuration tasks are performed using the SAP HANA resident HDBLCM. For more information about the different SAP HANA database lifecycle manager types, see Related Information.

The SAP HANA platform LCM tools offer a wide variety of parameters which can modify the platform LCM task you are performing. Some parameters can be modified in interactive mode when the graphical user, command-line, or Web user interface requests a value for a given parameter.

Procedure

1. Change to the directory where the SAP HANA database lifecycle manager is located:

Option	Description
Installation Medium (Intel-Based Hardware Platforms)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64`
Installation Medium (IBM Power Systems)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_PPC64`
SAP HANA resident HDBLCM	`cd <sapmnt>/<SID>/hdblcm`

   To access the SAP HANA database lifecycle manager Web user interface, see Related Information.

2. Start the SAP HANA platform lifecycle management tool:

Option	Description
Graphical Interface	`./hdblcmgui`
Command-line Interface	`./hdblcm`

To start the SAP HANA platform LCM tools in interactive mode, simply **do not** enter the parameter for batch mode (`--batch` or `-b`) as a call option. You can enter any other required parameters as call options or load a configuration file. The program runs in interactive mode and requests any missing parameters values, which must be verified or changed. You are provided with a summary of parameter values, which you can accept to run the program to completion, or reject to exit the program.

Related Information

Choosing the Correct SAP HANA HDBLCM for Your Task [page 586]
Use the Web User Interface to Perform Platform LCM Tasks [page 591]

10.1.1.1.3.2 Use Advanced Interactive Mode to Perform Platform LCM Tasks

Interactive mode is a method for running SAP HANA platform lifecycle management (LCM) tools which starts the program and requires you to enter parameter values successively before the program is run. If you would like to enter call options not available in interactive mode, or make use of the configuration file, you can use a combination of interactive mode and advanced parameter entry methods.

Context

In general, installation and update is carried out from the installation medium. Configuration tasks are performed using the SAP HANA resident HDBLCM. For more information about the different SAP HANA database lifecycle manager types, see Related Information.

The SAP HANA platform LCM tools offer a wide variety of parameters which can modify the platform LCM task you are performing. Some parameters can be modified in interactive mode when the graphical user, command-line, or Web user interface requests a value for a given parameter. However, some parameters are not available in interactive mode, and must be specified either as a call option with the call to the platform LCM tool, or from within a configuration file.

Procedure

1. Review which parameters are offered in interactive mode.

   If the parameter you want to configure is not available in interactive mode, you have two options. You can either enter the parameter key-value pair as a call option with the call to the platform LCM tool. Alternatively, you can generate a configuration file template, and edit the parameters value in the configuration file. Then call the configuration file as a call option with the call to the platform LCM tool.

   Using the configuration file for interactive mode is recommended if you plan to perform the exact same platform LCM task multiple times.

2. Change to the directory where the SAP HANA database lifecycle manager is located:

Option	Description
Installation Medium (Intel-Based Hardware Platforms)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64`
Installation Medium (IBM Power Systems)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_PPC64`
SAP HANA resident HDBLCM	`cd <sapmnt>/<SID>/hdblcm`

3. If you plan to use a configuration file, prepare it with the following steps:
   a. Generate the configuration file template using the SAP HANA platform lifecycle management tool:

      Run the SAP HANA platform LCM tool using the parameter `dump_configfile_template` as a call option. Specify an action and a file path for the template. A configuration file template and a password file template are created.

      ```
      ./hdblcm --action=<LCM action> --dump_configfile_template=<file path>
      ```

   b. Edit the configuration file parameters. Save the file.
   c. Edit the password file. Save the file.
4. Start the SAP HANA platform lifecycle management tool:

   Start the SAP HANA database lifecycle manager in either the graphical user interface or in the command-line interface, with a call option:

   ```
   ./hdblcmgui --<parameter key>=<parameter value>
   ```

   or

   ```
   ./hdblcm --<parameter key>=<parameter value>
   ```

   If you are using a configuration file, you must use the call option `--configfile=<file path>`.

Related Information

Choosing the Correct SAP HANA HDBLCM for Your Task [page 586]

10.1.1.1.3.3 Use Batch Mode to Perform Platform LCM Tasks

Batch mode is a method for running the SAP HANA database lifecycle manager which starts the program and runs it to completion without requiring you to interact with it any further. All required parameter values must be passed as call options or from a configuration file.

Prerequisites

- When using batch mode, passwords must either be defined in the configuration file, or passed to the installer using an XML password file and streamed in via standard input. In both cases, it is necessary to prepare the passwords. For more information, see *Specifying Passwords*.

Context

In general, installation and update is carried out from the installation medium. Configuration tasks are performed using the SAP HANA resident HDBLCM. For more information about the different SAP HANA database lifecycle manager types, see Related Information.

If you are new to performing the desired SAP HANA platform LCM task in batch mode, it is recommended to run some tests before using batch mode in a production environment.

Procedure

1. Change to the directory where the SAP HANA database lifecycle manager is located:

Option	Description
Installation Medium (Intel-Based Hardware Platforms)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64`
Installation Medium (IBM Power Systems)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_PPC64`
SAP HANA resident HDBLCM	`cd <sapmnt>/<SID>/hdblcm`

2. Start the SAP HANA platform lifecycle management tool:

```
./hdblcm --batch <additional parameters>
```

or

```
./hdblcm -b <additional parameters>
```

It is mandatory to provide an SAP HANA system ID (SID) and user passwords during installation. In batch mode, you are restricted to providing these parameter values as call options on the command line (for passwords, by means of an XML file) or in a configuration file. If you don't provide parameter values for the other required parameters, you implicitly accept the default values.

Example

The following example installs the SAP HANA server and client as a single-host system. The SAP system ID and instance number are also specified from the command line. The system passwords are read from a standard input stream by the installer. All other parameter defaults are automatically accepted and no other input is requested in order to complete the installation.

```
cat ~/hdb_passwords.xml | ./hdblcm --batch --action=install --
components=client,server --sid=DB1 --number=42  --read_password_from_stdin=xml
```

If a configuration file is used in combination with batch mode, an identical system can be installed with a simplified call from the command line. In the following example, passwords are defined in the configuration file, in addition to the action, components, SAP system ID, and instance number.

```
./hdblcm --batch --configfile=/var/tmp/H01_configfile
```

Related Information

Choosing the Correct SAP HANA HDBLCM for Your Task [page 586]
Specifying Passwords
Use LCM Configuration Files to Enter Parameters [page 599]
Entering Platform LCM Parameters as Call Options from the Command Line [page 601]

10.1.1.1.4 Performing LCM Tasks by Parameter Entry Method

SAP HANA platform lifecycle management (LCM) parameter values can be entered in a variety of methods: interactively by iteratively providing values in either the graphical interface or command prompt, as command-line options with the call to the platform LCM tool, or in a configuration file.

SAP HANA platform lifecycle management parameter values allow you to customize your SAP HANA installation, update, or configuration. Parameter values can be entered by **one or more** of the following methods:

Interactively (Default)	Using either command line interface, the graphical interface or the Web user interface, most parameters are requested interactively. Default parameter values are proposed in brackets and can be changed or confirmed. Parameters that are not requested (or specified via another method) accept the default value.

Command Line Options	Parameters are given in their accepted syntax as a space delimited list after the program call (for example, `hdblcm` or `hdblcmgui`). The specified parameters replace the defaults. If any mandatory parameters are excluded, they are requested interactively (unless batch mode is specified). All parameters can be entered from the command line. For more details about the accepted parameter syntax, see the inline help output (`--help`) for the individual SAP HANA lifecycle management tool.
Configuration File	The configuration file is a plain text file, for which a template of parameter key-value pairs can be generated, edited, and saved to be called in combination with the program call. If any mandatory parameters are not specified, they are requested interactively (unless batch mode is used). All parameters can be entered in the configuration file. For more information about the configuration file, see Related Information.

> ### Note
>
> If parameters are specified in the command line, they override the corresponding parameters in the configuration file. Parameters in the configuration file override default settings.
>
> **Order of parameter precedence:**
>
> Command Line > Configuration File > Default

10.1.1.1.4.1 Entering Platform LCM Parameters Interactively

SAP HANA platform LCM interactive mode is default interaction mode for all platform LCM programs and interfaces.

You can run the graphical, command-line, or Web user interface in interactive mode by simply starting the program, and entering parameter values as they are requested by the program. In interactive mode, parameter default values are suggested in brackets and can be accepted with `Enter`.

Not all parameters are requested in interactive mode. If you would like to configure a parameter not offered in interactive mode, you must enter it as a call option with the call to the platform LCM program, or use corresponding configuration file for the platform LCM task.

10.1.1.1.4.2 Use LCM Configuration Files to Enter Parameters

By defining a prepared configuration file during installation, specified parameter values are used by the SAP HANA platform lifecycle management (LCM) tools to build a customized SAP HANA system.

Context

The configuration file is a plain text file of specified parameters, written in the same syntax as in the command line (except without the leading two dashes --). A configuration file template can be generated, edited, and saved to be called with the call to the SAP HANA database lifecycle manager (HDBLCM).

The configuration file template provides a brief, commented-out summary of each parameter. Each parameter is set to its default value.

Procedure

1. Change to the directory where the SAP HANA database lifecycle manager is located:

Option	Description
Installation Medium (Intel-Based Hardware Platforms)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64`
Installation Medium (IBM Power Systems)	`cd <installation medium>/DATA_UNITS/HDB_LCM_LINUX_PPC64`
Installation Archive downloaded from SAP Support Portal (SAP Service Marketplace)	`cd SAP_HANA_DATABASE`
SAP HANA resident HDBLCM	`cd <sapmnt>/<SID>/hdblcm`

   In general, installation and update is carried out from the installation medium. Configuration tasks are performed using the SAP HANA resident HDBLCM. For more information about the two SAP HANA database lifecycle manager types, see Related Information.

2. Generate the configuration file template using the SAP HANA platform lifecycle management tool:

   Run the SAP HANA platform LCM tool using the parameter `dump_configfile_template` as a call option. Specify an action and a file path for the template. A configuration file template and a password file template are created.

   ```
   ./hdblcm --action=<LCM action> --dump_configfile_template=<file path>
   ```

3. Edit the configuration file parameters. Save the file.

   It is recommended that at least the SAP system ID (`sid`) and the instance number (`number`) are uniquely defined. There are several required parameters, that are provided default values in case they are not customized. For more information, refer to the default values.

   Some file path parameters have automatic substitution values as part of the default file path, using the `sid` (SAP HANA system ID) and `sapmnt` (installation path) parameters, so that the substituted values create file paths that are unique and system-specific. For example, the default for the data file path is: `datapath=/hana/data/${sid}`, where `sid` is automatically replaced by the unique SAP HANA system ID.

4. Start the SAP HANA platform lifecycle management tool:

   Run the SAP HANA platform LCM tool using the parameter `configfile` as a call option. Specify the file path of the edited template.

   ```
   ./hdblcm --configfile=<file path>
   ```

You can specify the path to a directory in which custom configuration files are saved using the parameter `custom_cfg` as a call option.

Related Information

configfile
custom_cfg
Choosing the Correct SAP HANA HDBLCM for Your Task [page 586]

10.1.1.1.4.3 Entering Platform LCM Parameters as Call Options from the Command Line

Call options are available for every SAP HANA platform LCM program.

You can use call options for a number of reasons:

- The parameter is not available in interactive mode, but can be entered as a call option.
- You are using batch mode.
- You are using a configuration file, but would like to override a parameter in the configuration file with a new value.
- You are installing an SAP HANA multiple-host system from the command line.

A call option is entered with the following notation:

```
./<program call> --<parameter1 key>=<parameter1 value> --<parameter2
key>=<parameter2 value>
```

Call options start with a double dash (`--`) if they are written in long-form syntax. Some parameters also have short-form syntax, in which they are preceded with a single dash (`-`). For more information about call option syntax, see the *Parameter Reference* topics.

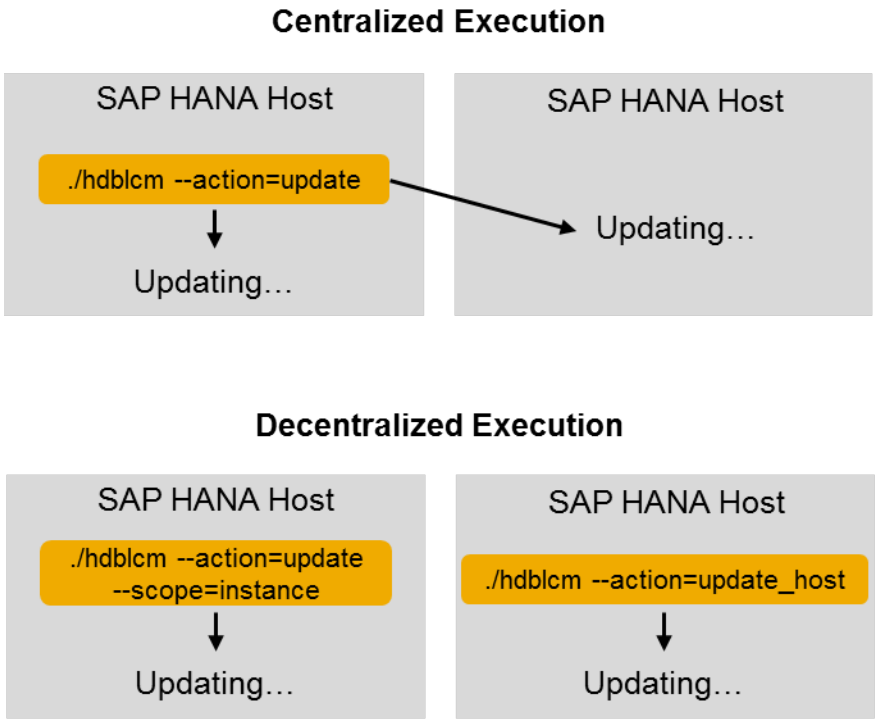
Related Information

Parameter Reference

10.1.1.1.5 Executing Platform LCM Tasks

SAP HANA platform lifecycle management tasks can be performed on multiple-host systems centrally, by running the SAP HANA database lifecycle manager (HDBLCM) from any worker host and using remote execution to replicate the call on all remaining system hosts. Otherwise, the platform LCM tasks can be executed first on a worker host, and then re-executed manually on each remaining host. This method is considered decentralized execution.

The following is an example of an SAP HANA system update performed centrally and decentrally.



Related Information

Centralized Execution of Platform LCM Tasks [page 602]
Decentralized Execution of Platform LCM Tasks [page 607]

10.1.1.1.5.1 Centralized Execution of Platform LCM Tasks

SAP HANA platform lifecycle management (LCM) tasks can be performed centrally on multiple-host SAP HANA systems in a number of ways depending on the available certificate keys and the remote execution configuration.

10.1.1.1.5.1.1 Using Secure Shell (SSH) to Execute Platform LCM Tasks

An SAP HANA system must be installed with root user credentials. During installation a secure shell (SSH) key is configured so that future platform LCM tasks can be performed remotely on multiple-host SAP HANA systems without requiring the root user password.

By default, the SAP HANA database lifecycle manager (HDBLCM) uses SSH during SAP HANA system installation or update. In order to use SSH, the SFTP subsystem must be active. Install the SAP Host Agent

on all system hosts to perform platform LCM tasks without root credentials. Once the SAP Host Agent is installed, it is used to perform any platform LCM tasks executed from the Web user interface or as the system administrator user <sid>adm.

> **Note**
>
> Platform LCM tasks cannot be executed remotely via SSH as the system administrator user <sid>adm.

The root user name must be the same for all hosts in a multiple-host system. The password of the root user must be identical on all hosts. If the passwords are not identical on all hosts or if you do not want to pass the passwords to the SAP HANA database lifecycle manager (HDBLCM), SSH keys can be distributed between the hosts in advance. Alternatively, platform LCM tasks can be executed without root credentials by using the SAP Host Agent.

Related Information

SAP Note 1944799
SAP Note 2009879
Using SAP Host Agent to Execute Platform LCM Tasks [page 603]

10.1.1.1.5.1.2 Using SAP Host Agent to Execute Platform LCM Tasks

Platform LCM tasks can be executed without root credentials by using the SAP Host Agent. The SAP Host Agent is installed and updated by default during SAP HANA system installation and update.

The SAP HANA database lifecycle manager (HDBLCM) relies on the SAP Host Agent for the following functionality to work:

- Execution as the system administrator user <sid>adm
- Connectivity to remote hosts via HTTPS (when no SSH or root user credentials are available)
- Execution from the SAP HANA database lifecycle manager Web user interface

> **Note**
>
> The SAP HANA cockpit uses the SAP Host Agent to execute tasks as the system administrator user <sid>adm, for example, stopping and starting the system, or troubleshooting a system experiencing performance problems.

If execution on the remote hosts is done via SSH (default, `--remote_execution=ssh`), the SAP HANA database lifecycle manager is able to connect to a remote host via SSH and install and configure the SAP Host Agent. In contrast, the remote execution via SAP Host Agent (`--remote_execution=saphostagent`) requires that the SAP Host Agent is installed and configured on all involved hosts in advance, which includes:

- Install SAP Host Agent
- Configure a Secure Sockets Layer (SSL) certificate for the SAP Host Agent, so that the HTTPS port 1129 is accessible. If you don't want to configure HTTPS, it is also possible to use the call option `--use_http`.

It tells the SAP HANA database lifecycle manager to communicate with the SAP Host Agent via HTTP. During the addition of new hosts to an SAP HANA system (also during the installation of a multiple-host system), the HTTPS of the SAP Host Agent is automatically configured by the SAP HANA database lifecycle manager.

> ⚠ Caution
>
> Use the call option `--use_http` with caution, because passwords are also transferred in plain text via HTTP.

Related Information

SAP Host Agent
Configuring SSL for SAP Host Agent on UNIX

10.1.1.1.5.1.2.1 Secure Sockets Layer (SSL) Certificate Handling

To enable secure communication with the SAP Host Agent over HTTPS, the SAP Host Agent needs a secure sockets layer (SSL) certificate in its security directory. This certificate is also used by the SAP HANA database lifecycle manager (HDBLCM) Web-based user interface because the Web pages are served by the SAP Host Agent.

The SAP HANA database lifecycle manager handles certificate management during system installation, update, or rename, as well as during the addition of new hosts as follows:

- If there is no certificate in the SAP Host Agent security directory, the SAP HANA database lifecycle manager generates one. The SAP HANA host name is used as the default certificate owner. The certificate owner can be changed by using the call option `--certificates_hostmap=<fully_qualified_domain_name>`.
- If there is an existing certificate, the following applies:
  - If the certificate host name is not passed to the SAP HANA database lifecycle manager, or if the certificate host name is the same as the owner of the current certificate, the current certificate is preserved.
  - If the certificate host name is passed via the call option `--certificates_hostmap` and it differs from the owner of the current certificate, a new certificate is generated.
  - During update of an SAP HANA system, if the certificates on all hosts are in place, the call option `--certificates_hostmap` is ignored and the current certificates are preserved.

If you want to use your own SSL certificates, see the SAP Host Agent documentation in Related Information.

Related Information

certificates_hostmap
SAP Note 1907566
SSL Configuration for the SAP Host Agent

10.1.1.1.5.1.2.2 Starting Platform LCM Tasks as the System Administrator User <sid>adm

When starting platform LCM tasks as the system administrator user `<sid>`adm, the SAP HANA database lifecycle manager (HDBLCM) requires the usage of SAP Host Agent for execution of remote and local operations.

The following tasks in the SAP HANA database lifecycle manager can be performed as the system administrator user `<sid>`adm:

- System update from the installation medium
- Installation or update of additional components from the SAP HANA resident HDBLCM
- Host addition and host removal
- System Landscape Directory (SLD) registration configuration
- Inter-service communication configuration

Make sure that SAP Host Agent is installed and configured (HTTPS-enabled) on all hosts of the SAP HANA system.

> **Note**
>
> Platform LCM tasks cannot be executed remotely via SSH as the system administrator user `<sid>`adm.

> **Note**
>
> Make sure that the system administrator user `<sid>`adm has permissions to read the paths passed as parameters (for example, the locations of the SAP HANA components).

10.1.1.1.5.1.2.3 Add Hosts Using SAP Host Agent

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in combination with the SAP Host Agent in the command-line interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).

- The host which is to be added has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The latest supported version of SAP Host Agent is installed on the host which is to be added. The SAP Host Agent will create the`<sapsys>` group, if it does not exist prior to installation. Make sure that the group ID of the `<sapsys>` group is the same on all hosts. For information about installing or updating the SAP Host Agent individually, see *Installing SAP Host Agent Manually* and *Upgrading SAP Host Agent Manually*.
- A Secure Sockets Layer (SSL) certificate is configured for the SAP Host Agent, so that the HTTPS port 1129 is accessible and the Personal Security Environment (PSE) for the server is prepared. For more information about SSL configuration for the SAP Host Agent, see *Configuring SSL for SAP Host Agent on UNIX*.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --remote_execution=saphostagent
   ```
3. Select the index for the *add_hosts* action.
4. Enter the names of the hosts to be added.
5. Enter the SAP Host Agent administrator (`sapadm`) password.
6. Define additional system properties.
7. Review the summary, and select *y* to finalize the configuration.

Results

You have added one or more new hosts to an SAP HANA system. The SAP HANA system you have configured is a multiple-host system.

The new hosts have been added to the SAP HANA landscape information. The new hosts have been added to the landscape information of the system database.

This configuration task can also be performed in batch mode and using a configuration file. For more information about the available configuration methods, see *Using the SAP HANA Platform LCM Tools*.

Related Information

remote_execution
Host Addition Concepts [page 1086]
Using the SAP HANA Platform LCM Tools [page 585]
Using SAP Host Agent to Execute Platform LCM Tasks [page 603]
SAP Host Agent Installation
SAP Host Agent Upgrade
SSL Configuration for the SAP Host Agent

10.1.1.1.5.2 Decentralized Execution of Platform LCM Tasks

In some circumstances platform LCM actions must be executed on each individual host of the multiple-host system. This is referred to as **decentralized execution**.

Typically, SAP HANA platform lifecycle management actions, such as update, rename, and inter-service communication configuration, can be performed on a multiple-host system from one host. This is referred to as **centralized execution** and requires SSH or root credentials. For more information, see Centralized Execution of Platform LCM Tasks in Related Information.

In some circumstances, a secure shell (SSH) key may not be installed or root credentials are not available. In this case, the platform LCM actions must be executed on each individual host of the multiple-host system, which is also known as **decentralized execution**. For more information about decentralized execution, see SAP Note 2048681 in Related Information.

Related Information

SAP Note 2048681
Executing Platform LCM Tasks [page 601]
Centralized Execution of Platform LCM Tasks [page 602]

10.1.1.1.6 Additional Information About Using the SAP HANA Platform LCM Tools

If you have already familiarized yourself with the way the SAP HANA database lifecycle manager (HDBLCM) works, you may be interested in additional information like log and trace files, Linux kernel parameter settings, or troubleshooting.

Related Information

Logging [page 608]
Linux Kernel Parameters [page 609]
General Troubleshooting for the SAP HANA Platform LCM Tools [page 611]

10.1.1.1.6.1 Logging

SAP HANA platform lifecycle management processes are logged by the system. The log files are stored in the following path:

`/var/tmp/hdb_<SID>_<action>_<time stamp>`

where `<action> :: = install | update | addhost | uninstall |` and so on.

The following log files are written while performing the action:

- `<hdbcommand>.log`: can be read using a text editor
- `<hdbcommand>.msg`: XML format for display in the installation tool with the GUI
- `<hostname>_tracediff.tgz`: provides a delta analysis of the original trace files, makes a detailed analysis easier

You can also view diagnostic files in the SAP HANA database explorer using the administration function. For more information, see *View Diagnostic Files in the SAP HANA Database Explorer* in the *SAP HANA Administration Guide*.

Instant Logging

If an LCM action crashes or hangs before the execution is finished, even if no LCM action trace is enabled, HDBLCM writes a trace, which has the function of a preliminary (unformatted) log file. Upon program completion, this preliminary logfile is removed and replaced by the real, formatted log file.

The environment variable `HDB_INSTALLER_TRACE_FILE=<file>` enables the trace.

The environment variable `HDBLCM_LOGDIR_COPY=<target directory>` creates a copy of the log directory.

Log Collection

If you perform platfom LCM actions on multiple-host SAP HANA systems, all log files are collected to a local folder to make error analysis more convenient.

To collect log files for multiple-host SAP HANA systems, an HDBLCM action ID is passed to each sub-program (underlying LCM tool) working on a remote host. Each sub-program writes a copy of the log file in to the following directory: `<installation path>/<SID>/HDB<instance number>/<host name>/trace`

Related Information

View Diagnostic Files in the SAP HANA Database Explorer

10.1.1.1.6.2 Linux Kernel Parameters

The following table describes the parameters and limits that are set by the SAP HANA database lifecycle manager (HDBLCM) during the installation or update of an SAP HANA database. The actual values may differ, depending on your system configuration.

> **Note**
>
> The SAP HANA database lifecycle manager (HDBLCM) will only change existing values if they are lower than the default values.

The SAP Host Agent can automatically optimize the following Linux Kernel Parameters:

- `net.ipv4.ip_local_port_range`
- `net.ipv4.ip_local_reserved_ports`

To configure the SAP Host Agent, make sure that the `/etc/sysctl.conf` configuration does not contain any of these two parameters. Afterwards, configure the SAP Host Agent profile parameters as described in *SAP Note 401162*.

Parameter	Description	Value	Location
`nofile`	Open file descriptors per user	`1048576`	`/etc/security/limits.conf`
`fs.file-max`	Open file descriptors per host	`20000000`	`/etc/sysctl.conf`
`fs.aio-max-nr`	Maximum number of asynchronous I/O requests	`18446744073709551615 (= 2^64-1 = ULONG_MAX)`	`/etc/sysctl.conf`
`vm.memory_failure_early_kill`	Method for killing processes when an uncorrected memory error occurs	`1`	`/etc/sysctl.conf`
`kernel.shmmax`	Maximum shared memory segment size (the default minimum value is 1 GB)	`1073741824`	`/etc/sysctl.conf`

Parameter	Description	Value	Location
`kernel.shmmni`	Maximum number of shared memory segments	`32768`	`/etc/sysctl.conf`
`kernel.shmall`	System-wide limit of total shared memory, in 4k pages	• RAM >= 35.5 TB: `(shmmax * shmmni) / 65536` • RAM < 35.5 TB: `(0.9 * <RAM in bytes>) / 4096`	`/etc/sysctl.conf`
`net.ipv4.ip_local_port_range`	Lower limit of ephemeral port range **Note** No changes are applied if the SAP HANA database lifecycle manager (HDBLCM) detects or installs SAP Host Agent version 7.20.162 or greater. The SAP Host Agent takes care of adjusting this parameter and setting it manually is neither recommended nor required. For more information, see SAP Note 2382421	`40000`	`/etc/sysctl.conf`
`vm.max_map_count`	Maximum number of Virtual Memory Areas (VMAs) that a process can own	`2147483647`	`/etc/sysctl.conf`

Related Information

SAP Note 2382421
SAP Note 401162
SAP Note 941735

10.1.1.1.6.3 General Troubleshooting for the SAP HANA Platform LCM Tools

The SAP HANA database lifecycle manager (HDBLCM) is a wrapper tool that calls the underlying HDB tools to perform the platform LCM action. If something unexpected happens when using HDBLCM, and the LCM action cannot be completed, you can check the logs and separately run the affected underlying tools.

> ⚠ Caution
>
> We only recommend the following underlying tools to be used for troubleshooting purposes.

Program Name	Description	Location
hdbinst	Command-line tool for installing the software	Installation media
hdbsetup	Installation tool with a graphical interface for installing or updating the software	Installation media
hdbuninst	Command-line tool for uninstalling the software and removing a host	Installation media and `<installation path>/<SID>/global/hdb/install/bin`
hdbaddhost	Command-line tool for adding a host to a system	`<installation path>/<SID>/global/hdb/install/bin`
hdbupd	Command-line tool for updating the software	Installation media

hdbrename	Command-line tool for renaming a system	`<installation path>/<SID>/global/hdb/install/bin`
		and
		`/usr/sap/<SID>/SYS/global/hdb/install/bin`

Program Name	Description	Location

hdbreg	Command-line tool for registering an SAP HANA system	`<installation path>/<SID>/global/hdb/install/bin`
		and
		`/usr/sap/<SID>/SYS/global/hdb/install/bin`

hdbremovehost	Command-line tool for removing a host	`<installation path>/<SID>/global/hdb/install/bin`
		and
		`/usr/sap/<SID>/SYS/global/hdb/install/bin`

hdbmodify	This command line tool removes and adds remote hosts.
	Furthermore, the listen interface can be changed ('local', 'global', 'internal').
		`<installation path>/<SID>/global/hdb/install/bin`
		and
		`/usr/sap/<SID>/SYS/global/hdb/install/bin`

hdbupdrep	Command-line tool for upgrading a repository by loading delivery units into the database	`<installation path>/<SID>/global/hdb/install/bin`
		and
		`/usr/sap/<SID>/SYS/global/hdb/install/bin`

10.1.1.1.6.4 Managing SAP HANA System Components

SAP HANA system components can be installed, updated, or uninstalled using the SAP HANA database lifecycle manager (HDBLCM).

The SAP HANA system is made up of the following components:

- SAP HANA mandatory components

    - SAP HANA server
    - SAP HANA client
    - Local secure store (LSS)
- SAP HANA additional components
    - SAP HANA studio
    - Application Function Libraries (AFL and the product-specific AFLs IBP, RTL, TRP, VCH, XRP)
    - SAP liveCache applications (SAP LCA or LCAPPS-Plugin)
    - SAP HANA smart data access (SDA)

> **Note**
>
> To install or uninstall the Solution Manager Diagnostics Agent, use Software Provisioning Manager (SWPM). For more information about the setting up the Solution Manager Diagnostics Agent using SWPM, see SAP Note 1858920 in Related Information.

> **Note**
>
> SAP LT replication configuration is a part of SL Toolset 1.0. For more information about configuring SAP LT replication, see SAP Note 1891393 in Related Information.

- SAP HANA options
    - SAP HANA dynamic tiering
    - SAP HANA streaming analytics
    - SAP HANA accelerator for SAP ASE

For more information about installing, updating, and uninstalling the SAP HANA mandatory components and SAP HANA additional components, see the *SAP HANA Server Installation and Update Guide*. For more information about installing, updating, and uninstalling the SAP HANA options, see SAP HANA option documentation in Related Information.

> **Caution**
>
> Be aware that you need additional licenses for SAP HANA options. For more information, see *Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities* in Related Information.

Related Information

SAP Note 1858920
SAP Note 1891393
Important Disclaimer for Features in SAP HANA [page 1677]

10.1.1.1.6.5 Check the Installation Using the Command-Line Interface

You can check the installation of an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface for troubleshooting.

Prerequisites

- You are logged in as root user.
- Any user has read and execute permissions for the directory that contains the installation medium.
- Depending on the storage solution, set the export options `rw`,`no_root_squash` for the installation directory.
- The operating system administrator (`<sid>adm`) user and other operating system users may exist prior to installation. Make sure that you have the passwords of the existing users, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) will not modify the properties of any existing user or group. The `<sid>adm` user passwords must be identical on all hosts.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --action=check_installation
   ```
3. Enter the required credentials.
4. Review the summary, and select *y* to finalize the configuration.

Results

The check tool outputs basic information about the configuration of the file system, system settings, permission settings, and network configuration. The checks are based on the property file stored in the following path:

```
<sapmnt>/<SID>/global/hdb/install/support/hdbcheck.xml
```

Use the generated log files as a reference in the case of troubleshooting. The log file is stored in the following path:

```
/var/tmp/hdb_<SID>_hdblcm_check_installation_<time stamp>/hdblcm.log
```

Related Information

check_installation

10.1.1.2 Users Created During Installation

The following users are automatically created during the installation: `<sid>adm`, `sapadm`, `SYSTEM`, and `<sid>crypt`.

User	Description

`<sid>adm`	The operating system administrator.
	• The user `<sid>adm` is the operating system user required for administrative tasks such as starting and stopping the system.
	• The user identifier (UID) of the `<sid>adm` user is defined during the system installation.
	• The password of the `<sid>adm` user is set during installation with the `password` parameter.
	• If you do not want the operating system user `<sid>adm` and its primary group to be created automatically, you can create it before installation. This might be the case if you use central user management such as Lightweight Directory Access Protocol (LDAP) or Network Information System (NIS). The SAP HANA database lifecycle manager (HDBLCM) will not modify the properties of any existing user or group.
	The following requirements apply:
	  • The name of the user must follow the schema `<sid>adm`. All letters must be lowercase.
	  • The user should have a UID greater than 999.
	  • The primary group of the user must be `sapsys`. The default GID of the `sapsys` group is 79.
	  • The UID of this operating system user and GID of its primary group must be unique and identical on each host of a multiple-host system.

User	Description

`sapadm`	The SAP Host Agent administrator.
	• If there is no SAP Host Agent available on the installation host, it is created during the installation along with the user `sapadm`.
	• If the SAP Host Agent is already available on the installation host, it is not modified by the installer. The `sapadm` user and password are also not modified.
	• The password of the `sapadm` user is set during installation with the `sapadm_password` parameter.
	• If you do not want the user `sapadm` and its primary group to be created automatically, you can create it before installation.
	The following requirements apply:
	  • The primary group of the user must be `sapsys`. The default GID of the `sapsys` group is 79.
	  • The GID of the primary group of the `sapadm` user must be unique and identical on each host of a multiple-host system.

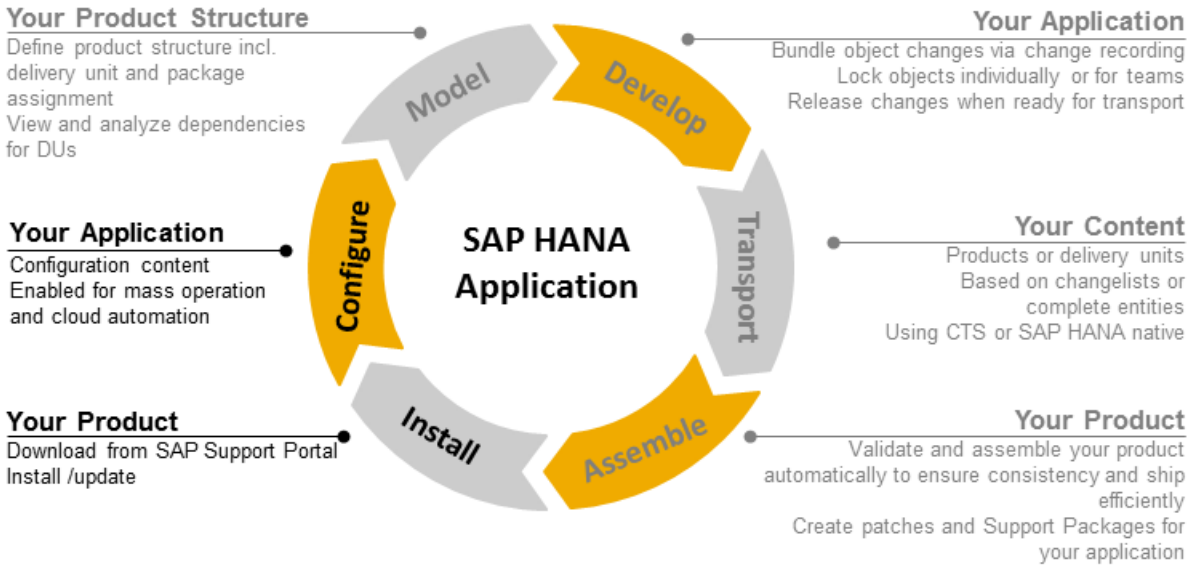
SYSTEM	The database superuser. • Initially, the SYSTEM user has all system permissions. Additional permissions can be granted and revoked again, however the initial permissions can never be revoked. • Two SYSTEM users area created: one for the system database and one for the tenant database. • The password of the SYSTEM user is set during installation with the `system_user_password` parameter.
`<sid>crypt`	The trusted local secure store (LSS) user. • The user `<sid>crypt` owns the storage of the encryption keys and other similarly sensitive data. • The user `<sid>crypt` is the only trusted user of the local secure store. Only processes called by a trusted user are accepted by the LSS right away.

Related Information

Predefined Database Users
Operating System User <sid>adm [page 552]
Predefined XS Advanced Users

10.2 SAP HANA Application Lifecycle Management

SAP HANA Application Lifecycle Management supports you in all phases of an SAP HANA application lifecycle, from modelling your product structure, through application development, transport, assemble, and install.



Phases of SAP HANA Application Lifecycle Management

The following are phases of SAP HANA application lifecycle management. Some phases are designed for developers only, while others, such as the installation of add-on products and software components, are designed for both.

- **Model**
  You define your product structure to provide a framework for efficient software development. This includes creating the following metadata: creating Repository packages for development, defining a package hierarchy and assigning packages to delivery units. The delivery units are then bundled in products.
- **Develop**
  You perform software developments in Repository packages. SAP HANA application lifecycle management supports you with change tracking functions which allow you to transport only changed objects.
- **Transport**
  You can transport your developed content in different ways according to your needs. You can choose between transporting products or delivery units, based on changelists or complete entities. The transport type can be native SAP HANA transport or transport using Change and Transport System (CTS). You can also export delivery units, and import them into another system.
- **Assemble**
  The developed software plus the metadata defined when modelling your product structure as well as possible translation delivery units are the basis for assembling your add-on product. You can also build Support Packages and patches for your product.

- **Install**
  You can install add-on products or software components that you downloaded from SAP Support Portal or assembled yourself.

All tasks related to the **Install** and **Configure** phases of SAP HANA application lifecycle management are documented in this *SAP HANA Administration Guide*. The tasks related to software development are documented in the *SAP HANA Developer Guide (For SAP HANA Studio)*. **All** phases of SAP HANA application lifecycle management are documented in the *SAP HANA Application Lifecycle Management Guide*.

Related Information

Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]

10.2.1 Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model

SAP HANA application lifecycle management provides functions for installing and updating SAP HANA products or individual software components of SAP HANA XS classic model that you have downloaded from the SAP Support Portal, or that you have assembled yourself.

Context

SAP HANA products consist of software components which are deployed to the SAP HANA repository. You have the following options to install and update SAP HANA products and software components:

- Using a **SAP Fiori application** integrated in the **SAP HANA Application Lifecycle Management XS application**. This application can be started in the following ways:
  - Start the SAP HANA Application Lifecycle Management on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`. Afterwards, choose the *INSTALLATION* tab or tile.
  - Using a link in SAP HANA Web-based Development Workbench.
    In the SAP HANA Web-based Development Workbench Editor tool, choose *Navigation Links* > *Lifecycle Management*. The SAP HANA Application Lifecycle Management home screen opens, where you can choose the *INSTALLATION* tab or tile..
  - Using the context menu in SAP HANA studio.
    Choose *Lifecycle Management* > *Application Lifecycle Management* > *Installation* from the context menu for a particular system in the *SAP HANA Administration Console* perspective in SAP HANA studio.

  The documentation about using SAP HANA Application Lifecycle Management to install and update SAP HANA products and software components describes the following use cases:

- *Installing and Updating SAP HANA Products*
- *Installing and Updating SAP HANA Software Components*

- Using the `hdbalm` **commandline tool**.
  To start `hdbalm`, start a command line client and navigate to the directory where `hdbalm` is located. You can also add this directory to your path.
  For more information about using `hdbalm` to install and update SAP HANA products and software components, see the following topics in the *SAP HANA Application Lifecycle Management Guide*:
  - *Using hdbalm*
  - *hdbalm install Command*

> ⓘ Note
>
> SAP HANA system components like the SAP HANA client, SAP HANA studio, and additional system components like Application Function Libraries (AFL and the product-specific AFLs POS, SAL, SCA, SOP, UDF), SAP liveCache applications (SAP LCA or LCAPPS-Plugin), XS advanced runtime applications, or SAP HANA smart data access (SDA) are installed and updated using the SAP HANA database lifecycle manager (HDBLCM). For more information, refer to the *SAP HANA Server Installation and Update Guide*.

Related Information

Installing and Updating SAP HANA Products [page 619]
Installing and Updating SAP HANA Software Components [page 622]
Installation and Update Options [page 624]
Using hdbalm
hdbalm install Command
SAP HANA Server Installation and Update Guide

10.2.1.1 Installing and Updating SAP HANA Products

You can install and update SAP HANA products using SAP HANA application lifecycle management.

Prerequisites

- You have a product archive of an SAP HANA product that you want to install or update, or you have a directory location that contains the extracted product archive files.

  > ⓘ Note
  >
  > An SAP HANA product archive is a `*.zip` file that contains one or more software component archives as well as metadata files. For more information about the archive types that are used to deliver SAP HANA content, read the information about *SAP HANA content* in the *SAP HANA Administration Guide*.

- You have the privileges granted by a role based on the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role template.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

   The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.
2. Choose the *INSTALLATION* tab.
3. Click in the *Archives* selection field to select a product archive, or the extracted product archive files, from your file directory that you want to install or update.

   The files of the product archive are uploaded. The header area contains information about the product version, including the action that is to be performed: Installation or Update.
4. The *Product Instances* tab lists all product instances that are part of the archive.

   For each product instance, the result of a software component check is displayed.

   The following results can occur:

   - *OK*
     The product instance can be installed.
   - *Downgrade*
     The product instance contains one or more software components that are already installed in newer versions than the ones to be installed. The installation of this product instance would lead to a downgrade of these software components. Downgrades are not allowed. To continue the installation of the product, you have to set the installation/update option *Keep newer version of software component*. In this case, the downgrading software components will skipped during the installation of the product instance.

     > **Note**
     >
     > If it is required that you install the software component that causes the downgrade, for example, if the newer version has errors and you want to revert to the previous version, you can use the `install` command of `hdbalm` with the option `ALLOW_DU_DOWNGRADE` to enable the downgrade. However, use this option with care, since this may affect other installed products which require the newer version of this software component.

   - *Some software components are installed already*
     If software components are already installed in the same version, by default, the system skips their installation during the installation/update of the product instance. If you want to reinstall the same version, you can set the option *Overwrite the same version of software component* in the installation and update options.

   Click in the line of the product instance to display more information about the software components that are part of the product instance. For each software component, a status is displayed, as well as the installed version and the new version. If you click on the status icon, you get more information about the status.
5. If required, set installation or update options.

The options allow you to override the default behavior of the installation or update for specific situations. Use them with care. For more information about the options, see *Installation and Update Options*.

6. Select product instances for installation.

   You can individually select single product instances. To install all instances, select the *Instance* check box in the header row.

   All instances of the product, that are already installed on your system will automatically be checked for updates. If the archive that you uploaded contains newer versions for one or several software components, they will automatically be updated. It doesn't matter whether you selected the respective instance for installation.

7. To start the installation, choose *Install*.

   The system displays the progress of the individual installation steps. You can click on each step to expand the log of the step.

Results

If errors occur during the installation or update, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA software component or product instance that caused the error. The *Support Information* tab contains the relevant information. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

If the installation or update finished successfully, you can start another installation using *New Installation*.

Related Information

Installation and Update Options [page 624]
Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]
SAP HANA Administration Guide [page 8]

10.2.1.2 Installing and Updating SAP HANA Software Components

You can install and update SAP HANA software components using SAP HANA application lifecycle management.

Prerequisites

- You have one or multiple archives of SAP HANA software components that you want to install or update.

  > ⓘ Note
  >
  > An SAP HANA software component archive is a `*.zip` file that contains one delivery unit archive (`*.tgz`) as well as metadata files. For more information about the archive types that are used to deliver SAP HANA content, read the information about *SAP HANA content* in this *SAP HANA Administration Guide*.
  >
  > Software components which need to be installed at the operating system level, such as Application Function Libraries (AFLs), are **not** installed using SAP HANA application lifecycle management.

- You have the privileges granted by the SAP HANA Application Lifecycle Management `sap.hana.xs.lm.roles::Administrator` role.

Procedure

1. Open the SAP HANA Application Lifecycle Management.

   The SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`.
2. Choose the *INSTALLATION* tab.
3. Click in the *Archives* selection field to select one or multiple software component archives from your file directory that you want to install or update.

   The software component archives are uploaded. For each software component the following information is displayed:
   - *Status*
     The following status values exist:
     - *New*
       The software component is not yet installed and can be installed.
     - *Update*
       The software component is already installed and can be updated to a higher version.
     - *Downgrade*
       The software component is already installed in a newer version than the one that is to be installed. The installation of this software component would lead to a downgrade. Downgrades are not allowed. You cannot continue to install the software component.

> **Note**
>
> If it is required that you install the software component that causes the downgrade, you can use the `install` command of `hdbalm` with the option `ALLOW_DU_DOWNGRADE` to enable the downgrade. However, use this option with care, since this may affect already installed products which require the newer version of this software component.

- *Already installed*
  If software components are already installed in the same version, by default, the system skips their installation during the installation/update. If you want to reinstall the same version, you can set the option *Overwrite the same version of software component* in the installation and update options.

- *Version* that is already *installed* in the system
- *New version* that is to be installed
- Whether the software component is part of a product that is already installed.
- *Information* relevant for the *support*

4. If required, set installation or update options.

   The options allow you to override the default behavior of the installation or update for specific situations. Use them with care. For more information about the options, see *Installation and Update Options*.

5. To start the installation, choose *Install*.

   The system displays the progress of the individual installation steps. You can click on each step to expand the log of the step.

Results

If errors occur during the installation or update, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA software component that caused the error. You can find this information in the support information of the component. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

If the installation or update finished successfully, you can start another installation using *New Installation*.

Related Information

Installation and Update Options [page 624]
Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]
SAP HANA Content [page 661]

10.2.1.3 Installation and Update Options

Installation and update options are available that allow you to influence the installation and update behavior, if required.

Installation and Update Options

Option	Corresponding Installation Option in hdbalm	Description
Overwrite the same version of software component	ALLOW_DU_SAME_VERSION	By default, the system does not install a software component if the same version is already installed. It is possible to override this behavior in the following situations: • If a previous installation operation failed, for example, because of activation errors. • If you run continuous integration scenarios in which the same version of a software component is installed regularly.
Keep newer version of software component	ALLOW_KEEP_DU_NEWER_VERSION	If the product instance to be installed contains software components with lower versions than the installed ones, installing the software component in the lower version would lead to a downgrade of this software component. This is not allowed. You cannot install a downgrading software component. If you want to install the product instance without the downgrading software components, you can use this option.This option is useful if a software component is part of several products. If the product to be installed contains the software component in a version which is lower than the one already installed, you can choose to retain the newer version. In this case, the installation of the software component is skipped.
Allow version update	ALLOW_DU_VERSION_UPDATE	Allows version updates of software components. In some cases, for example, if a software component is part of several products, a version update of a software component could render one product inoperable. If the system detects an inconsistency, it aborts the operation. You can use this option to turn off this behavior.

Option	Corresponding Installation Option in hdbalm	Description
Roll back installation if activation errors occur (default)	This is the default behavior in `hdbalm`.	By default, the installation is canceled if any activation errors occur and the complete installation is rolled back. Installation is also rolled back if you modified objects in your system and a modified object cannot be activated because it references an object that is part of the installation archive. This can occur, for example, if a procedure or view references a table in the archive.
Ignore activation errors of referencing objects	USE_TWO_COMMIT_ACTIVATION	If an installation fails since an object outside of the archive cannot be activated due to references to an object in the archive, you can repeat the installation with this activation option. In this case, the object remains broken in the system after the installation, but the installation itself finishes successfully. You must correct the errors manually after the installation. You can check the transport log after performing the installation without this option to find out whether the activation errors were caused by objects in the archive or outside of the archive. After repeating the installation with this option, check the transport log to find out which objects must be repaired afterwards.

Related Information

hdbalm install Command

Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]

10.2.2 Installing and Updating Products and Software Components in SAP HANA XS Advanced Model

Application lifecycle management for SAP HANA XS advanced model provides functions for installing and updating products as well as individual software components of SAP HANA XS advanced that you have downloaded from the SAP Support Portal.

Prerequisites

- The prerequisites described under *Prerequisites and Authorizations* are fulfilled. The link to the topic is in the *Related Information*.
- You have an SAP HANA XS advanced product or software component archive that you want to install or update.

> ⓘ Note
>
> An SAP HANA XS advanced software component archive is a `*.zip` file that consists of a multitarget application archive (MTA archive = `*.mtar` file) and an `SL_MANIFEST.xml` file that contains metadata, such as version, vendor, support package, and patch level for the MTA archive.
>
> An SAP HANA XS advanced product archive is a `*.zip` file that consists of one or multiple software component archives plus a `pd.xml` and a `stack.xml` file. Both files contain metadata for the product, such as version, support package level, and vendor.

Context

> ⓘ Note
>
> From SPS 11, SAP HANA includes an additional run-time environment for application development: SAP HANA extended application services (XS), advanced model. SAP HANA XS advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of SAP HANA extended application services (XS), classic model. SAP recommends that customers and partners who want to develop new applications use SAP HANA XS advanced model. If you want to migrate existing XS classic applications to run in the new XS advanced run-time environment, SAP recommends that you first check the features available with the installed version of XS advanced; if the XS advanced features match the requirements of the XS classic application you want to migrate, then you can start the migration process.

You have the following options to install and update SAP HANA products and software components in SAP HANA XS advanced:

- Using the XS advanced command line interface (CLI)
- Using the XS Advanced Application Lifecycle Management graphical user interface

The links to the corresponding topics are in the *Related Information*.

> **Note**
>
> SAP HANA system components like the SAP HANA client, SAP HANA studio, and additional system components like Application Function Libraries (AFL and the product-specific AFLs POS, SAL, SCA, SOP, UDF), SAP liveCache applications (SAP LCA or LCAPPS-Plugin), XS advanced runtime applications, or SAP HANA smart data access (SDA) are installed and updated using the SAP HANA database lifecycle manager (HDBLCM). For more information, refer to the *SAP HANA Server Installation and Update Guide*.

Related Information

Prerequisites and Authorizations [page 627]
Installing and Updating Using the Command Line Interface [page 628]
Installing and Updating Using the XS Advanced Application Lifecycle Management Graphical User Interface [page 647]
SAP HANA Server Installation and Update Guide

10.2.2.1 Prerequisites and Authorizations

The following prerequisites have to be fulfilled when you use functions required for installing and updating SAP HANA products and software components in SAP HANA XS advanced model.

- The XS advanced run time is installed and available on the SAP HANA server.
  For more information, see *Installing XS Advanced Runtime* in the *SAP HANA Server Installation and Update Guide*.
- When using the XS Advanced Application Lifecycle Management Graphical User Interface, the following software components are installed in addition to the XS advanced run time:
  - SAP UI5 component: `XSACUI5FESV344P` in version 1.44.8 or higher
  - SAP HANA XS Advanced Application Lifecycle Management Product Installer UI: `XSAC_ALM_PRODUCT_INSTALLER_UI1`

  You can download the components from the Software Download Center at https://support.sap.com/swdc and install them using the `xs install` command in the command line interface.
  For more information, see the *SAP HANA Server Installation and Update Guide*.

  > **Note**
  >
  > If you have performed the default SAP HANA medium installation, both the SAP UI5 component and the SAP HANA XS Advanced Application Lifecycle Management Product Installer UI are already installed.
- Optional when using the XS advanced command line interface (XS CLI) for installation and update: The XS advanced command line client is installed on your local machine.
  The XS CLI client tools are installed by default on the SAP HANA server. You can log on to the server and execute the installation command there. However, if you want to connect to SAP HANA from your local machine, you must download and install the client tools locally. The XS CLI client tools (`xs.onpremise.runtime.client_<platform>-<version>.zip`) can be downloaded from the SAP HANA server, from the installation DVD, or from the SAP support portal.

- The SAP HANA database user that is used to perform the installation or update has one of the following permissions assigned:
  - The user has the `XS_CONTROLLER_USER` parameter assigned as well as the *SpaceDeveloper* role for each space in which the user wants to perform an installation or update.
  - The user has the `XS_CONTROLLER_ADMIN` parameter assigned.
    This scope allows the installation in all spaces.

  For more information on assigning roles in SAP HANA XS advanced, see *Setting Up Security Artifacts* in the *SAP HANA Administration Guide*.

Related Information

SAP HANA Administration Guide [page 8]
SAP HANA Server Installation and Update Guide
Installing an SAP HANA System Including the XS Advanced Runtime

10.2.2.2 Installing and Updating Using the Command Line Interface

To install and update products and software components in SAP HANA XS advanced, the `xs install` command is available in the XS advanced command line interface (CLI). Using this command you can install or update one product archive or one software component archive at a time.

Procedure

1. Start the XS advanced command line interface (CLI).
2. Log on to the SAP HANA XS advanced runtime in the organization and space in which you want to install or update the product or software component.

   To do this, use the `xs login` command with the following arguments and options:

Argument/Option	Description
`-u`	SAP HANA database user with the permissions as described in the *Prerequisites* section
`-p`	password
`-o`	organization in which the installation or update takes place

Argument/Option	Description
`-s`	space in which the installation or update takes place

Sample Code

```
xs login - u demo -p test -o demoorg -s demospace
```

For more information, see the *XS CLI: Logon and Setup* topic in *SAP HANA Developer Guide (for SAP HANA XS Advanced Model)*.

3. If you want that the archives to be installed are checked with antivirus software before the installation or update process, proceed as described under *Set Up a Virus Scan for Installation Archives*. The link to the topic is in the *Related Information* section.
4. Start the installation or update of the product or software component.

   The `xs install` command is available in the XS advanced CLI both for installing product and software component archives in XS advanced and updating these. The `xs install` command detects whether the archive is a product archive or a software component archive. It also detects whether the product or software component is installed already and subsequently executes either an installation or update operation.

   Enter the `xs install` command and specify the path to the archive. If required, enter any additional options. For example, to install a specific instance of a product, you can use the `-i` option and specify the product instance. Or to make sure that the entity you are about to install is a product, you can add the `-pv` option. In this case, the installation is only performed if you specify a product archive for the `xs install` command. If you specify a software component archive, the installation is not performed.

   Sample Code

   ```
   xs install .../sap_demo/target/XSASAMPLEPRODUCT1.0.zip
   ```

   Instead of `xs install` you can also use the `xs ins` alias. For more information on the options, see *Installation and Update Options in XS Advanced Model*. For installation examples, see *Examples: Installing and Updating Products and Software Components in XS Advanced Model*. The links are in the *Related Information* section.

Results

Before installing or updating the product or software component, the system performs different checks. If no errors are found, the system performs the installation or update with the arguments and options that you specified. During the process, the product installer calls the deploy service that performs the actual deployment. Afterwards, the product installer registers the product or software component as installed.

If the installation or update cannot be performed, it is possible, in some situations, to use additional options to override the default behavior of the system. For more information, see *Checks Before Installing or Updating Products or Software Components in XS Advanced Model* and *Installation and Update Options in XS Advanced Model*.

If errors occur during the installation or update, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA product or software component that caused the error. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

To display the correct log file, use one of the following commands with the process ID that you find in the result of the installation or update process.

- To display the log of a product installation, use the `display-installation-logs` command with the `--pv` option.

  ```
  xs display-installation-logs <process ID> --pv
  ```

- To display the log of a software component installation, use the `display-installation-logs` command with the `--scv` option.

  ```
  xs display-installation-logs <process ID> --scv
  ```

To display the history of installation or uninstallation processes, you can use the `display-installation-history` command.

```
xs display-installation-history
```

For more information on the commands used for installation, use `xs help <command>` in the XS advanced CLI, or see *The XS Command-Line Interface Reference* section in the *SAP HANA Developer Guide for XS Advanced Model*.

Related Information

Installation and Update Options in XS Advanced Model [page 634]
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
The XS Command-Line Interface Reference
XS CLI: Logon and Setup
XS CLI: Plug-ins
Examples: Installing and Updating Products and Software Components in XS Advanced Model [page 637]
Checks Before Installing or Updating Products or Software Components in SAP HANA XS Advanced Model [page 632]
Set Up a Virus Scan for Installation Archives [page 631]

10.2.2.2.1 Set Up a Virus Scan for Installation Archives

You can set an environment variable in your system to enable a default virus scan for all software component archives that you want to install or update.

Prerequisites

You have installed and configured the SAP virus scan interface as described in SAP Note 786179.

Context

If the antivirus software that you are using does not check the software component archives that you want to install or update, you can use the SAP virus scan interface and set the environment variable `SCAN_UPLOADS` to the value `true`. This way, the system checks all archives that you want to install or update.

By default, no antivirus protection is set for the product installer.

Procedure

1. In the commandline tool, set the XS advanced environment variable `SCAN_UPLOADS` to **`true`**.

   Sample Code

   ```
   xs set-env product-installer SCAN_UPLOADS true
   ```

   For more information about setting environment variables in XS advanced, see *XS CLI: Application Management* in the *SAP HANA Developer Guide For SAP HANA XS Advanced Model*.
2. Restart the product installer.

   The restart is required to ensure that the change to the environment variable takes effect.

   Sample Code

   ```
   xs restart product-installer
   ```

   For more information about restarting applications in XS advanced, see *XS CLI: Application Management* in the *SAP HANA Developer Guide For SAP HANA XS Advanced Model*.

Related Information

Installing and Updating Products and Software Components in SAP HANA XS Advanced Model [page 626]

SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
XS CLI: Application Management

10.2.2.2.2 Checks Before Installing or Updating Products or Software Components in SAP HANA XS Advanced Model

To ensure consistency of SAP HANA products, the system executes different checks before installing or updating a product or a software component in SAP HANA XS advanced.

Product installations only: Check whether the product to be installed is already installed and in which version

If the product to be installed is not yet installed, the installation will be performed. If it is already installed, the system checks the installed version. If it is already installed in the same version, or in a lower support package level, the installation or update will be performed.

- Product is already installed in higher version
  If the version of the product to be installed is lower than the installed version, the system terminates the process because installing the lower version would lead to a downgrade of the product.
  You can override this behavior and allow a downgrade of the product. To do this, you can use the `ALLOW_PV_DOWNGRADE` option with the `xs install` command.
- Product is already installed in lower version
  If the version of the product to be installed is higher than the installed version, the system updates the installed version automatically.

> ⓘ Note
>
> The version of a product usually consists of one or more numbers in an ascending order. In addition to the version number, a support package level is provided for the product.

> ✤ Example
>
> The version number is `1.0`. In this case, the following versions are considered version updates: `1.1`, `2.0`, or `2`.

Check whether the software component is already installed and in which version

If the software component to be installed was not yet installed, the installation will be performed. If it was already installed, the system checks the installed version. If it is installed in a lower support package or patch level, the update will be performed.

> **Note**
>
> The version of a software component has the following form: "`#.#.#`", for example `1.0.3`, where
>
> - `1` = the version
> - `0` = the support package level
> - `3` = the patch level

- Software component is already installed in higher version
  If a version of a software component to be installed is lower than an installed version, the system terminates the installation.
  You have the following options to override this behavior:
  - You can allow a downgrade of the software component. To do this, use the `ALLOW_SC_DOWNGRADE` option.
  - For product installation only: You can skip the installation of all software components that are part of the archive and that are already installed in higher versions. To do this, use the `ALLOW_KEEP_SC_NEWER_VERSION` option.
- Software component is already installed in same version
  If a version of a software component to be installed is the same as the installed version, the system proceeds as follows:
  - Product installation: The system does not install this software component. The installation of this software component is skipped during the installation of the product.
  - Software component installation: The system terminates the installation.

  You can override this behavior and allow the reinstallation of the same version. To do this, use the `ALLOW_SC_SAME_VERSION` option for this software component.

  > **Note**
  >
  > If the software component is installed in the system in the same version with the status *BROKEN*, it is automatically reinstalled.
- Software component is already installed in lower version
  If a version of a software component to be installed is higher than an installed version, the system updates the installed version automatically.

Check for dependencies on SAP HANA platform components or other XS advanced components

If the software component has dependencies on SAP HANA platform components or other XS advanced components that are not installed, the system terminates the process and displays the missing software components. You must install or update the missing software components before you can restart the current installation or update.

For more information on the options to override the default behavior, see *Installation and Update Options in XS Advanced Model*. The link is in the *Related Information* section.

Check whether extension descriptor is valid, if an extension descriptor is used

If an extension descriptor is used for the installation process, the system checks that the extension descriptor file does not exceed a specific file size and that the syntax of the extension descriptor file is correct. If the file is too big or if the syntax is incorrect, the system will not start the installation process.

For more information on extension descriptors, see the *The MTA Deployment Extension Description* topic in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*.

Related Information

Installing and Updating Products and Software Components in SAP HANA XS Advanced Model [page 626]
Installation and Update Options in XS Advanced Model [page 634]
Display installed Products and Software Components in XS Advanced Model [page 642]
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
The MTA Deployment Extension Descriptor

10.2.2.2.3 Installation and Update Options in XS Advanced Model

Installation and update options are available in SAP HANA XS advanced that allow you to influence the installation and update behavior, if required.

The following is the default syntax for the `xs install` command in the XS advanced CLI:

```
xs install <ARCHIVE> [-p <TARGET_PLATFORM>] [-pv
| --PRODUCT_VERSION] [-scv | --SOFTWARE_COMPONENT_VERSION]
[-t <TIMEOUT>] [-e <EXT_DESCRIPTOR_1>[,<EXT_DESCRIPTOR_2>]] [-
o <VERSION_OPTION_1>[,<VERSION_OPTION_2>]] [-i | --INSTANCES
<INSTANCE_1[,INSTANCE_2]>] [--delete-services] [--delete-service-brokers] [--no-
start] [--ignore-lock]
```

The following is an example of a product installation:

```
xs install /sap_demo/target/XSASAMPLEPRODUCT1.0.zip -pv -o ALLOW_SC_SAME_VERSION
```

For more installation examples, see *Examples: Installing and Updating Products and Software Components in XS Advanced Model*. The link is in the *Related Information* section.

Installation and Update Arguments

Argument	Description
<ARCHIVE>	The path to (and name of) the archive containing the product or software component (SCV) to install, update, or downgrade

Installation and Update Options

Option	Description
`-p <TARGET_PLATFORM>`	Specify the target platform where the product or software component will be installed. If not specified explicitly, a target platform is created implicitly as '`<ORG> <SPACE>`'.
`-pv | --PRODUCT_VERSION`	Install a product. The installation is performed only if the given archive is a product archive. Otherwise, the installation will fail.
`-scv | --SOFTWARE_COMPONENT_VERSION`	Install a software component. The installation is performed only if the given archive is a software component archive. Otherwise, the installation will fail.
`-e <EXT_DESCRIPTOR_1>[,<EXT_DESCRIPTOR_2>]`	Define one or more extensions to the installation/deployment descriptors; multiple extension descriptors must be separated by commas. For more information on extension descriptors, see the *The MTA Deployment Extension Description* topic in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model* and *The Multitarget Application Model* guide.
`-t <TIMEOUT>`	Specify the maximum amount of time (in seconds) that the installation service must wait for the installation operation to complete

Option	Description
`-o <VERSION_OPTION_1>[,<VERSION_OPTION_2>]`	Specify options which can be used to **override** the default behavior of the `install` command. The following options are available: • ALLOW_PV_DOWNGRADE Allows a downgrade of the product. By default, the system does not install a product if the product is already installed in a higher product version or support package stack since this would lead to a downgrade of the product. It is possible to override this behavior, for example, if the newer version has errors and you want to revert to the previous version. This option is available for product installations only. • ALLOW_KEEP_SC_NEWER_VERSION Skips the installation of a software component if a newer version is already installed in the system. By default, the system does not install a product if a newer version of one of the software components contained in the product archive is already installed. It is possible to override this behavior. This option is useful, for example, if a software component is part of several products. If the product to be installed contains the software component in a lower version than the one already installed, you can choose to retain the newer version. If you use this option, the installation of this software component is skipped. This option is available for product installations only. • ALLOW_SC_DOWNGRADE Allows a downgrade of the software component. By default, the system does not install a software component if this leads to a downgrade of the software component. It is possible to override this behavior, for example, if the newer version has errors and you want to revert to the previous version. **⚠ Caution** Use this option carefully. • ALLOW_SC_SAME_VERSION Reinstalls the same version of the software component. By default, the system does not install a software component, if the same version is already installed. It is possible to override this behavior, for example, if you run continuous integration scenarios in which the same version of a software component is installed regularly.
`-i, --INSTANCES <INSTANCE_1>[,<INSTANCE_2>]`	By default all instances are installed; a comma-separated list of instances can be specified to limit the number of instances installed
`--delete-services`	Recreate changed services and/or delete discontinued services
`--delete-service-brokers`	Delete discontinued service brokers

Option	Description
`--no-start`	Do not start applications that are updated during the installation
`--ignore-lock`	Force installation even if the space targeted for installation is locked
`--deploy-passthrough <{"<key>" : "<value>", ...}>`	Option for the deploy service

See also the `xs install` command reference in the *XS CLI: Plugins* topic in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*.

Related Information

Installing and Updating Products and Software Components in SAP HANA XS Advanced Model [page 626]
Examples: Installing and Updating Products and Software Components in XS Advanced Model [page 637]
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
The MTA Deployment Extension Descriptor
The Multitarget Application Model
XS CLI: Plug-ins

10.2.2.2.4 Examples: Installing and Updating Products and Software Components in XS Advanced Model

The examples show how you can use the `xs install` command.

In the following examples you must be logged on to the XS command line interface (XS CLI) with a user with the authorizations required for installation and in the organization and space in which you want to perform the installation.

Installation of New Product

The following example installs the product *XSA Sample Product* in version `1.0, SPS 0 (initial shipment stack)` contained in the file `XSASAMPLEPRODUCT_1.0.zip`:

```
XSA Sample Product (sap.com) 1.0 SPS 0
Product Instance 1
SCV_A 1.0.0
SCV_B 1.0.0
```

No version of this product is currently installed. The following command is used:

```
xs install XSASAMPLEPRODUCT_1.0.zip
```

After the installation, the `xs list-products` command displays the product as follows:

```
name                 vendor    version   SPS   instance ids
-------------------------------------------------------------
XSA Sample Product   sap.com   1.0       0     1
```

The detail display for *XSA Sample Product* looks as follows:

```
xs list-products "XSA Sample Product"
---------------------------------
name                    XSA Sample Product
vendor                  sap.com
version                 1.0
support package stack   0
---------------------------------
instance id    software component     version   state
-----------------------------------------------------
1              -                      -         INSTALLED
               SCV_A                  1.0.0     INSTALLED
               SCV_B                  1.0.0     INSTALLED
-----------------------------------------------------
```

Update with Support Package Stack

The following example installs the product *XSA Sample Product* in version `1.0, SPS 5` contained in the file `XSASAMPLEPRODUCT_1.0.5.zip` in the system:

```
XSA Sample Product (sap.com) 1.0 SPS 5
Product Instance 1
SCV_A 1.5.0
SCV_B 1.5.0
```

Version `1.0, SPS 0` (initial shipment stack) of *XSA Sample Product* containing software components `SCV_A` in version `1.0.0` and `SCV_B` in version `1.0.0` is currently installed. To ensure that the archive to be installed is a product archive, the `-pv` option is used.

```
xs install XSASAMPLEPRODUCT_1.0.5.zip -pv
```

After the update, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
---------------------------------
name                    XSA Sample Product
vendor                  sap.com
version                 1.0
support package stack   5
---------------------------------
instance id    software component     version   state
-----------------------------------------------------
1              -                      -         INSTALLED
               SCV_A                  1.5.0     INSTALLED
               SCV_B                  1.5.0     INSTALLED
-----------------------------------------------------
```

Installation of Lower Support Package Version

The following example installs the product *XSA Sample Product* in version `1.0, SPS 3` contained in the file `XSASAMPLEPRODUCT_1.0.3.zip` in the system:

```
XSA Sample Product (sap.com) 1.0 SPS 3
Product Instance 1
SCV_A 1.3.0
SCV_B 1.3.0
```

Version `1.0, SPS 5` of *XSA Sample Product* containing software components `SCV_A` in version `1.5.0` and `SCV_B` in version `1.5.0` is currently installed. If the installation was started without any options, it would fail. To allow the downgrade of the support package version, you must use the `ALLOW_PV_DOWNGRADE` option. In addition, to allow a downgrade of the software components, you must use the `ALLOW_SC_DOWNGRADE` option.

```
xs install XSASAMPLEPRODUCT_1.0.3.zip -o ALLOW_PV_DOWNGRADE, ALLOW_SC_DOWNGRADE
```

After the installation, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
---------------------------------
name                    XSA Sample Product
vendor                  sap.com
version                 1.0
support package stack   3
---------------------------------
instance id    software component     version   state
-----------------------------------------------------
1              -                      -         INSTALLED
               SCV_A                  1.3.0     INSTALLED
               SCV_B                  1.3.0     INSTALLED
-----------------------------------------------------
```

Installation of Higher Product Version

The following example installs the product *XSA Sample Product* in version `2.0, SPS 1` contained in the file `XSASAMPLEPRODUCT_2.0.1.zip` in the system:

```
XSA Sample Product (sap.com) 2.0 SPS 1
Product Instance 1
SCV_A 2.1.0
SCV_B 2.1.0
```

Version `1.0, SPS 3` of *XSA Sample Product* containing software components `SCV_A` in version `1.3.0` and `SCV_B` in version `1.3.0` is currently installed. The installation of version `2.0, SPS 1` of the *XSA Sample Product* updates both the product version and the software component versions automatically.

```
xs install XSASAMPLEPRODUCT_2.0.1.zip -pv
```

After the installation, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
---------------------------------
name                    XSA Sample Product
```

```
vendor                  sap.com
version                 2.0
support package stack   1
--------------------------------
instance id    software component     version   state
-------------------------------------------------------
1              -                      -         INSTALLED
               SCV_A                  2.1.0     INSTALLED
               SCV_B                  2.1.0     INSTALLED
-------------------------------------------------------
```

Installation of Lower Product Version

The following example installs the product *XSA Sample Product* in version `1.5, SPS 3` contained in the file `XSASAMPLEPRODUCT_1.5.3.zip` in the system:

```
XSA Sample Product (sap.com) 1.5 SPS 3
Product Instance 1
SCV_A 1.3.5
SCV_B 1.3.5
```

Version `2.0, SPS 1` of *XSA Sample Product* containing software components `SCV_A` in version `2.1.0` and `SCV_B` in version `2.1.0` is currently installed. To allow a downgrade of the product version, you must use the `ALLOW_PV_DOWNGRADE` option with the command. In addition, to allow a downgrade of the software components, you must use the `ALLOW_SC_DOWNGRADE` option.

```
xs install XSASAMPLEPRODUCT_1.5.3.zip -o ALLOW_PV_DOWNGRADE, ALLOW_SC_DOWNGRADE
```

After the installation, the `xs list-products` command displays the product as follows:

```
xs list-products "XSA Sample Product"
--------------------------------
name                    XSA Sample Product
vendor                  sap.com
version                 1.5
support package stack   3
--------------------------------
instance id    software component     version   state
-------------------------------------------------------
1              -                      -         INSTALLED
               SCV_A                  1.3.5     INSTALLED
               SCV_B                  1.3.5     INSTALLED
-------------------------------------------------------
```

Installation of Software Component

The following example installs the software component *SCV_A* in version `1.2.3` contained in the file `SCV_A_123.zip`. No version of this software component is currently installed. The `-scv` option is used to make sure that the archive to be installed is a software component archive.

```
xs install SCV_A_123.zip -scv
```

After the installation, the `xs list-components` command displays the software component as follows:

```
xs list-components
software component                version
----------------------------------------
SCV_A (sap.com)                    1.2.3
```

Installation of Product with Lower Version of Software Component

The following example installs the product *XSA Test Product* in version `1.0, SPS 3` contained in the file `XSATESTPRODUCT_1.0.3.zip`. No version of the product is currently installed. However, the product contains the software component *SCV_A* in version `1.0.3` which was already installed individually in version `1.2.3`.

You have the following options to proceed with the installation:

- To allow a downgrade of the software component, you can use the `ALLOW_SC_DOWNGRADE` option with the command.

  ```
  xs install XSATESTPRODUCT_1.0.3.zip -o ALLOW_SC_DOWNGRADE
  ```

  After the installation, the `xs list-components` command displays the software component as follows:

  ```
  xs list-components
  software component                version
  ----------------------------------------
  SCV_A (sap.com)                    1.0.3
  ```

- To keep the newer version of the software component, you can use the `ALLOW_KEEP_SC_NEWER_VERSION` option with the command.

  ```
  xs install XSATESTPRODUCT_1.0.3.zip -o ALLOW_KEEP_SC_NEWER_VERSION
  ```

  After the installation, the `xs list-components` command displays the software component as follows:

  ```
  xs list-components
  software component                version
  ----------------------------------------
  SCV_A (sap.com)                    1.2.3
  ```

10.2.2.2.5 Display installed Products and Software Components in XS Advanced Model

To display products and software components of SAP HANA XS advanced that are already installed, the `xs list-products` and `xs list-components` commands are available.

Prerequisites

The prerequisites are fulfilled as described in the *Prerequisites and Authorizations* topic. The link is in the *Related Information* section.

Context

Instead of `xs list-products` you can also use the `xs lp` alias. Instead of `xs list-components` you can also use the `xs lc` alias.

For more information, see the *XS CLI: Plugins* topic in *SAP HANA Developer Guide (for SAP HANA XS Advanced Model)*.

Procedure

1. Start the XS advanced command-line interface (CLI).
2. Log on to the SAP HANA XS advanced runtime in the organization and space where you want to display installed products or software components.
3. You have the following options:
   - To display all products that are installed in the current organization and space, use the `xs list-products` command without any arguments.

     ```
     xs list-products
     ```

     The system lists all installed products with information about vendor, version, support package level and installed instances.
   - To display all software components that are installed in the current organization and space, use the `xs list-components` command.

     ```
     xs list-components [--all]
     ```

     The system lists all installed software components with information about vendor and version.
     The version is displayed in the format `<software component version>.<support package level>.<patch level>`.
     If you use the `--all` option with the `list-components` command, the system also displays software components for which installations have failed and which are in status *BROKEN*.

- To display detailed information for a specific installed product, use the `xs list-products` command and add the name of the product `<PRODUCT NAME>` as argument. Optionally, or if another product with the same name and different vendor exists, add the `<VENDOR>`.

  > **Example**
  >
  > ```
  > xs list-products XSASAMPLEPRODUCT sap.com
  > ```

  > **Note**
  >
  > If the product name contains a space, enter the product name in quotation marks: `xs list-products "SAMPLE PRODUCT" sap.com`

  The system lists the specified product with information about vendor, version, and support package level. In addition, it lists all installed product instances and the software components that are assigned to the instances. For these, it lists the version and the state in which the software component exists in the system. They can have the following states:

  - *INSTALLED*: The software component or product instance is successfully installed.
  - *BROKEN*: The software component is installed in a broken state. This status can occur, for example, if there was an error in the deploy step during installation.
  - *INSTALLING*: The installation of this software component is currently running.
  - *INCOMPLETE*: The installation of this product instance is incomplete.
  - *MISSING*: This software component is missing.

Example

The output for the `xs list-products` command can look as follows:

> **Sample Code**
>
> ```
> name         vendor    version   SPS   instance ids
> ------------------------------------------------
> XSA Sample Product   sap.com   1.0       0     1,3
> ```

The output for the `xs list-products "XSA Sample Product"` command can look as follows:

> **Sample Code**
>
> ```
> --------------------
> name      XSA Sample Product
> vendor    sap.com
> version   1.0
> SP       0
> --------------------
> instance id    software component     version   state
> ------------------------------------------------------
> 1              -                       1.0       INSTALLED
>                JAVA_HELLO_XSA_B        1.0.0     INSTALLED
>                JAVA_HELLO_XSA_A        1.0.0     INSTALLED
> 3              -                       1.0       INSTALLED
>                JAVA_HELLO_XSA_C        1.1.0     INSTALLED
> ```

```
               JAVA_HELLO_XSA_D        1.1.0     INSTALLED
------------------------------------------------------------
```

Related Information

Prerequisites and Authorizations [page 627]
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
XS CLI: Plug-ins

10.2.2.2.6 Uninstall Products and Software Components in SAP HANA XS Advanced Model

Application lifecycle management for SAP HANA XS advanced model provides functions for uninstalling products as well as individual software components of SAP HANA XS advanced.

Prerequisites

- The prerequisites described under *Prerequisites and Authorizations* are fulfilled. The link to the topic is in the *Related Information*.
- You have a product or software component of SAP HANA XS advanced that you want to remove.

Context

You can uninstall products and software components of SAP HANA XS advanced that were installed using the `xs install` command.

Procedure

1. Start the XS advanced command-line interface (CLI).
2. Log on to the SAP HANA XS advanced runtime in the organization and space where you want to uninstall an installed product or software component.
3. Optional: Display the product or software component using the `xs list-products` or `xs list-components` command.
4. Start the uninstallation of the product or software component.

Enter the `xs uninstall` command and specify the name of the product or software component to be uninstalled, as well as the vendor, if required. In addition, you can enter options as required. The following is the default syntax for the `xs uninstall` command in the XS advanced CLI:

```
xs uninstall <NAME> [<VENDOR>] [-pv | --PRODUCT_VERSION] [-scv | --
SOFTWARE_COMPONENT_VERSION] [-f] [--ignore-scv-reuse] [--delete-services] [--
delete-service-brokers]  [--ignore-lock]
```

The following arguments are available:

Uninstallation Arguments

Uninstallation Argument	Description
`<NAME>`	The name of an installed product version (PV) or software component version (SCV)
[`<VENDOR>`]	The name of the vendor of the specified product or software component version; **optional**: only needed when the same product or software component name exists with different vendors

The following options are available:

Uninstallation Options

Uninstallation Option	Description
`-pv | --PRODUCT_VERSION`	Remove the specified product. To make sure that the entity you are about to uninstall is a product, you can add the `-pv` option. In this case, the uninstallation is only performed if you specify a product name as `<NAME>`. If you specify a software component name, the uninstallation will fail.
`-scv | --SOFTWARE_COMPONENT_VERSION`	Remove the specified software component. To make sure that the entity you are about to uninstall is a software component, you can add the `-scv` option. In this case, the uninstallation is only performed if you specify a software component name as `<NAME>`. If you specify a product name, the uninstallation will fail.

Uninstallation Option	Description
`--ignore-scv-reuse`	Remove the specified software component even if it is used in other installed products. **Note** You can use this option for uninstalling software components only. By default, a software component will not be uninstalled if it is also part of another installed product. You can override this behavior by using the `--ignore-scv-reuse` option.
`-f`	Remove the specified product or software component without any system prompts or confirmation
`-i, --INSTANCES <INSTANCE_1>[,<INSTANCE_2>]`	By default all instances are uninstalled; a comma-separated list of instances can be specified to limit the number of instances to be uninstalled
`--delete-services`	Recreate changed services and/or delete discontinued services
`--delete-service-brokers`	Delete discontinued service brokers
`--ignore-lock`	Force removal of the product or software component even if the target space is locked

Sample Code

```
xs uninstall 'XSA SAMPLE PRODUCT' -pv
```

Instead of `xs uninstall` you can also use the `xs unins` alias. For more information on the `xs uninstall` command, use the `xs help uninstall` command.

Results

The system undeploys and unregisters the specified product or software component from the SAP HANA server in the organization and space to which you are logged on.

If errors occur during the uninstallation, an error message indicates the reason for the error and the system provides a log with more detailed information. If you cannot solve the problem and you need to open a customer message, ensure that you assign it to the message component of the SAP HANA product or software component that caused the error. Do **not** assign the message to the component of SAP HANA application lifecycle management since this may slow down the problem solving process.

To display the correct log file, use the `xs display-installation-logs` command with the log ID that you find in the result of the uninstallation process and one of the `--unins_scv` or `--unins_pv` options.

```
xs display-installation-logs <log ID> --unins_scv
```

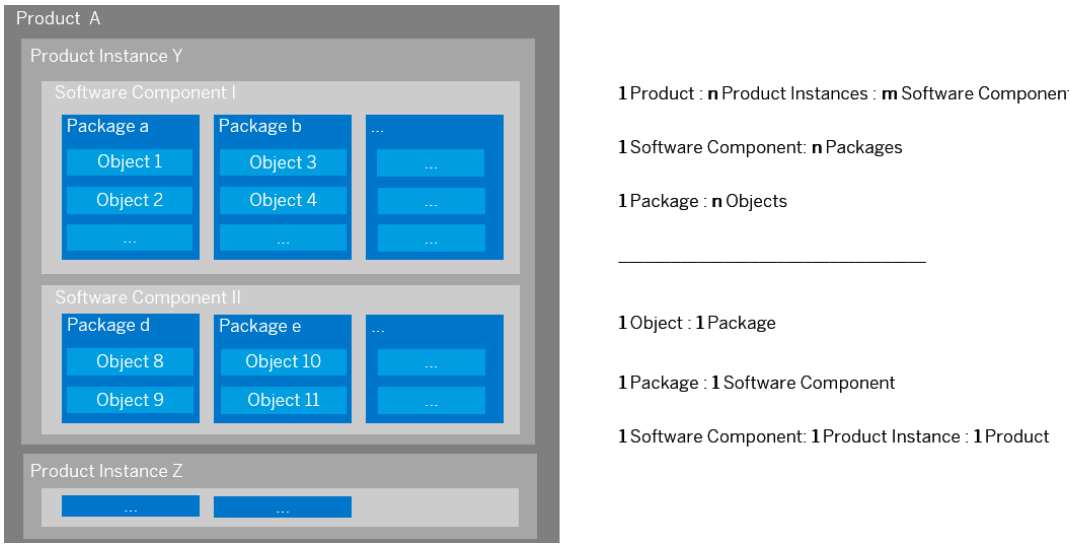
Related Information

Prerequisites and Authorizations [page 627]
XS CLI: Plug-ins

10.2.2.3 Installing and Updating Using the XS Advanced Application Lifecycle Management Graphical User Interface

You can use the XS Advanced Application Lifecycle Management graphical user interface to install, update, and uninstall products and software components in SAP HANA XS advanced model.

An SAP HANA product consists of one or several software components and can have one or several instances. Some SAP HANA software components require an MTA extension descriptor when they are installed. If this is the case for your product or software component, you can upload one or more MTA extension descriptor files together with the installation file. For more information on MTA extension descriptors, see The *MTA Deployment Extension Description* in the *SAP HANA Developer Guide for XS Advanced Model* and *The Multitarget Application Model* guide. The links can be found in the *Related Information* section.



Structure of an SAP HANA Product in XS Advanced

Accessing the XS Advanced Application Lifecycle Management Graphical User Interface

1. To access the XS Advanced Application Lifecycle Management, choose one of the following options:
   - Use the following URL: `https://<server>:53280/index.html`
     53280 is the default port for the XS Advanced Application Lifecycle Management graphical user interface when port-based routing is used.
     If the routing configuration was changed to hostname routing during the installation, the URL can look different. For more information on routing configuration, see the *SAP HANA Server Installation and Update Guide* and SAP Note 2245631. In this case, you can check for the URL to access XS Advanced Application Lifecycle Management in the following places:
     - Using the XS advanced command line interface (CLI):
       Use the `xs version` (`xs -v`) command. Below the information about the installed server version, under *Registered Service URLs*, it shows the *product-installer* URL.
     - Using the SAP HANA XS Advanced Cockpit:
       In the tile catalog, choose the *Application Monitor* tile. In the list of applications (opening may take some time), locate the *product-installer-ui* and choose the *URL* link.
   - In SAP HANA Cockpit, for a specific resource, choose one of the links available in the *Application Lifecycle Management* section:
     - For software components: *Install, update and uninstall XS advanced components*
     - For products: *Install, update and uninstall XS advanced products*
     - To display installed software components and products: *Show history*
   - In the XS Advanced Cockpit, from the *Spaces* in the *Organizations* section, choose the *SAP* space.
     In the *SAP* space, locate the link to the *product-installer-ui* and start the UI by clicking on the link displayed under *Application Routes*.
     The *SAP* space is only visible if the user is assigned to it as Space Manager.
2. Log in with your user credentials.
   For more information, see Prerequisites and Authorizations [page 627].
3. Select the space in which you would like to work by choosing *Switch Space* under *Additional Functionality*.

> **Note**
>
> When you use the XS Advanced Application Lifecycle Management graphical user interface for the first time, you are asked to select a space in which you would like to work. Select one from the list of spaces or use the search functionality on top of the list of spaces. This list displays only those spaces that you are authorized for. You can change the space that you are working in later on at any time.

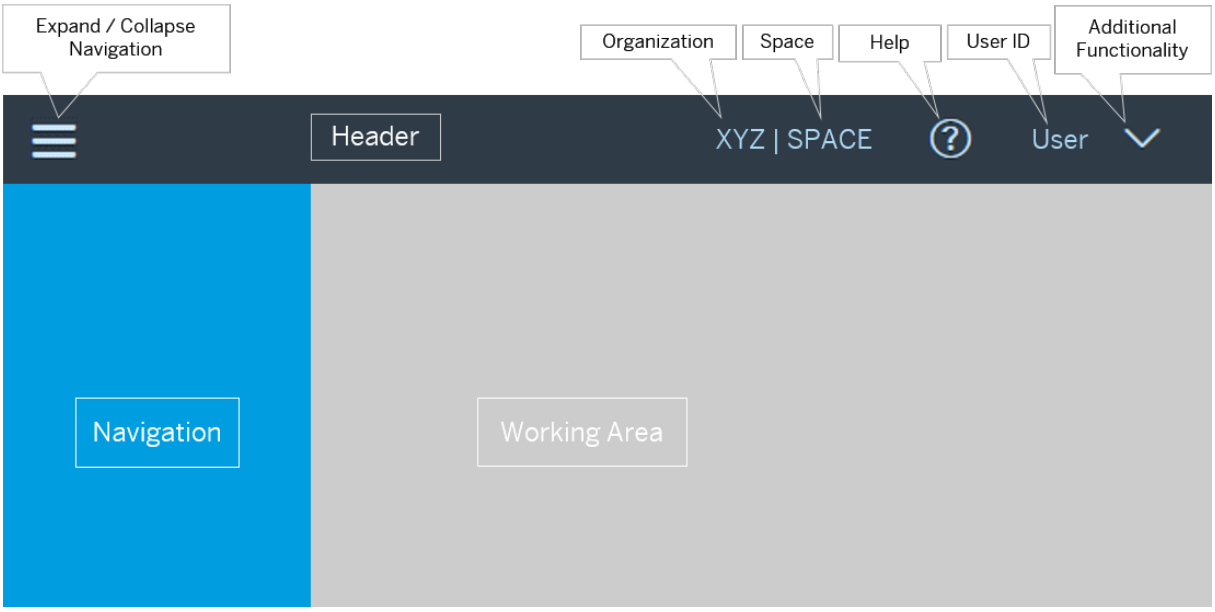
Elements of the XS Advanced Application Lifecycle Management Graphical User Interface

In the header of the graphical user interface on the right, you can find the name of the space and the SID of the system that you are currently working with. The questionmark leads you to detailed information about how to use the XS Advanced Application Lifecycle Management graphical user interface. By clicking on your user ID, additional functions like switching spaces and logging out are offered.

On the left side, you can decide whether you would like to work on software components or products or would like to take a look into the installation history. In the working area, you can see details for the topic that you chose:

- Software Components
  A list of the software components that are already installed in the selected space is shown. There are several options to manage the installed software components. If no software components are installed in the selected space, the list is empty.
- Products
  On the left, a list of products that are already installed in the selected space is shown. On the right, details for the product selected in the list are provided. If no products are installed in the selected space, the list is empty.
- Installation History
  The installation history provides an overview of all installation, update and uninstallation activities for both software components and products in the selected space. It comprises detailed information about the activities, for example, type and status of the activity, name of the product or software component, version before and after the activity was executed. For a detailed installation, update or uninstallation screen with all actions that were executed, you can click on an activity. You can also download the log if you need it for further analysis, go to the product concerned or to the screen of software components to see the current state.
  To find activities for a certain user or to search for a process ID, you can use the search above the list. The menu to the right of the search field offers different sorting options.

The following figure shows the elements of the XS Advanced Application Lifecycle Management graphical user interface:



Elements of the XS Advanced Application Lifecycle Management Graphical User Interface

Related Information

SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)
The MTA Deployment Extension Descriptor
The Multitarget Application Model

10.2.2.3.1 Install SAP HANA Software Components

You can use the XS Advanced Application Lifecycle Management graphical user interface to easily install new software components on your SAP HANA system.

Prerequisites

- The required installation file is available at a file location that you can reach from your computer.
- Make sure that you have selected the correct space.

Procedure

1. In the navigation area on the left, choose *Software Components* and then *Install/Update* in the working area.
2. Browse for the installation archive and, if required, enter one or more extension descriptors.
3. To upload the selected file, choose *Continue*.

   > ### Note
   >
   > If you want to clear the file selection, choose *Reset*. If you want to cancel the installation process, choose *Cancel*.

   A list of the uploaded software components with some detailed information (for example, currently installed version, uploaded version, status) is displayed. The status for the software component that you uploaded is *Installation*.
4. Start the installation.

   In the working area, you can now find information about the software component, its version, installation time and so on. You can also follow the installation progress which is done in the three steps *Validation*, *Deployment* and *Registration*. The icons for the different steps will change after a step is completed. You can click on the respective step for detailed information.

   As soon as a green hook is shown for each step, the installation is successfully done.

Results

The new software component is successfully installed on your SAP HANA system. You can check this by refreshing the list of installed software components.

10.2.2.3.2 Update SAP HANA Software Components

You can use the XS Advanced Application Lifecycle Management graphical user interface to easily update software components that are already installed on your SAP HANA system.

Prerequisites

- Make sure that the installation file is available at a file location that you can reach from your computer.
- Make sure that you have selected the correct space.
- A previous version of the software component has already been installed to the selected space on your SAP HANA system.
- The software component is not part of a product.

> ### Note
>
> A software component can only be updated if it is not part of a product. If it is part of a product, you need to update the complete product.

Procedure

1. In the navigation area on the left, choose *Software Components*.

   A list of the software components that are already installed in the selected space is shown.

   > ### Note
   >
   > If the list is empty, there are no software components installed in the selected space and you cannot update any software component. Make sure that you selected the correct space.

2. In the working area, choose *Install/Update*.
3. Browse for the installation archive and, if required, enter an extension descriptor.
4. To upload the selected file, choose *Continue*.

   > ### Note
   >
   > If you want to clear the file selection, choose *Reset*. If you want to cancel the update process, choose *Cancel*.

   A list of the uploaded software components with detailed information (for example, currently installed version, uploaded version, status) is displayed. Depending on the installed and uploaded versions of the software component, the following situations are possible:

   - If the version of the software component that you uploaded is higher than the one that is already installed, the status displayed in the list is *Update*. You can start the update directly.
   - If the version that is already installed is the same as the one that you uploaded, you cannot start the installation directly. The status for the selected software component is *Already installed, to overwrite*

set the corresponding option. To do so, choose *Options* in the upper right side of the working area and select *Overwrite the same version of software component*. This can be helpful if you would like to repair the installed version. In the list of uploaded software components, the status is changed to *Overwrite*. You can now start the update process.

- If the version of the software component that is already installed is higher than the one that you just uploaded, you cannot proceed. You cannot downgrade software components during the update process.

5. To start the update process, choose *Start Installation*.

   In the working area, you can now find information about the software component, its version, installation time and so on. You can also follow the installation progress which is done in the three steps *Validation*, *Deployment* and *Registration*. The icons for the different steps will change after a step is completed. You can click on the respective step for detailed information.

   As soon as a green hook is shown for each step, the installation is successfully done.

Results

The new version of the software component is successfully installed on your SAP HANA system. You can check this by refreshing the list of installed software components.

10.2.2.3.3 Install SAP HANA Products

You can use the XS Advanced Application Lifecycle Management graphical user interface to easily install new products on your SAP HANA system.

Prerequisites

- The required installation file is available at a file location that you can reach from your computer.
- Make sure that you have selected the correct space.

Procedure

1. In the navigation area on the left, choose *Products* and then *Install/Update* below the list of installed products in the working area.
2. Browse for the installation archive and, if required, enter an extension descriptor.
3. To upload the selected file, choose *Continue*.

> **Note**
>
> If you want to clear the file selection, choose *Reset*. If you want to cancel the installation process, choose *Cancel*.

Details about the uploaded product installation file (for example, vendor, version) are displayed. In addition, you can find information about the product instances that are part of the installation file. You can either decide to install the complete product with all instances or you can select one or several instances for installation.

> **Note**
>
> It might happen that software components that are part of the product to be installed are already installed on your system. In this case you have to decide on how to proceed with these software components before you can start the installation. To do so, choose *Options* in the upper right side of the working area and maintain the settings.

4. Start the installation.

   In the working area, you can now find information about the product, its version, installation time and so on. You can also follow the installation progress which is done in the three steps *Validation*, *Deployment* and *Registration*. The icons for the different steps will change after a step is completed. You can click on the respective step for detailed information.

   As soon as a green hook is shown for each step, the installation is successfully done. You can now download the installation log.

Results

The new product is successfully installed on your SAP HANA system. You can check this by refreshing the list of installed products.

10.2.2.3.4 Update SAP HANA Products

You can use the XS Advanced Application Lifecycle Management graphical user interface to easily update products that are already installed on your SAP HANA system.

Prerequisites

- Make sure that the installation file is available at a file location that you can reach from your computer.
- Make sure that you have selected the correct space.
- The product has already been installed to the selected space on your SAP HANA system.

Procedure

1. In the navigation area on the left, choose *Products*.

   A list of the products that are already installed in the selected space is shown.

   > ⓘ Note
   >
   > If the list is empty, there are no products installed in the selected space and you cannot update any product. Make sure that you selected the correct space.

2. In the workling area below the list of installed products, choose *Install/Update*.
3. Browse for the installation archive and, if required, enter an extension descriptor.
4. To upload the selected file, choose *Continue*.

   > ⓘ Note
   >
   > If you want to clear the file selection, choose *Reset*. If you want to cancel the update process, choose *Cancel*.

   A preview shows detailed information (for example, vendor, currently installed version, uploaded version) about the uploaded product and a list of product instances that are part of the uploaded installation file. You can either decide to update the complete product with all instances or you can select one or several instances for the update. The overview of instances also shows if software components of the respective instance are already installed. By default, instances that are already installed are selected for update. You cannot deselect them. To find out more about the software components concerned, you can click on the name of the instance.

   Depending on the installed and uploaded versions of the product and its software components, the following situations are possible:

   - If the versions of the product and all its instances and software components that you uploaded are higher than the versions of those that are already installed, you can start the update directly.
   - If an instance or some software components are already installed with the same version that is part of the uploaded file, these software components are not reinstalled automatically. To install them, choose *Options* in the upper right side of the working area and select *Overwrite the same version of software component*. This can be helpful if you would like to repair installed versions. In the list of instances, the *Prerequisite Check* is changed to *Same version will be overwritten*.
   - If the version of the software component that is already installed is higher than the one that you just uploaded, you would downgrade the already installed software components which is not possible during the update process. In this case, choose *Options* in the upper right side of the working area and select *Keep newer version of software component*.
   - You can select additional instances of the product for installation during the update process.

5. To start the update process, choose *Start Installation*.

   In the working area, you can now find information about the software component, its version, installation time etc. You can also follow the installation progress which is done in the three steps *Validation*, *Deployment* and *Registration*. The icons for the different steps will change after a step is completed. You can click on the respective step for detailed information.

   As soon as a green hook is shown for each step, the installation is successfully done.

Results

The new version of the software component is successfully installed on your SAP HANA system. You can check this by refreshing the list of installed software components.

10.2.2.3.5 Uninstall SAP HANA Products or Software Components

You can use the XS Advanced Application Lifecycle Management graphical user interface to easily uninstall products or software components from your SAP HANA system.

Prerequisites

The software component that you want to delete must not be part of an installed product.

> **Note**
>
> If the software component that you want to delete is part of an installed product, you cannot uninstall it unless you uninstall the complete product or the product instance that contains the software component.

Procedure

1. In the navigation area on the left, choose *Software Component* or *Product* according to your needs.

   A list of installed software components or products is shown. If you want to uninstall an individual instance of a product, you can select it in the details of the selected product.
2. Choose *Uninstall* for the software component or product that you would like to uninstall.

   You are prompted to confirm the uninstallation of the selected software component or product and to decide whether or not to delete services that were created during the previous deployment. These services might contain important user data, for example database content, that cannot be recovered after it is removed. If you choose to keep the services but want to delete them later, you can use the `xs delete-service` command in the command line client. For an overview of all existing services in a given space and their bound applications, use the `xs services` command.
3. To uninstall the software component or product and the related services if selected, choose *Uninstall*.

   In the working area, you can now find information about the software component or product, its version, installation time and so on. You can also follow the uninstallation progress which is done in the three steps *Validation*, *Undeployment* and *Deregistration*. The icons for the different steps will change after a step is completed. You can click on the respective step for detailed information.

   As soon as a green hook is shown for each step, the uninstallation is successfully done.

Results

The software component or product is successfully uninstalled from your SAP HANA system. You can check this by refreshing the list of installed software components or products.

10.2.3 Configuring SAP HANA Applications with the Process Engine

The Process Engine (PE) is a framework available with SAP HANA application lifecycle management to enable automated technical configuration.

After the installation of a product or a delivery unit, an application typically must be configured before it can be used. The configuration tasks are described in the installation guides that are provided on the SAP Help Portal (`help.sap.com`). Instead of performing cumbersome and error-prone manual activities, you can use the Process Engine to automate application configuration completely or partially. As a prerequisite, your application must provide content for the automated technical configuration.

The Process Engine (PE) framework is installed with SAP HANA application lifecycle management as automated content. It is available from the following locations:

- On the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm/pe/ui/`
- Using the *Configuration Services (Process Engine)* tile in SAP HANA Application Lifecycle Management XS user interface
- Using the *Manage Configuration Services* tile in SAP HANA cockpit

Related Information

Tutorial: Execute a Configuration Service with Process Engine [page 657]
Process Engine Roles [page 656]
Troubleshooting [page 660]

10.2.3.1 Process Engine Roles

To grant users the privileges they require to perform tasks with the Process Engine, you must assign them the relevant Process Engine roles.

The following table lists the roles that are available for tasks related to the Process Engine. The roles are hierarchical and interlinked. The `sap.hana.xs.lm.roles::Administrator` role is the *Administrator* role of SAP HANA application lifecycle management and grants the privileges of all other Process Engine-related roles as well as application lifecycle management roles. For more information, see *SAP HANA Application Lifecycle Management Roles* in the *SAP HANA Application Lifecycle Management Guide*.

→ Recommendation

As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

Roles available for the Process Engine

Role	Description
`sap.hana.xs.lm.pe.roles::PE_Display`	The user can monitor processes and display services.
`sap.hana.xs.lm.pe.roles::PE_Execute`	In addition to the previous role, the user can start, stop, skip, and resume processes.
`sap.hana.xs.lm.pe.roles::PE_Activate`	In addition to the previous roles, the user can activate services from repository files.
`sap.hana.xs.lm.roles::Administrator`	The user can install products. This role includes all previous roles.

Related Information

SAP HANA Application Lifecycle Management Roles

10.2.3.2 Tutorial: Execute a Configuration Service with Process Engine

In this tutorial, you use the demo content delivered with the Process Engine to execute a configuration service.

Prerequisites

- An SAP HANA system is available.
- SAP HANA XS is up and running on the SAP HANA system.
- Depending on the task you want to perform with the Process Engine, you must have the privileges based on a role granted by one of the Process Engine role templates described in *Process Engine Roles*. The link to the topic is in the *Related Information* section. The privileges of the `sap.hana.xs.lm.pe.roles::PE_Activation` role allows you to perform all Process Engine tasks.

Context

The Process Engine uses different terms for identifying design time or runtime artifacts. The *service* is the core entity at design time. It has multiple attributes describing its purpose and steps representing the executable entities. They perform the actual work during execution. An executable can be a JavaScript function in an XS JavaScript library or an SQL stored procedure. When starting a service, the Process Engine creates a *process* based on a service. It copies all steps associated with the service as tasks, and it copies the parameters of the selected variant to the parameters of the process. Furthermore, the Process Engine associates a *status* with the process.

You execute the following steps to configure the demo service:

- Activate the demo service.
  Services are delivered as repository objects. The services required by the administrator need to be enabled once before use. This activity is called *activation*.
- Prepare the demo service parameters.
  The demo service needs parameters during execution. The set of required parameters is stored under a common key, the *variant*. Before you can start a service you need to prepare variants. Since you are about to start the service for the first time, you do not have any variants prepared. If you repeat an execution, you can use an existing variant. For the demo service, you enter *user* and *password*. Since this is a demo example, the user does not need to exist and the password can be any set of characters.
- Start the demo service.
  The demo service consists of the following steps:
  - JS_APPVAR by JavaScript
    This step executes a JavaScript function that shows how to consume and return parameters in JavaScript.
  - SQL_APPVAR by SQL Script
    This step executes a SQL script function that shows how to consume and return parameters in SQL script.

> **Note**
>
> The demo content does not perform any configuration of the system. It only writes messages into the log of the Process Engine. It provides you with a hands-on experience for using the Process Engine.

Procedure

1. Open SAP HANA Application Lifecycle Management.

   SAP HANA Application Lifecycle Management is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/lm`
2. Choose *Configuration Services (Process Engine)*.

   The process engine opens in a new browser window or a tab.
3. Select *Services* on the left-hand side of the screen.

   A list of services available for configuration appears. Active services are indicated by a green status icon, inactive services have a grey status icon. Inactive services must be activated before they can be started.

You find the demo service *DEMO_VarCont* as inactive in the list.

4. Select *DEMO_VarCont* and choose *Activate*.

   Note that the activation process can take some time. After the service was activated successfully, you can start it.

   If the activation were not successful, you can find the error messages in a detailed log.

5. To display the service after the activation, choose *Go To Service*.

   The details of the active service *Demo Service with Process Engine Variant Container* appear.

6. To prepare the parameters for the demo service, choose *Maintain Variants*.

   A new screen for variant maintenance appears.

7. Enter a *user* name and a *password* as parameters and choose *Save As*.
8. Enter a *variant ID* and, optionally, a *description*, and then choose *Create*.
9. Return to the previous screen to view the variant that you just created.

   If you open the *Steps* tab on screen, two executable steps are displayed.

10. To start the service, select your variant and choose *Start Variant*.

    The *Process* tab opens and a new process appears at the top of the list.

11. Select the process to navigate to the process details.

    A list of tasks appears.

12. Choose *Refresh* to observe the progress of the process.

    The overall process status is displayed on top of the progress bar. It is a cumulation of the statuses of the individual tasks.

    The status icons allow you to intervene in the process execution if errors occur. You can click on an icon to display the task log.

13. When the process completion reaches 100%, choose the *Parameters* tab.

    You see an overview on the scalar parameters and their changes during execution.

    Entries for the step *NA* show the parameter values after the variant container is copied and before the process execution starts. The other entries show the values after the step was executed.

14. To view the log of the *Consuming and returning parameters via SQL* task, select this task at the bottom of the screen.
    a. Search for a message with a green status that starts with *JavaScript function sends*. At the end of the message, you see the parameter value of your **user**.
    b. Search for a message with a green or orange status that contains the text *... password received*.

       The step compares the received value of the parameter you entered as **password** with a value set by the demo service. If you entered the password as set by the demo service, the Process Engine issues the message *Correct password received*. If you entered a different password, the Process Engine writes *Incorrect password received* in the log.

Results

You have used the demo configuration service of the Process Engine. You have activated the demo service, prepared the parameters, and executed the service. Afterward, you have checked the logs of the Process Engine.

Related Information

Process Engine Roles [page 656]
Troubleshooting [page 660]

10.2.3.3 Troubleshooting

If a process stops with errors, you should first analyze the logs to find out why an error occured. Afterward, you have various options to respond to the error situation.

The Process Engine provides a process log and a task log. If a single task has an error you can start with the task log to analyze if an error message is due to a specific step. If this does not help, you can open the process log and search or filter for error messages.

- **Process Log**
  This is a collection of all task logs and additional entries related to the process. This log contains all messages with technical details, including the log of the internal activities of the Process Engine. You can find this log when you open the *Log* tab in the single process view.
- **Task Log**
  This is the log of the execution of a single task. Messages with technical details about the Process Engine usually are not displayed here.You get this log when you navigate to a task view by clicking on a task in the single process view.

You have the following options to respond to an error:

- If the error is only temporary or you solved the error already, you can execute the step again by choosing *Resume*.
- You can decide to perform the task manually and skip the execution of the task by choosing *Skip*.
- You can cancel the current process and start a new one. To do this, choose *Cancel*.
- If you cannot resolve the error, and you need to contact SAP, open an incident and assign it to the support component of the application that provides the configuration content or, alternatively, to component HAN-LM-APP. Make sure that you attach the diagnosis information that you can download for each process using the link on the *Diagnosis Information* tab.

10.3 SAP HANA Content

SAP HANA content is structured in the way that delivery units (DUs) are used to group SAP HANA content artifacts (such as analytic, attribute or calculation views, and SQLScript procedures).

DUs are grouped to SAP HANA products in order to ship and install SAP HANA applications with all dependent artifacts (grouped in DUs). To distribute SAP HANA content, a product archive (*.ZIP file) or a delivery unit archive (*.tgz file) is used. There are various ways of acquiring and deploying these archive types.

SAP HANA content, which is developed on SAP HANA Extended Application Services (SAP HANA XS), classic model, can also be grouped in a DU.

For more information about SAP HANA content deployed automatically during platform installation or upgrade, see *Components Delivered as SAP HANA Content* in the *SAP HANA Security Guide*.

Related Information

Components Delivered as SAP HANA Content

10.3.1 SAP HANA Archive Types

The difference between the various archive types is their method of deployment, and when the content is deployed.

The following archive types are available:

- **Product archive file** (`*.ZIP`)
  A product version archive is a `*.ZIP` file containing 1-n software component archive files and the following metadata files: `stack.xml`, `pd.xml`. A software component archive file is created for each DU containing its archive file (`*.tgz`).
  A product is usually the entity that delivers SAP HANA applications, but it can also be used for transports. SAP HANA content that can be downloaded independently is shipped as SAP HANA products in SAP HANA product archives. SAP HANA content that is not part of the SAP HANA database is called SAP HANA content add-on (or SAP HANA product). SAP HANA content add-ons are developed as part of the SAP HANA platform or as part of an application that runs on top of SAP HANA.
  For information about how to deploy a product archive, see *Deploy a Product Archive (*.ZIP)*.
- **Software Component Archive** (`*.ZIP`)
  A software component archive is a `*.ZIP` file (in previous versions also `*.SAR` files were delivered as software component archives) containing one delivery unit archive file (`*.tgz`) and (optionally) a corresponding translation DU and the metadata file `SL_MANIFEST.XML`. A software component archive can be deployed with the same tool as product archives.
  For information about how to deploy a software component archive, see *Deploy a Product Archive (*.ZIP)*.
- **Delivery unit archive file** (`*.tgz`)
  A delivery unit archive is a `*.tgz` file containing the SAP HANA content artifacts that are created in the SAP HANA repository. A DU is used to deliver one or more software components from SAP (or a partner) to a customer.

For distribution using export/import and deployment, a DU is contained in a delivery unit archive (`*.tgz` file). It contains the objects and packages of a DU together with the metadata file `manifest.txt`. The transport is also offered at DU level.
The following types of delivery unit archive files are available:

- **Delivery unit archives as part of the SAP HANA database**
  The following types of delivery unit archive files that are part of the SAP HANA database are available:
  - **Automated content** is installed together with SAP HANA and imported into the SAP HANA repository during installation. This is an integral part of the SAP HANA database and is used by every SAP HANA database customer.
    Automated content is located on the SAP HANA system in the following folder: `/usr/sap/<SID>/SYS/global/hdb/auto_content`.
  - **Non-automated content** is installed with SAP HANA, but needs to be imported into the SAP HANA repository manually by the system administrator. It is used for integral parts of the SAP HANA database, but is only used by a small number of customers.
    Non-automated content is located on the SAP HANA system in the following folder: `/usr/sap/<SID>/SYS/global/hdb/content`.
    Delivery unit archives that are non-automated content of the SAP HANA database need to be deployed manually.
- **Independent delivery unit archives that are not part of the SAP HANA database**
  Delivery unit archives that are not installed together with the SAP HANA database and are not part of the SAP HANA database need to be deployed manually.

For information about how to deploy or activate a delivery unit archive, see *Deploy a Delivery Unit Archive (*.tgz)*.

Related Information

Deploy a Product Archive (*.ZIP) [page 662]
Deploy a Delivery Unit Archive (*.tgz) [page 663]

10.3.2 Deploy a Product Archive (*.ZIP)

SAP HANA application lifecycle management provides a method of deploying a product archive file (`*.ZIP` file containing a product) or software component archive files (`*.ZIP`).

For more information, see *Installing and Updating SAP HANA Products and Software Components* in the *SAP HANA Application Lifecycle Management Guide*.

Related Information

Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]

10.3.3 Deploy a Delivery Unit Archive (*.tgz)

The following deployment methods for deploying a delivery unit archive file (*.tgz file containing a DU) are provided:

- SAP HANA Application Lifecycle Management
  Choose *Products* > *Delivery Units* > *Import*.
  This tool runs on the SAP HANA XS Web server.
  For more information, see *Import a Delivery Unit* in the *SAP HANA Developer Guide (For SAP HANA Studio)*.
- SAP HANA Application Lifecycle Management
  SAP HANA application lifecycle management provides functions for installing and updating SAP HANA products:
  - SAP Fiori application integrated in the SAP HANA Application Lifecycle Management XS application
  - `hdbalm` command line tool

  For more information, see *Installing and Updating SAP HANA Products and Software Components* in the *SAP HANA Application Lifecycle Management Guide*.
- SAP HANA studio
  Import function of the SAP HANA Modeler
  Choose *File* > *Import* > *SAP HANA Content* > *Delivery Unit*.

For more information, see *SAP HANA Modeling Guide*.

Related Information

Import a Delivery Unit

Installing and Updating SAP HANA Products and Software Components in SAP HANA XS Classic Model [page 618]

11 Landscape Management and Network Administration

Manage your SAP HANA landscape and integrate SAP HANA into your network environment.

Related Information

Landscape Management [page 664]
Network Administration [page 714]

11.1 Landscape Management

Manage your SAP HANA system landscape efficiently and respond flexibly to changing resource requirements.

Depending on your SAP HANA deployment model and landscape architecture, you can reconfigure and reorganize your system in a number of ways.

SAP HANA Tools

Copy and Move Operations

The following SAP HANA mechanisms allow you to copy and move systems and databases:

- Platform LCM Tools:
  An SAP HANA system can be safely and efficiently reconfigured by decoupling the system hosts from the installation path through unregistration, and re-coupling them in a different configuration through registration. System reconfiguration tasks can be performed with the SAP HANA database lifecycle manager (HDBLCM).
- SAP HANA System Replication:
  SAP HANA system replication can be used to create a copy of an SAP HANA system in a quick and simple way.
  System replication mechanisms can also be used to copy and move tenant databases securely and conveniently from one SAP HANA system to another with near-zero downtime.
- Backup and Recovery:
  You can create a homogeneous copy of an SAP HANA database by recovering an existing database to a different database. A homogenous database copy is a quick way to set up a cloned database, for example, for training, testing, or development.

System Rename

An SAP HANA system can be renamed by changing the system identifiers, like host names, SID, and instance number. Changing system identifiers can be performed with the SAP HANA database lifecycle manager (HDBLCM).

SAP Landscape Management

SAP Landscape Management is an add-on to SAP NetWeaver installed as an application with the SAP NetWeaver Application Server for Java (SAP NetWeaver AS for Java).

The enterprise edition of SAP Landscape Management software helps you to simplify and automate the efforts required to configure, provision, deploy, monitor, and manage your systems in both physical and virtualized infrastructures.

With SAP Landscape Management you can prepare, relocate, restart, start, stop, and unprepare single-host and multiple-host SAP HANA systems, and perform system replication operations.

Security Considerations

When creating a system copy you should consider the following security-related tasks:

- Update TLS certificate to match the new environment.
- Rotate encryption keys for persistence and log encryption.
- Update own certificates in in-database PSEs to make them unique.
- Smart Data Access (SDA) connections: check whether the copy of your system should still have access to the systems the original system has access to.
- Remote table replication: check whether the copy of your system should still act as the source or replica system like the original.

Related Information

Copying and Moving a System Using Platform LCM Tools [page 666]
Copying a System Using System Replication [page 705]
Copying and Moving Tenant Databases [page 671]
Copying a Database Using Backup and Recovery [page 1065]
Renaming a System [page 706]
SAP Landscape Management, enterprise edition

11.1.1 Copying and Moving a System Using Platform LCM Tools

An SAP HANA system can be safely and efficiently reconfigured by decoupling the system hosts from the installation path through unregistration, and re-coupling them in a different configuration through registration. System reconfiguration tasks can be performed with the SAP HANA database lifecycle manager (HDBLCM).

11.1.1.1 Relocate the SAP HANA System

It may become necessary to move the SAP HANA system to different hardware. If so, you need to unregister the SAP HANA system and re-register it on the new hardware. System relocation can be performed with the SAP HANA database lifecycle manager (HDBLCM).

Context

Relocation can be performed on both the entire SAP HANA system or on an individual SAP HANA instance. So, you can flexibly decide if you want to relocate only one host (for example, in the case of host outage) or relocate all hosts in the system (for example, in a system scale up).

> **Note**
>
> An SAP HANA system can only be relocated to a target system that runs on the same hardware platform as the source system.

Procedure

1. Unregister the SAP HANA instance or the SAP HANA system.
   a. Log on to the SAP HANA source host.

      If you are unregistering a SAP HANA multiple-host system, you can log on to any system host. If you are unregistering a multiple-host system and would like to unregister one instance at a time, perform the unregistration of each local host.
   b. Change to the SAP HANA resident HDBLCM directory:

      ```
      cd <sapmnt>/<SID>/hdblcm
      ```

      By default, `<sapmnt>` is `/hana/shared`.
   c. Start the unregister task:
      - To unregister hosts using the SAP HANA database lifecycle manager command-line interface:
        - Start the command-line tool interactively:

          ```
          ./hdblcm
          ```

and enter the index of the `unregister_instance` action, if you only want to unregister the local host from the SAP HANA system. Enter the index of the `unregister_system` action, if you want to unregister all hosts in the SAP HANA system. Or,

- To unregister hosts using the SAP HANA database lifecycle manager graphical user interface:
  1. Start the graphical user interface tool:

     ```
     ./hdblcmgui
     ```

     The SAP HANA database lifecycle manager graphical user interface appears.
  2. Choose *Unregister SAP HANA System*.

d. To continue with the task proceed as follows:

- In the command line interface: Enter *y*.
- In the graphical interface:
  1. To display the summary of the configuration data, choose *Next*.
  2. To execute the configuration task, choose *Run*. The system displays the configuration progress.
  3. After the configuration task has finished, you can:
     - View the log. To do so, choose *View Log*.
     - Exit the graphical user interface. To do so, choose *Finish*.

2. Mount the installation path (`sapmnt`), the datapath, and the logpath on the target hosts.
3. If the local secure store (LSS) is installed, copy its installation directory (default: `/lss/shared`) to the target host.
4. Register the new host.

a. Log on to the SAP HANA target host.

   If you are registering an SAP HANA multiple-host system, you can log on to any system host. If you are registering a multiple-host system and would like to register one instance at a time, perform the registration on the local host before the remote hosts.

b. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.

c. Start the register task:

- To register hosts using the SAP HANA database lifecycle manager command-line interface:
  - Start the command-line tool interactively:

    ```
    ./hdblcm
    ```

    and enter the index of the `register and rename` action, or
- To register hosts using the SAP HANA database lifecycle manager graphical user interface:
  1. Start the graphical user interface tool:

     ```
     ./hdblcmgui
     ```

     The SAP HANA database lifecycle manager graphical user interface appears.
  2. Choose *Register and Rename SAP HANA System*.

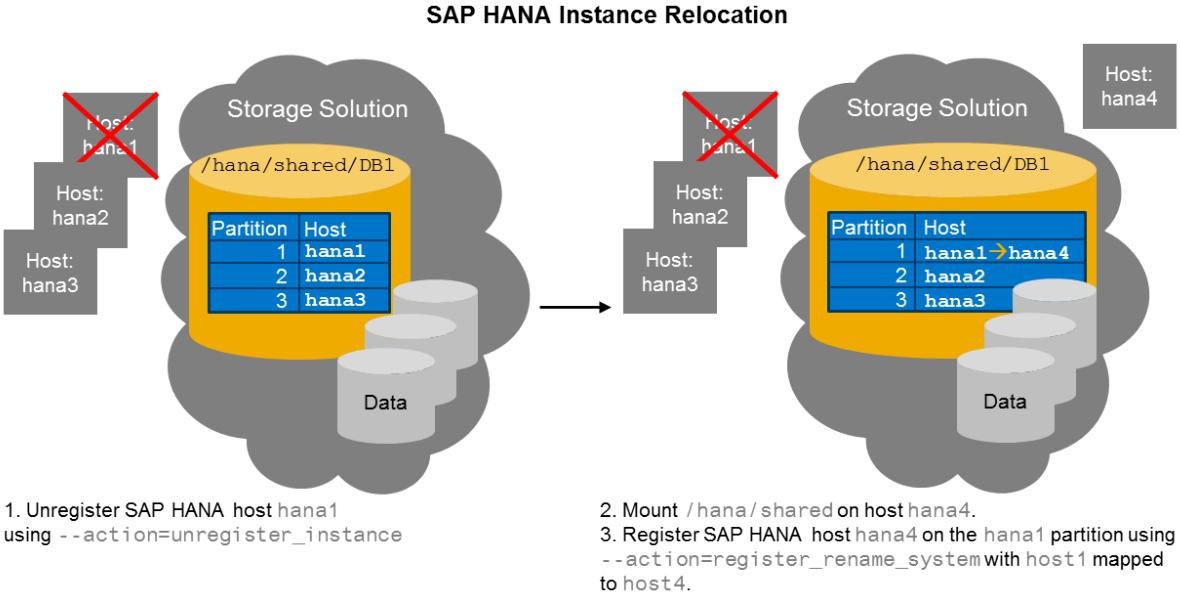
d. To continue with the task proceed as follows:

- In the command line interface: Enter *y*.
- In the graphical interface:
  1. To display the summary of the configuration data, choose *Next*.
  2. To execute the configuration task, choose *Rename*. The system displays the configuration progress.
  3. After the configuration task has finished, you can:
     - View the log. To do so, choose *View Log*.
     - Exit the graphical user interface. To do so, choose *Finish*.

> **Note**
>
> When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the SAP HANA resident HDBLCM directory of the SAP HANA system, execute the following command:
>
> ```
> ./hdblcm --action=unregister_instance --help
> ```
>
> ```
> ./hdblcm --action=unregister_system --help
> ```
>
> ```
> ./hdblcm --action=register_rename_system --help
> ```

Example

The following is an example of SAP HANA instance relocation from one host to another:



Related Information

SAP HANA Server Installation and Update Guide

11.1.1.2 Copy or Clone an SAP HANA System

You can use the SAP HANA database lifecycle manager (HDBLCM) to make a copy or a clone of an SAP HANA system by copying the file system containing the SAP HANA database installation from an old storage solution to a new storage solution, and registering the copied SAP HANA system on new hosts.

Prerequisites

Before cloning the SAP HANA system, you must create a physical copy of the SAP HANA system (storage snapshot, file systems copy). The source system must be offline or a database snapshot must have been taken on the source system before the physical copy of the SAP HANA system is created.

> ⓘ Note
>
> An SAP HANA system can only be cloned or copied to a target system that runs on the same hardware platform as the source system.

Context

Cloning an SAP HANA system produces a new SAP HANA system, identical to the existing one. Copying an SAP HANA system produces a new SAP HANA system with the same landscape as the existing one, but slightly different parameter settings. If the interactive parameter defaults are accepted during host registration, the system is effectively cloned. If the new system parameters are set to different values, the new system is similar, but not identical to the source system.

You could, for example, copy an existing production system, and accept all parameter defaults during host registration except `system_usage`, which would be specified as "test". This configuration would allow you to have an almost identical copy of the existing system for test or quality assurance purposes.

> ⚠ Caution
>
> Keep in mind that in a system copy refresh scenario all users and roles are overwritten in the target system.

Procedure

1. Copy the file system containing the SAP HANA database installation from the old storage solution to the new storage solution.

2. Mount the installation path (`sapmnt`), the data path, and the log path on the target hosts.
3. If the local secure store (LSS) is installed, copy its installation directory (default: `/lss/shared`) to the target host.
4. Register the new SAP HANA system on the target hosts.
   a. Log on to the SAP HANA target host.
   b. Change to the SAP HANA resident HDBLCM directory:

      ```
      cd <sapmnt>/<SID>/hdblcm
      ```

      By default, `<sapmnt>` is `/hana/shared`.
   c. Start the register task:
      - To register hosts using the SAP HANA database lifecycle manager command-line interface:
        - Start the command-line tool interactively:

          ```
          ./hdblcm
          ```

          and enter the index of the `register_rename_system` action, or
      - To register hosts using the SAP HANA database lifecycle manager graphical user interface:
        1. Start the graphical user interface tool:

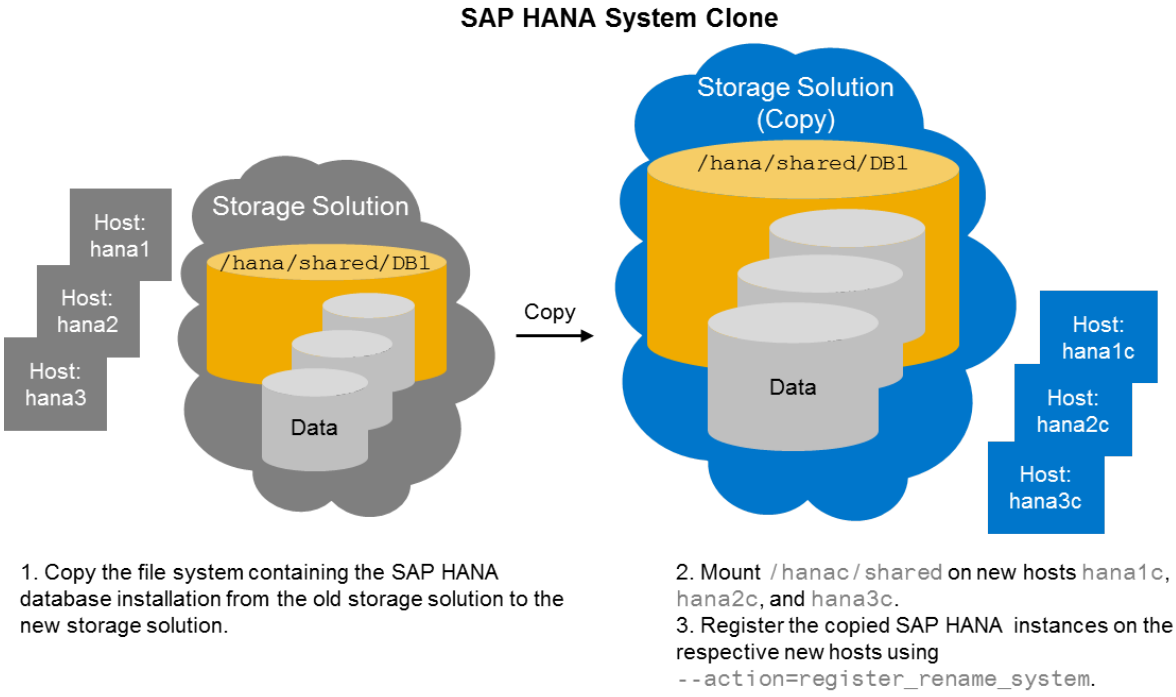
           ```
           ./hdblcmgui
           ```

           The SAP HANA database lifecycle manager graphical user interface appears.
        2. Choose *Register and Rename SAP HANA System*.
   d. To continue with the task proceed as follows:
      - In the command line interface: Enter *y*.
      - In the graphical interface:
        1. To display the summary of the configuration data, choose *Next*.
        2. To execute the configuration task, choose *Run*. The system displays the configuration progress.
        3. After the configuration task has finished, you can:
           - View the log. To do so, choose *View Log*.
           - Exit the graphical user interface. To do so, choose *Finish*.

> **Note**
>
> When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the SAP HANA resident HDBLCM directory of the SAP HANA system, execute the following command:
>
> ```
> ./hdblcm --action=register_rename_system --help
> ```

Example

The following is an example of an SAP HANA being cloned:



Related Information

SAP HANA Server Installation and Update Guide

11.1.2 Copying and Moving Tenant Databases

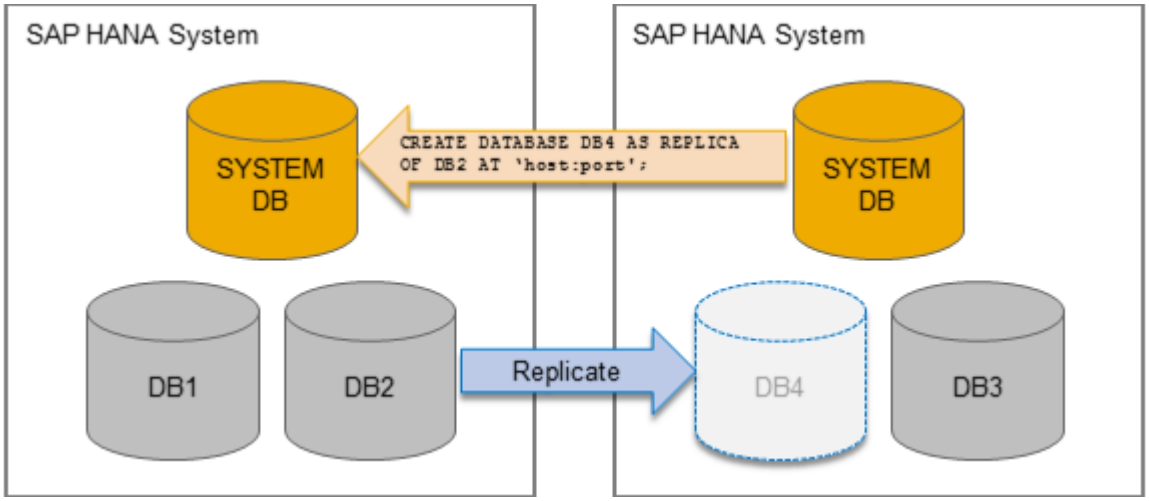
Using SAP HANA system replication mechanisms, SAP HANA tenant databases can be copied and moved securely and conveniently from one SAP HANA system to another with near-zero downtime. This allows you to respond flexibly to changing resource requirements and to manage your system landscape efficiently.

The following sections provide an overview of copying or moving a tenant database using system replication.

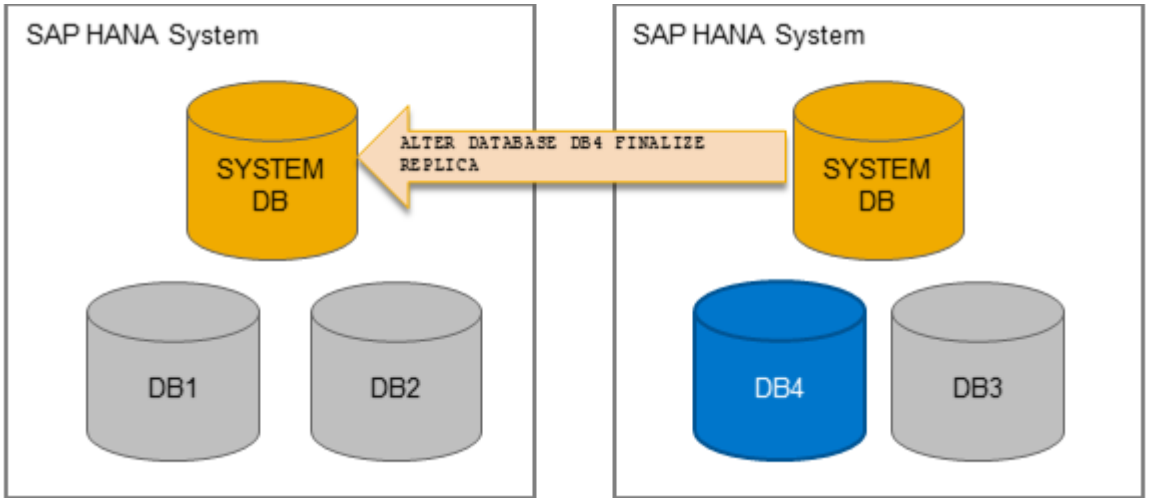
- Process Overview [page 672]
- Use Cases [page 673]
- Which Data Is Copied or Moved? [page 674]
- Recoverability After Copy or Move [page 674]
- Client Communication After Move [page 675]
- Prerequisites and Implementation Considerations [page 675]
- Other Copy and Move Methods [page 676]

Process Overview

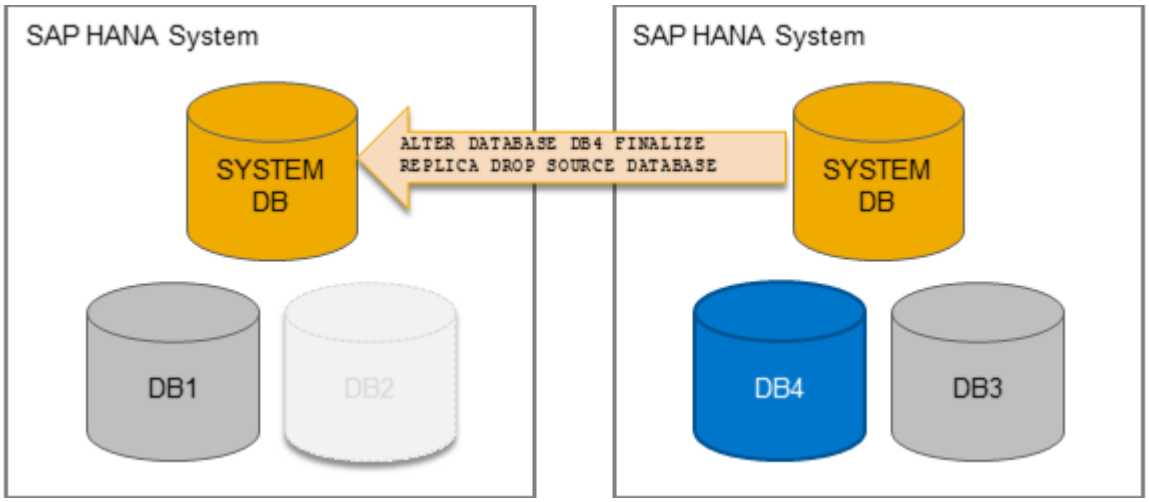
Copying and moving a tenant database are essentially the same process. First, the tenant database is copied through the replication of all of its data to a newly created tenant database in a target system:



Once all data has been successfully transferred, the new tenant database is started as a separate, independent database:



If the aim is to move the tenant database to the new system, the original tenant database is deleted and the new tenant database takes over:



The only difference between copying and moving a tenant database therefore is what happens to the original tenant database after all data has been transferred to the new tenant database in the target system.

In both cases, the new tenant database starts running as a fully separate, independent database.

Several tenant databases can be copied or moved to a system at the same time. It is also possible to copy or move a tenant database to a system with a different isolation level than the source system.

You can also use this mechanism to create a copy of a tenant database on the same host or to copy/move a tenant database to another host that is part of the same system.

Use Cases

Copying and moving a tenant database from one system to another in this way has several applications, including:

- Load balancing between systems
  For example, a tenant database is running a more demanding workload than anticipated, so you move it to a system running on a host with more CPU resources.
- Management of deployment environment
  For example, you want to copy a tenant database running in your test system to the live production system.
- Tenant-database-specific upgrades
  For example, you want to upgrade a single tenant database but not the entire system, so you move the tenant database to a system already running the higher version.
- Template databases
  For example, you create a tenant database with a default configuration that you want to reuse as the basis for new tenant databases in other systems. You can simply copy the tenant database as a template to other systems.

Which Data Is Copied or Moved?

When a tenant database is copied or moved, data is replicated from the original tenant database to the new tenant database in the target system.

The following table indicates which types of data are replicated and which are not.

Type of Data	Replicated?
Data and logs of the tenant database	Yes
Trace and log files	No
Data backups	No
Configuration (*.ini) files with tenant-database-specific values This refers to files in the directory `$DIR_INSTANCE/../SYS/global/hdb/custom/config/<database_name>`	No
Certificates and certificate collections stored in the tenant database This refers to the digital certificates and certificate stores used for certificate-based user authentication and secure communication between SAP HANA and JDBC/ODBC clients. **Note** If these certificates are stored in the file system in personal security environments (PSEs), they will not be replicated. To ensure that they are replicated, migrate the file-system-based PSEs to in-database certificate collections before copying or moving the tenant database. For more information about how to do this, see SAP Note 2175664.	Yes
Encryption root keys (for data and log volume encryption, backup encryption and the internal data encryption service) **Note** The root key used for data volume encryption is not replicated during the move process.	Copy: No Move: Partially
Key management configurations if the local secure store (LSS) is used with an external key management system (KMS)	Yes
Application function libraries	No

Recoverability After Copy or Move

When you copy a tenant database, the new tenant database does not have a backup history and cannot be recovered immediately after being copied. For this reason, it is important to perform a full data backup after you copy.

When you move a tenant database, the backup history of the original tenant database is retained in the new tenant database. As long as data and log backups of the source system are at a location accessible to the target system, the new tenant database is recoverable immediately after the move.

> ⚠ Caution
>
> If you subsequently create a tenant database in the source system with the same name as the moved tenant database, the backup files of the original database are overwritten.

Client Communication After Move

We recommend that you do not specify physical host names in the SQL client connect string. Otherwise, you would have to reconfigure all of your applications after a move. Instead, configure virtual host names or virtual IP addresses. For more information on virtual IP addresses, see *Configure Host-Independent Tenant Addresses*.

Prerequisites and Implementation Considerations

- The copy and move process involves the creation of a new tenant database in the target system. Therefore, the target tenant database must not already exist in the target system.
- The target system must have a software version equal to or higher than the source system.
- If the source system is using the LSS in conjunction with an external KMS, then the target system must also use the LSS and a key management configuration ID (other than `DEFAULT`) must be provided to initiate the copy or move. Only a key administrator (user with `ENCRYPTION ROOT KEY ADMIN`) system privilege can obtain this information from the system view `SYS.KEY_MANAGEMENT_CONFIGURATIONS`. The key administrator may add a dedicated configuration for the target system in the source system before the copy or move.
  If the source system is not using the LSS with a KMS, then the target system may use either the LSS or the SSFS secure store in the system. For more information, see *Server-Side Secure Stores* and *Using the LSS with an External Key Management System*.
- If data volume encryption is enabled in the original system, data is decrypted before replication and then re-encrypted (with a new root key) in the new database. However, during the copy and move process, data must be replicated via a secure (SSL/TLS) network connection by default.
- In a running system replication, it is possible to copy or move tenant databases into a primary system or from a primary system into another target system different than the secondary system.
- There can be no changes to the topology of the original tenant database while the move or copy is in progress. In other words, until the copy or move has been finalized, it is not possible to add services to or remove services from the source tenant database.
- If the source system is configured for host auto-failover, the copy or move process will fail in the event of failover to a standby host. If this happens, the simple solution is to delete the new tenant database on the target system, which will interrupt and clean up the copy or move process on the source and target side. Afterwards, the copy or move process has to be restarted. Alternatively, if restarting the process is not feasible, altering the network settings to make the standby host accessible through the same IP address as the failed host will resume the copy or move process.
- The following components must not be configured in the source tenant database:
  - Rserve server
  - SAP HANA dynamic tiering (extended storage server)
  - SAP HANA accelerator for SAP ASE (extended transaction service)

- SAP HANA streaming analytics (streaming host)

Other Copy and Move Methods

Backup and Recovery

It is possible to use backup and recovery to copy or move tenant databases between two systems. However, we recommend using SAP HANA system replication as described here. The main advantage of using system replication over backup and recovery is the absence of downtime. Using backup and recovery, you would have to shut down the original database after backing it up until the new database is successfully recovered. This is particularly critical if you are moving a tenant database. System replication is also a more convenient method because you don't need to move files between the different systems.

To copy or move a tenant database within the same system, we recommend using backup and recovery.

SAP HANA Database Lifecycle Manager (HDBLCM)

To copy or clone an entire system, use the SAP HANA database lifecycle manager (HDBLCM) as described in the *SAP HANA Platform Lifecycle Management* section of the *SAP HANA Administration Guide*.

Related Information

SAP Note 3316268
SAP Note 2175664
Security of the Copy and Move Process [page 681]
Copy and Move Process [page 677]
Copy or Clone an SAP HANA System [page 669]
Default Host Names and Virtual Host Names [page 744]
Host Name Resolution for SQL Client Communication [page 750]
Configure Host-Independent Tenant Addresses [page 107]
Connections for Tenant Databases
Server-Side Secure Stores [page 501]
Using the LSS with an External Key Management System [page 508]

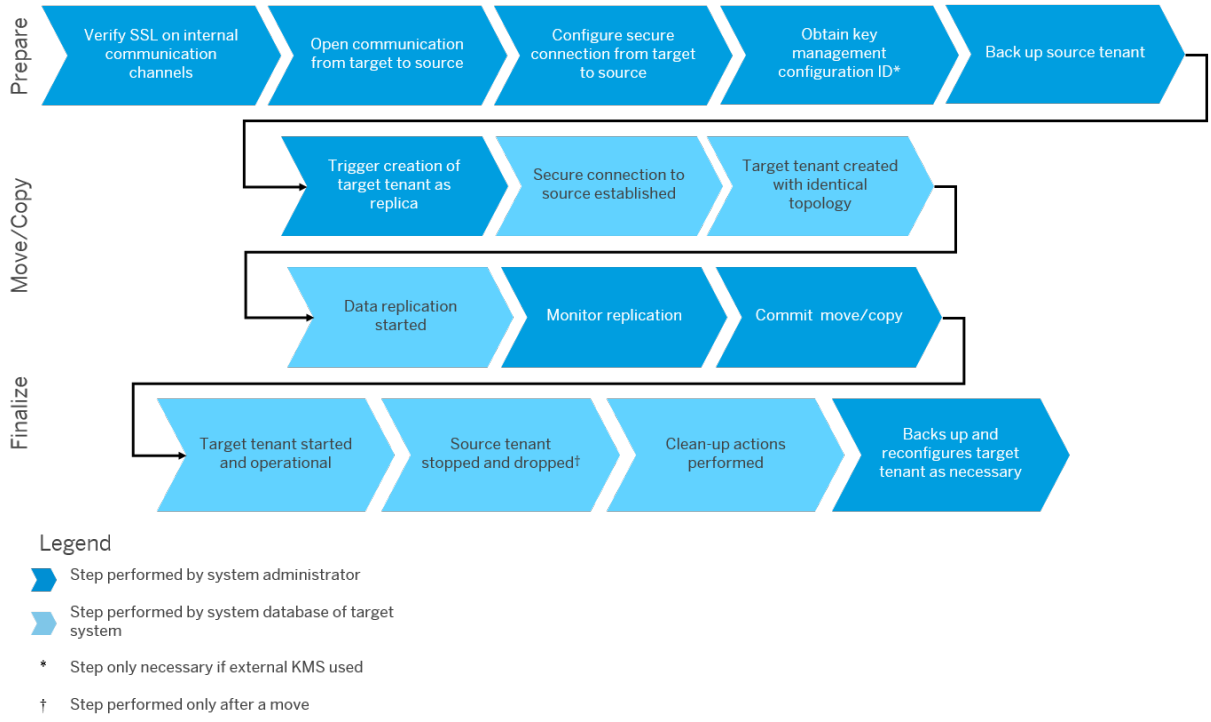
11.1.2.1 Copy and Move Process

Understand the stages and steps involved in copying and moving a tenant database from a source system to a target system using SAP HANA system replication.

Overview

The process of copying or moving a tenant database is driven entirely by the target system.

The following figure shows the stages involved, as well as who performs the individual steps in each stage: the system administrator or the target system. Each step is then described in more detail.



Tenant Move/Copy Process Flow

Prepare

Who?	Does What?	Where?
System administrator	Verifies that TLS/SSL is enabled on internal communication channels	System database of source and target system

Who?	Does What?	Where?
	Configures secure connection from target system to source system by: 1. Creating a certificate collection with the purpose `DATABASE REPLICATION` and adding the root certificate of the source system to the new collection This will allow trust to be established between the system databases of the target and source system for external communication via SQL. 2. Creating a credential in the target system to enable authenticated access from the target system to the source system	System database of target system
	Opens communication from the target system to the source system by enabling source system services to listen on all network interfaces	System database of source system
	Creates a credential to enable authenticated access to the source system for the purpose of copying or moving a tenant database	System database of target system
	If necessary, obtains a valid key management configuration ID (other than `DEFAULT`) from the key administrator (that is a user with system privilege `ENCRYPTION ROOT KEY ADMIN`). The key administrator may create a dedicated configuration for the new database in the source system before the copy or move and provide the ID of this configuration to the system administrator. **Note** This step is only relevant if the local secure store (LSS) is being used in conjunction with an external key management system (KMS).	System database of source system
System administrator	Backs up the source tenant database	System database of source system

For more information, see *Preparing to Copy or Move a Tenant Database*.

Copy and Move

Who?	Does What?	Where?
System administrator	Triggers the creation of the tenant database as a replica of the source tenant database by executing the SQL statement `CREATE DATABASE AS REPLICA`	System database of target system

Who?	Does What?	Where?
System database of target system	Establishes a secure connection to the system database using the stored credentials created above For subsequent secure communication between the systems, a set of public and private key pairs and public-key certificates is generated in the source system database. These will be used to secure communication from the target system database to the source system database, and from the target tenant database to the source tenant database. The generated key pairs and certificates are imported into two newly created certificate collections in the target system database.	From target system to source system
	Creates a new tenant database with the same topology as the tenant database in the source system	Target system
	Initiates replication of data between the services in the source tenant database and the corresponding services in the target database	From source tenant database to target tenant database
System administrator	Monitors the progress of data replication in system view `SYS_DATABASES.M_DATABASE_REPLICAS`	System database of target system or source system
	Once the replication status is `ACTIVE` (indicating that all data has been transferred), commits the copy by executing the SQL statement `ALTER DATABASE FINALIZE REPLICA` In the case of a move, the administrator indicates that the source tenant database is to be dropped: `DROP SOURCE DATABASE`.	System database of the target system

For more information, see *Copy a Tenant Database to Another System* and *Move a Tenant Database to Another System*.

Finalize

Who?	Does What?	Where?
System database in target system	Starts the target tenant database	Target system
	If the source tenant database is being moved to the target system, stops and drops the source tenant database	Source system

Who?	Does What?	Where?
	Performs clean-up operations: • Deletes any cross-database dependencies to the original tenant database in other tenant databases of the source system (**move only**) • Deletes any remote identity dependencies of users in the new tenant database in the target tenant database (**copy and move**) • Generates a new root key used for data volume encryption and re-encrypts data if data volume encryption is enabled (**copy and move**)	Source system and tenant database in target system
System administrator	Performs manual post-copy or post-move steps: • Back up the target tenant database This is only necessary after a copy since the new tenant database does not have a backup history and cannot be recovered. After a move, the new tenant database has the backup history of the original tenant database and can be recovered if data and log backups of the source system are at a location accessible to the target system. • Reverse preparatory steps required to secure the copy process • If necessary, reconfigure parameters in *.ini files with tenant-database-specific values • If necessary, reconfigure cross-database access	System database of the target system
Key administrator	If necessary, changes the key management configuration. **Note** This step is only relevant if the local secure store (LSS) is being used in conjunction with an external key management system (KMS).	System database of the target system

Cancel

Who?	Does What?	Where?
System administrator	Cancels the creation of the tenant database as a replica of the source tenant database by executing the SQL statement `DROP DATABASE <database_name>`	System database of target system
System database of target system	Drops the target tenant database	Target system
	Performs clean-up operations	Target system

Related Information

Security of the Copy and Move Process [page 681]
Preparing to Copy or Move a Tenant Database [page 684]
Copy a Tenant Database to Another System [page 693]
Move a Tenant Database to Another System [page 695]
Create and Manage a Key Management Configuration [page 510]

11.1.2.2 Security of the Copy and Move Process

Copying or moving a tenant database from one system to another is a secure end-to-end process.

Secure Network Communication

The copy and move process ensures end-to-end data encryption and host authentication on the basis of X.509 client certificates. Dedicated certificates and trust stores (referred to as certificate collections) are created as part of the copy or move process for the purpose of that specific copy or move. Certificates and certificate collections are stored directly in the system databases as database objects.

> **Note**
>
> If secure network communication is not required, it can be disabled. As a result, a copy or move of a tenant database can be initiated without the exchange of certificates. For more information, see *Disable Secure Network Communication*.

For more information about in-database certificate management, see the *SAP HANA Security Guide*.

Authorization and Authentication

The copy and move process is triggered from the system database of the target system by a system administrator. To be able to execute the copy or move statements, the administrator user requires the system privilege `DATABASE ADMIN`.

To be able to establish a connection to the system database of the source system, the target system must be authenticated on the source system. This is achieved through the creation of a credential in the secure internal credential store of the system database of the target system. The required credential must be created manually by an administrator in the system database of the target system before the copy or move is started. For more information about the secure internal credential store, see the *SAP HANA Security Guide*.

Encryption Key Handling

To protect data saved to disk from unauthorized access at operating system level, the SAP HANA database supports data encryption in the persistence layer for data and log volumes, and data and log backups. An internal data encryption service is also available to applications requiring data encryption. The instance secure store in the file system (SSFS) or the local secure store (LSS) is used to protect the root keys for these encryption services.

When a tenant database is **copied**, all encryption root keys in the tenant database in the target system are different to those in the tenant database of the source system.

When a tenant database is **moved**, all encryption root keys in the tenant database in the target system – with the exception of the root key used for data volume encryption – are the same as those in the source tenant database.

If the source system is using the LSS to protect root keys and the LSS is connected to an external key management system, then the target system must also use the LSS and the configuration ID of a key management configuration must be provided to initiate the copy or move process. Only the key administrator in the source system (user with `ENCRYPTION ROOT KEY ADMIN` system privilege) can obtain this information from the system view `SYS.KEY_MANAGEMENT_CONFIGURATIONS`. This key management configuration is replicated to the target system and is initially used in the copied or moved database. The key administrator of the source system may prepare a dedicated configuration for the new database before the copy or move, or the key administrator of the target system can change the replicated configuration after the copy or move.

Cross-Database Dependencies

If cross-database access is enabled in the original tenant database, some configured dependencies are automatically deleted to ensure no unauthorized communication paths or user mappings can be exploited in the copied or moved tenant database.

Permitted Communication Paths

Part of the configuration of cross-database access involves specifying which tenant databases may communicate with each other and in which direction.

After a tenant database is moved to another system, it is deleted in the source system. However, communication paths referencing it will still exist in one or more of the other tenant databases in the source system. When the move operation is finalized, all such references to the original database are **automatically** deleted in other tenant databases of the source system.

Communication paths configured in the tenant database in the target system must manually be reconfigured after a move or copy.

User Mappings

Another aspect of cross-database access configuration is the mapping of users in one tenant database to users in another tenant database using remote identities.

After a tenant database is moved or copied to another system, some of its database users may still be associated as remote identities for users in other databases in the source system. When the move or copy operation is finalized, all remote identity information of users in the new tenant database is **automatically** deleted.

New user mappings must be manually configured in the tenant database in the target system.

Related Information

Disable Secure Network Communication [page 683]
Using the LSS with an External Key Management System [page 508]

11.1.2.2.1 Disable Secure Network Communication

During the copy and move process, data is replicated via a secure (TLS/SSL) network connection by default. If this is not necessary in your scenario, you can disable this requirement.

Prerequisites

- You have the system privilege INIFILE ADMIN.
- Neither the source system or target system is configured for high isolation (the value of the `database_isolation` in the `[multidb]` section of the `global.ini` file is **`low`**) .
- The value of parameter `ssl` in the `[communication]` section of the `global.ini` file is set to **`off`**.
- External communication is not configured for TLS/SSL in either the source system or target system.

Procedure

Using for example the SAP HANA cockpit or `ALTER SYSTEM ALTER CONFIGURATION` statement, set the parameter `[multidb]enforce_ssl_database_replication` in the `global.ini` file to **`false`**.

Related Information

Configuring SAP HANA System Properties (INI Files) [page 126]
Modify a System Property in SAP HANA Cockpit

11.1.2.3 Preparing to Copy or Move a Tenant Database

Before you copy or move a tenant database to another system, you must perform several steps. These are primarily to enable the systems to communicate with each other securely.

Context

> ⓘ Note
>
> During the copy and move process, data is replicated via a secure (TLS/SSL) network connection by default. If you do not require a secure network connection and have disabled this feature, you can skip the first two steps: *Verify TLS/SSL Configuration of Internal Communication Channels* and *Set Up Trust Relationship Between Target and Source Systems*. For more information, see *Disable Secure Network Communication*.

1. Verify TLS/SSL Configuration of Internal Communication Channels [page 685]
   In both the source system and the target system, verify that TLS/SSL is enabled on internal communication channels on the basis of the system public key infrastructure (system PKI).
2. Set Up Trust Relationship Between Target and Source Systems [page 686]
   Create a certificate collection in the system database of the target system and add either the public-key certificate of the system database of source system, or the root certificate of the source system. This certificate is used to secure communication between the systems via external SQL connections.
3. Open Communication From Target to Source System [page 688]
   Open communication from the target system to the source system by enabling services in the source system to listen on all network interfaces.
4. Create Credential for Authenticated Access to Source System [page 690]
   Create a credential to enable authenticated access to the source system for the purpose of copying or moving a tenant database.
5. Obtain Key Management Configuration ID [page 691]
   In the system database of the source system, obtain the ID of key management configuration to be replicated to the target system.
6. Back Up Tenant Database [page 692]
   Back up the tenant database that will be copied or moved.

Related Information

Disable Secure Network Communication [page 683]

11.1.2.3.1 Verify TLS/SSL Configuration of Internal Communication Channels

In both the source system and the target system, verify that TLS/SSL is enabled on internal communication channels on the basis of the system public key infrastructure (system PKI).

Prerequisites

You have a user in the system database of both systems with the system privilege `INIFILE ADMIN`.

Context

During the copy and move process, data is replicated via a secure (TLS/SSL) network connection by default. If you do not require a secure network connection and have disabled this feature, you can skip this step. For more information, see *Disable Secure Network Communication*.

Procedure

> → Remember
>
> This step must be performed in both the source system and the target system.

1. Verify the `SYSTEM` layer values of the following parameters in the `global.ini` file:

   - `[communication] ssl` must be set to **`systempki`**
   - `[system_replication_communication] enable_ssl` must be set to **`on`**

   You can verify (and change) these settings using the SAP HANA cockpit or alternatively, you can configure them by executing the following SQL statements:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'SYSTEM') SET
   ( 'communication', 'ssl') = 'systempki' WITH RECONFIGURE;
   ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'SYSTEM') SET
   ( 'system_replication_communication', 'enable_ssl') = 'on' WITH RECONFIGURE;
   ```

2. Restart the system.

Task overview: Preparing to Copy or Move a Tenant Database [page 684]

Next task: Set Up Trust Relationship Between Target and Source Systems [page 686]

Related Information

TLS/SSL Configuration on the SAP HANA Server
Disable Secure Network Communication [page 683]
Configuring SAP HANA System Properties (INI Files) [page 126]
Modify a System Property in SAP HANA Cockpit

11.1.2.3.2 Set Up Trust Relationship Between Target and Source Systems

Create a certificate collection in the system database of the target system and add either the public-key certificate of the system database of source system, or the root certificate of the source system. This certificate is used to secure communication between the systems via external SQL connections.

Prerequisites

- You have a user in the system database of the target system with the system privileges CERTIFICATE ADMIN, TRUST ADMIN, and DATABASE ADMIN.
- You have a copy of the `extract_certificates.py` python script. The python script file must be accessible to the `<sid>adm` user. You can find the script attached to *SAP Note 2175664*.
- You have the public-key certificate of the system database of the source system (or the root certificate of the source system) used for external communication.
  If this certificate does not already exist, you can create it using the SAPGENPSE tool or the SAP Web Dispatcher administration tool, both of which are delivered with SAP HANA. The certificate must be imported into the source system.

  > ⚠ Caution
  >
  > By default, SAP HANA allows encrypted communication for all exposed interfaces leveraging self-signed certificates. Although self-signed certificates allow communication encryption, full communication security can only be reached leveraging certificates signed by a Certificate Authority (CA).

  If the certificate does exist, its location depends on how you manage certificates in your system. Certificates stored in database (recommended) are contained in the certificate store. The required certificate is assigned to the collection with purpose *SSL*. Certificates stored in the file system are contained in tenant database-specific personal security environments or PSEs (default `$SECUDIR/sapsrv.pse`).
  For more information, see *TLS/SSL Configuration on the SAP HANA Server* in the SAP HANA Security Guide and *Managing Client Certificates* in the SAP HANA Administration Guide.

Context

During the copy and move process, data is replicated via a secure (TLS/SSL) network connection by default. If you do not require a secure network connection and have disabled this feature, you can skip this step. For more information, see *Disable Secure Network Communication*.

> **Note**
>
> If you already have a CA-signed certificate, you can skip steps 1 through 3.

Procedure

1. Create a personal security environment (PSE) file using the SAPGENPSE tool.

   ```
   sapgenpse gen_pse -p <path>/<file name>.pse -x "" -noreq "CN=<FQDN of source host>"
   ```

   > **Example**
   >
   > ```
   > sapgenpse gen_pse -p foo.pse -x "" -noreq "CN=sourcehost.domain"
   > ```

2. Extract the generated private key and the self-signed certificate from the PSE file using the `extract_certificates.py` script.

   ```
   python <path to script>/extract_certificates.py -p <file name>.pse
   ```

   The script will print a list of one or more SQL statements that can be transferred to an SQL console using copy and paste.

3. In the system database of the source system, create a certificate collection and set its purpose to `SSL`. You can choose any name for the certificate collection.

   You can do this using the *Certificate Collections* app of the SAP HANA cockpit or by executing the following SQL statements:

   ```
   CREATE PSE <collection name>;
   ALTER PSE <collection name> SET OWN CERTIFICATE '<private key and certificate>';
   SET PSE <collection name> PURPOSE SSL;
   ```

   > **Tip**
   >
   > You can generate the `ALTER PSE` SQL statement using the `extract_certificates.py` script.

4. In the system database of the target system, create a certificate collection and set its purpose to `DATABASE REPLICATION`. You can choose any name for the certificate collection.

   You can do this using the *Certificate Collections* app of the SAP HANA cockpit or by executing the following SQL statements:

   ```
   CREATE PSE <collection name>;
   SET PSE <collection name> PURPOSE DATABASE REPLICATION;
   ```

5. If not already in the certificate store, import the public-key certificate of the system database of the source system (or the root certificate of the source system) into the certificate store of the target system.

   You can do this using the *Certificate Store* app of the SAP HANA cockpit or by executing the following SQL statement:

   ```
   CREATE CERTIFICATE FROM '<certificate content>';
   ```

6. Add the system database certificate (or root certificate) to the new collection.

   You can do this using the *Certificate Collections* app of the SAP HANA cockpit or by executing the following SQL statement:

   ```
   ALTER PSE <collection name> ADD CERTIFICATE <certificate id>;
   ```

   → Tip

   You will find the certificate ID in the system view `SYS.CERTIFICATES`.

   ```
   SELECT * FROM SYS.CERTIFICATES;
   ```

Task overview: Preparing to Copy or Move a Tenant Database [page 684]

Previous task: Verify TLS/SSL Configuration of Internal Communication Channels [page 685]

Next task: Open Communication From Target to Source System [page 688]

Related Information

TLS/SSL Configuration on the SAP HANA Server
SAP Note 2175664 - Migration of file system based X.509 certificate stores to in-database certificate stores
Disable Secure Network Communication [page 683]

11.1.2.3.3 Open Communication From Target to Source System

Open communication from the target system to the source system by enabling services in the source system to listen on all network interfaces.

Prerequisites

You have the credentials of operating system administrator `<sid>adm` for the source system.

Context

Use the SAP HANA database lifecycle manager (HDBLCM) to configure inter-service communication so that the services of the target system can listen on all available network interfaces.

> **Note**
>
> It is only necessary to perform this step in the source system. However, if you later want to be able to monitor the progress of the copy or move operation from the source system, you can also do it in the target system.

Procedure

> **Note**
>
> The following procedure describes how to do this using the Web user interface. For more information about using the command-line interface or graphical user interface of the SAP HANA database lifecycle manager, see the *SAP HANA Administration Guide*.
>
> Instead of using the SAP HANA database lifecycle manager (HDBLCM), you can execute the following SQL statement:
>
> ```
> ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'SYSTEM') SET ( 'communication',
> 'listeninterface') = '.global' WITH RECONFIGURE;
> ```

1. Open the SAP HANA database lifecycle manager by entering the following URL in a browser:
   `https://<host>:1129/lmsl/HDBLCM/<sid>/index.html`
2. Click the tile *Configure Inter-Service Communication*.
3. When prompted, enter the password of the `<sid>adm` user.
4. Select the setting *global*.
5. Click *Run* to apply the new setting.
6. Close the application and log out.
7. Restart the SAP HANA database.

Task overview: Preparing to Copy or Move a Tenant Database [page 684]

Previous task: Set Up Trust Relationship Between Target and Source Systems [page 686]

Next task: Create Credential for Authenticated Access to Source System [page 690]

Related Information

Internal Host Name Resolution [page 747]

11.1.2.3.4 Create Credential for Authenticated Access to Source System

Create a credential to enable authenticated access to the source system for the purpose of copying or moving a tenant database.

Prerequisites

- You have a user in the system database of the source system with the system privilege `EXECUTE` on the stored procedure `GET_DATABASE_REPLICATION_PSE`.
- You have a user in the system database of the target system with the system privilege `CREDENTIAL ADMIN`.

Context

You create a credential in the secure internal credential store of the system database of target system.

The credential store is used in SAP HANA to securely store credentials required for outbound connections. For more information about the secure internal credential store, see the *SAP HANA Security Guide*.

Procedure

In the system database of the target system, create a credential by executing the following SQL statement:

```
CREATE CREDENTIAL FOR COMPONENT 'DATABASE_REPLICATION' PURPOSE
'<host:internal_port_of_system_DB_of_source_system>'
TYPE 'PASSWORD' USING
'user="<user_in_system_DB_of_source_system_with_DATABASE_ADMIN_and_INIFILE_ADMIN>
";password="<password>"'
```

The values required for each parameter are as follows:

Parameter	Required Value
COMPONENT	DATABASE_REPLICATION
PURPOSE	Host name and internal port number of the system database of the source system
TYPE	PASSWORD
USING	User name of a user in the system database of the source system with the system privileges DATABASE ADMIN and INIFILE ADMIN

Sample Code

```
CREATE CREDENTIAL FOR COMPONENT 'DATABASE_REPLICATION'
PURPOSE 'host123456.acme.corp:30001' TYPE 'PASSWORD' USING
'user="DATABASE_ADMINISTRATOR";password="<password>"'
```

Task overview: Preparing to Copy or Move a Tenant Database [page 684]

Previous task: Open Communication From Target to Source System [page 688]

Next task: Obtain Key Management Configuration ID [page 691]

Related Information

SAP HANA Security Guide

11.1.2.3.5 Obtain Key Management Configuration ID

In the system database of the source system, obtain the ID of key management configuration to be replicated to the target system.

Prerequisites

- The source system uses the local secure store (LSS) in conjunction with an external key management system (KMS) to protect encryption root keys.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Context

When moving or copying a tenant database, a key management configuration ID (other than `DEFAULT`) must be provided if the source system uses the LSS with a KMS. Since only a key administrator can obtain the key management configuration ID, this means that the key administrator must permit the transfer of a key management configuration to the target system.

This key management configuration is replicated to the target system and is initially used in the copied or moved database. The key administrator of the source system may create a dedicated configuration for the new database already before the copy or move, or the key administrator of the target system can change the replicated configuration after the copy or move.

Procedure

Get the required configuration ID by querying system view `SYS.KEY_MANAGEMENT_CONFIGURATIONS`.

Task overview: Preparing to Copy or Move a Tenant Database [page 684]

Previous task: Create Credential for Authenticated Access to Source System [page 690]

Next task: Back Up Tenant Database [page 692]

Related Information

Create and Manage a Key Management Configuration [page 510]
KEY_MANAGEMENT_CONFIGURATIONS System View

11.1.2.3.6 Back Up Tenant Database

Back up the tenant database that will be copied or moved.

Context

You can back up the tenant database from the system database or from the tenant database directly. For more information, see *Creating Backups* in the *SAP HANA Administration Guide*.

Task overview: Preparing to Copy or Move a Tenant Database [page 684]

Previous task: Obtain Key Management Configuration ID [page 691]

Related Information

Creating Backups

11.1.2.4 Copy a Tenant Database to Another System

Copy a tenant database from one SAP HANA system to another. The new copied tenant database runs as a separate, independent database.

Prerequisites

- All general system prerequisites are fulfilled. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- All preparatory steps have been completed. For more information, see *Preparing to Copy or Move a Tenant Database*.
- You have a user in the system database of the target system with the system privilege `DATABASE ADMIN` and `CATALOG READ`.

Procedure

1. Create a tenant database in the target system as a copy of the original tenant database in the source system.

   You do this by executing the `CREATE DATABASE` statement:

   Code Syntax

   ```
   CREATE DATABASE <target_database_name> [ AT [ LOCATION ]
   '<target_hostname>[:<port_number_master_indexserver> ] ' ]
    { ADD '<servicetype>' [ AT
   [ LOCATION ] '<target_hostname>[:<port_number_service>
   ]@<source_hostname>:<port_number_service>' ] }...
    { AS REPLICA OF [ <source_database_name> ] AT [ LOCATION ]
   '<source_hostname>[:<port_number_systemdb> ]' KEY MANAGEMENT CONFIGURATION
   <id> }
    [ OS USER '<username>' OS GROUP '<groupname>' ]
    [ NO START ]
    [ <restart_mode> RESTART ]
   ```

   Note

   - As the location of the source tenant database, you specify the host name and port number for internal communication of the **system database** of the source system.
   - The source and target host names must match the host names specified during installation.
   - If you enabled SSL, the host name must match the common name (CN) specified in the public-key certificate of the system database of source system.
   - If you specify a service list, the number and type of services must match the source database.
   - If your systems are configured for high isolation, you specify a valid OS user or OS group of the tenant database.
   - Fallback snapshots are not copied to the target system.

Sample Code

```
CREATE DATABASE TARGET_DATABASE AS REPLICA OF SOURCE_DATABASE AT
'host123456.acme.corp:30001';
```

With the execution of this statement, the system database of the target system does the following:

- Establishes a secure connection to the system database of the source system
- Creates a tenant database with the same topology as the tenant database in the source system
- Starts replicating data between the services in the source tenant database and the corresponding services in the target database

2. Monitor replication progress of data replication from the original tenant database to the new tenant database.

   Use the system view `SYS_DATABASES.M_DATABASE_REPLICAS` to monitor the status of data replication in the system database of the target system or `SYS.M_DATABASES` to monitor directly in the new tenant database.

   The current status of replication is shown in the field `REPLICATION_STATUS`. The value is aggregated across all individual services of the system, e.g. the system global status is only `ACTIVE`, if all individual services have replication status `ACTIVE`.

   The following replication statuses are possible:

Status	Description
UNKNOWN	The secondary system did not connect to the primary system since the last restart of the primary system.
INITIALIZING	Data transfer is initialized. In this state, the secondary system cannot be used.
SYNCING	The secondary system is syncing again (e.g. after a temporary connection loss or restart of the secondary system).
ACTIVE	Initialization or sync with the primary system is complete and the secondary system is continuously replicating. If a crash occurs, no data is lost in `SYNC` mode.
ERROR	A connection error occurred (details can be found in REPLICATION_STATUS_DETAILS).

The view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` provides detailed information about the replication process at the service level.

→ Tip

If the replication status is `ERROR`, use system view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` to investigate further.

If you cannot create a replica because the source database is still in status "REPLICATING" even though the target database is already dropped, execute the statement `ALTER DATABASE <database_name> CANCEL REPLICA` on the source system to clean up the system before reattempting replication.

Note

You can also monitor from the source system if you opened communication between the systems in both directions. For more information, see *Open Communication From Target to Source System*.

3. When replication status is `ACTIVE` (indicating that the new tenant database is in sync with the original tenant database), stop replication and finalize the copy by executing the following statement in the system database of the target system:

   ```
   ALTER DATABASE <new_database_name> FINALIZE REPLICA
   ```

   With the execution of the above statement, the system database of the target system performs the following actions in the new tenant database:

   - Starts the new tenant database
   - Changes the root key for data volume encryption and re-encrypts data in the new database if data volume encryption is enabled
   - Deletes remote identities of database users if the original tenant database was configured for cross-database access

Next Steps

Perform the required manual post-move tasks.

Related Information

Copying and Moving Tenant Databases [page 671]
Preparing to Copy or Move a Tenant Database [page 684]
Perform Manual Post-Copy/Move Tasks [page 703]
CREATE DATABASE Statement (Tenant Database Management)
M_DATABASE_REPLICAS System View

11.1.2.5 Move a Tenant Database to Another System

Move a tenant database in one SAP HANA system to another. After a move, the original tenant database is deleted and the new tenant database takes over.

Prerequisites

- All general system prerequisites are fulfilled. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- All preparatory steps have been completed. For more information, see *Preparing to Copy or Move a Tenant Database*.
- You have a user in the system database of the target system with the system privilege `DATABASE ADMIN` and `CATALOG READ`.

Procedure

1. Create a tenant database in the target system as a copy of the original tenant database in the source system.

   You do this by executing the `CREATE DATABASE` statement:

   Code Syntax

   ```
   CREATE DATABASE <target_database_name> [ AT [ LOCATION ]
   '<target_hostname>[:<port_number_master_indexserver> ] ' ]
    { ADD '<servicetype>' [ AT
   [ LOCATION ] '<target_hostname>[:<port_number_service>
   ]@<source_hostname>:<port_number_service>' ] }...
    { AS REPLICA OF [ <source_database_name> ] AT [ LOCATION ]
   '<source_hostname>[:<port_number_systemdb> ]' KEY MANAGEMENT CONFIGURATION
   <id> }
    [ OS USER '<username>' OS GROUP '<groupname>' ]
    [ NO START ]
    [ <restart_mode> RESTART ]
   ```

   Note

   - As the location of the source tenant database, you specify the host name and port number for internal communication of the **system database** of the source system.
   - The source and target host names must match the host names specified during installation.
   - If you enabled SSL, the host name must match the common name (CN) specified in the public-key certificate of the system database of source system.
   - If you specify a service list, the number and type of services must match the source database.
   - If your systems are configured for high isolation, you specify a valid OS user or OS group of the tenant database.
   - Fallback snapshots are not moved to the target system.

   Sample Code

   ```
   CREATE DATABASE TARGET_DATABASE AS REPLICA OF SOURCE_DATABASE AT
   'host123456.acme.corp:30001';
   ```

   With the execution of this statement, the system database of the target system does the following:

   - Establishes a secure connection to the system database of the source system
   - Creates a tenant database with the same topology as the tenant database in the source system
   - Starts replicating data between the services in the source tenant database and the corresponding services in the target database

2. Monitor replication progress of data replication from the original tenant database to the new tenant database.

   Use the system view `SYS_DATABASES.M_DATABASE_REPLICAS` to monitor the status of data replication in the system database of the target system or `SYS.M_DATABASES` to monitor directly in the new tenant database.

The current status of replication is shown in the field `REPLICATION_STATUS`. The value is aggregated across all individual services of the system, e.g. the system global status is only `ACTIVE`, if all individual services have replication status `ACTIVE`.

The following replication statuses are possible:

Status	Description
UNKNOWN	The secondary system did not connect to the primary system since the last restart of the primary system.
INITIALIZING	Data transfer is initialized. In this state, the secondary system cannot be used.
SYNCING	The secondary system is syncing again (e.g. after a temporary connection loss or restart of the secondary system).
ACTIVE	Initialization or sync with the primary system is complete and the secondary system is continuously replicating. If a crash occurs, no data is lost in `SYNC` mode.
ERROR	A connection error occurred (details can be found in REPLICATION_STATUS_DETAILS).

The view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` provides detailed information about the replication process at the service level.

> **→ Tip**
>
> If the replication status is `ERROR`, use system view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` to investigate further.
>
> If you cannot create a replica because the source database is still in status "REPLICATING" even though the target database is already dropped, execute the statement `ALTER DATABASE <database_name> CANCEL REPLICA` on the source system to clean up the system before reattempting replication.

> **ⓘ Note**
>
> You can also monitor from the source system if you opened communication between the systems in both directions. For more information, see *Open Communication From Target to Source System*.

3. When replication status is `ACTIVE` (indicating that the new tenant database is in sync with the original tenant database), stop replication and finalize the move by executing the following statement in the system database of the target system:

```
ALTER DATABASE <new_database_name> FINALIZE REPLICA DROP SOURCE DATABASE
```

With the execution of the above statement, the system database of the target system performs the following actions:

- Starts the new tenant database
- Changes the root key for data volume encryption and re-encrypts data in the new database if data volume encryption is enabled
- Drops the original tenant database in the source system

> **ⓘ Note**
>
> To ensure that the new tenant database can be recovered to the most recent consistent state after the move, data backups are not deleted as part of the move process. This is important in the event

> that a backup is created in the original tenant database after replication has finished but before the original database is finally deleted.

- Deletes any communication paths configured for cross-database access that reference the original tenant database in the other tenant databases of the source system

Next Steps

Perform the required manual post-move tasks.

Related Information

Copying and Moving Tenant Databases [page 671]
Preparing to Copy or Move a Tenant Database [page 684]
Perform Manual Post-Copy/Move Tasks [page 703]
CREATE DATABASE Statement (Tenant Database Management)
M_DATABASE_REPLICAS System View

11.1.2.6 Copy a Tenant Database to Another Host within the same System

Copy a tenant database from one host to another within the same system.

Prerequisites

- All general system prerequisites are fulfilled. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- All preparatory steps have been completed. For more information, see *Preparing to Copy or Move a Tenant Database*.
- You have a user in the system database with the system privilege `DATABASE ADMIN` and `CATALOG READ`.

Procedure

1. Create a tenant database as a copy of the original tenant database.

You do this by executing the `CREATE DATABASE` statement:

Code Syntax

```
CREATE DATABASE <target_database_name> [ AT [ LOCATION ]
'<target_hostname>[:<port_number_master_indexserver> ] ' ]
 { ADD '<servicetype>' [ AT
[ LOCATION ] '<target_hostname>[:<port_number_service>
]@<source_hostname>:<port_number_service>' ] }...
 { AS REPLICA OF [ <source_database_name> ] AT [ LOCATION ]
'<source_hostname>[:<port_number_systemdb> ]' KEY MANAGEMENT CONFIGURATION
<id> }
 [ OS USER '<username>' OS GROUP '<groupname>' ]
 [ NO START ]
 [ <restart_mode> RESTART ]
```

Note

- As the location of the source tenant database, you specify the host name and port number for internal communication of the **system database**.
- The source and target host names must match the host names specified during installation.
- If you do not specify a target host name, the host with the lowest number of services will be used.
- If you enabled SSL, the host name must match the common name (CN) specified in the public-key certificate of the system database.
- If you specify a service list, the number and type of services must match the source database.
- If your systems are configured for high isolation, you specify a valid OS user or OS group of the tenant database.
- Fallback snapshots are not copied.

Sample Code

```
CREATE DATABASE TARGET_DATABASE AS REPLICA OF SOURCE_DATABASE AT
'host123456.acme.corp:30101';
```

With the execution of this statement, the system database does the following:

- Creates a tenant database with the same topology as the source tenant database
- Starts replicating data between the services in the source tenant database and the corresponding services in the target database

2. Monitor replication progress of data replication from the original tenant database to the new tenant database.

   Use the system view `SYS_DATABASES.M_DATABASE_REPLICAS` to monitor the status of data replication in the system database or `SYS.M_DATABASES` to monitor directly in the new tenant database.

   The current status of replication is shown in the field `REPLICATION_STATUS`. The value is aggregated across all individual services of the system, e.g. the system global status is only `ACTIVE`, if all individual services have replication status `ACTIVE`.

The following replication statuses are possible:

Status	Description
UNKNOWN	The secondary system did not connect to the primary system since the last restart of the primary system.
INITIALIZING	Data transfer is initialized. In this state, the secondary system cannot be used.
SYNCING	The secondary system is syncing again (e.g. after a temporary connection loss or restart of the secondary system).
ACTIVE	Initialization or sync with the primary system is complete and the secondary system is continuously replicating. If a crash occurs, no data is lost in `SYNC` mode.
ERROR	A connection error occurred (details can be found in REPLICATION_STATUS_DETAILS).

The view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` provides detailed information about the replication process at the service level.

> → Tip
>
> If the replication status is `ERROR`, use system view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` to investigate further.
>
> If you cannot create a replica because the source database is still in status "REPLICATING" even though the target database is already dropped, execute the statement `ALTER DATABASE <database_name> CANCEL REPLICA` on the source system to clean up the system before reattempting replication.

3. When replication status is `ACTIVE` (indicating that the new tenant database is in sync with the original tenant database), stop replication and finalize the copy by executing the following statement in the system database:

```
ALTER DATABASE <new_database_name> FINALIZE REPLICA
```

With the execution of the above statement, the system database performs the following actions in the new tenant database:

- Starts the new tenant database
- Changes the root key for data volume encryption and re-encrypts data in the new database if data volume encryption is enabled
- Deletes remote identities of database users if the original tenant database was configured for cross-database access

Next Steps

Perform the required manual post-move tasks.

Related Information

Copying and Moving Tenant Databases [page 671]

Preparing to Copy or Move a Tenant Database [page 684]
Perform Manual Post-Copy/Move Tasks [page 703]
CREATE DATABASE Statement (Tenant Database Management)
M_DATABASE_REPLICAS System View

11.1.2.7 Create a Copy of a Tenant Database on the Same Host

Create a copy of a tenant database on the same host.

Prerequisites

- All general system prerequisites are fulfilled. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- All preparatory steps have been completed. For more information, see *Preparing to Copy or Move a Tenant Database*.
- You have a user in the system database with the system privilege `DATABASE ADMIN` and `CATALOG READ`.

Procedure

1. Create a tenant database as a copy of the original tenant database.

   You do this by executing the `CREATE DATABASE` statement:

   Code Syntax

   ```
   CREATE DATABASE <target_database_name>
    { ADD '<servicetype>' [ AT
   [ LOCATION ] '<hostname>[:<port_number_service>
   ]@<hostname>:<port_number_service>' ] }...
    { AS REPLICA OF [ <source_database_name> ] KEY MANAGEMENT CONFIGURATION
   <id> }
    [ OS USER '<username>' OS GROUP '<groupname>' ]
    [ NO START ]
    [ <restart_mode> RESTART ]
   ```

   Note

   - If you enabled SSL, the host name must match the common name (CN) specified in the public-key certificate of the system database.
   - If you specify a service list, the number and type of services must match the source database.
   - If your systems are configured for high isolation, you specify a valid OS user or OS group of the tenant database.

- Fallback snapshots are not copied.

{·} Sample Code

```
CREATE DATABASE TARGET_DATABASE AS REPLICA OF SOURCE_DATABASE;
```

With the execution of this statement, the system database does the following:

- Creates a tenant database with the same topology as the source tenant database
- Starts replicating data between the services in the source tenant database and the corresponding services in the target database

2. Monitor replication progress of data replication from the original tenant database to the new tenant database.

   Use the system view `SYS_DATABASES.M_DATABASE_REPLICAS` to monitor the status of data replication in the system database or `SYS.M_DATABASES` to monitor directly in the new tenant database.

   The current status of replication is shown in the field `REPLICATION_STATUS`. The value is aggregated across all individual services of the system, e.g. the system global status is only `ACTIVE`, if all individual services have replication status `ACTIVE`.

   The following replication statuses are possible:

Status	Description
UNKNOWN	The secondary system did not connect to the primary system since the last restart of the primary system.
INITIALIZING	Data transfer is initialized. In this state, the secondary system cannot be used.
SYNCING	The secondary system is syncing again (e.g. after a temporary connection loss or restart of the secondary system).
ACTIVE	Initialization or sync with the primary system is complete and the secondary system is continuously replicating. If a crash occurs, no data is lost in `SYNC` mode.
ERROR	A connection error occurred (details can be found in REPLICATION_STATUS_DETAILS).

   The view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` provides detailed information about the replication process at the service level.

   → Tip

   If the replication status is `ERROR`, use system view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` to investigate further.

   If you cannot create a replica because the source database is still in status "REPLICATING" even though the target database is already dropped, execute the statement `ALTER DATABASE <database_name> CANCEL REPLICA` on the source system to clean up the system before reattempting replication.

3. When replication status is `ACTIVE` (indicating that the new tenant database is in sync with the original tenant database), stop replication and finalize the copy by executing the following statement in the system database:

```
ALTER DATABASE <new_database_name> FINALIZE REPLICA
```

With the execution of the above statement, the system database performs the following actions in the new tenant database:

- Starts the new tenant database
- Changes the root key for data volume encryption and re-encrypts data in the new database if data volume encryption is enabled
- Deletes remote identities of database users if the original tenant database was configured for cross-database access

Next Steps

Perform the required manual post-move tasks.

Related Information

Copying and Moving Tenant Databases [page 671]
Preparing to Copy or Move a Tenant Database [page 684]
Perform Manual Post-Copy/Move Tasks [page 703]
CREATE DATABASE Statement (Tenant Database Management)
M_DATABASE_REPLICAS System View

11.1.2.8 Perform Manual Post-Copy/Move Tasks

After you have committed the copy or move and the new tenant database is up and running, you must perform several manual tasks.

Procedure

1. Back up the new root keys to a root key backup file (`*.rkb`) in a secure location.

   > ⚠ Caution
   >
   > Store the root key backup file in a safe location. Losing this file may result in the database being unrecoverable.

2. Perform a full data backup of the new tenant database (copy only).
3. Reverse the preparatory steps required to secure the copy or move process:
   - Close network communication from the target system to the source system. See *Open Communication From Target to Source System*.
   - Delete the credential used by the system database of the target system to access the source system. See *Create Credential for Authenticated Access to Source System*.

- Delete the certificate collection with purpose DATA REPLICATION created to secure external communication between the systems. See *Set Up Trust Relationship Between Target and Source Systems*.
- If you created and signed the certificate yourself, delete the PSE file from the file system.

4. If necessary, reconfigure parameters in *.ini files with tenant-database-specific values. See *Configuration Parameters in Tenant Database Systems*.
5. Reconfigure cross-database access, if required. See *Enable and Configure Cross-Database Access*.
6. After configuring cross-database access, rebuild or repair cross-database objects, if required. For improved performance, SAP HANA stores object IDs of remote objects in the catalog persistence. If a tenant database is copied or moved, these remote object IDs are likely to change and must be rebuilt by executing the `CHECK_CATALOG` procedure. The following objects are supported: functions, procedures, synonyms, SQL views, and calculation views.

> **Note**
>
> Only rebuild and repair cross-database objects if you want the original settings to be fully restored. For security reasons, you may not want cross-database objects to be restored on a copied or moved system.

> **Example**
>
> Rebuild all objects with remote dependencies by executing the following procedure:
>
> ```
> CALL CHECK_CATALOG ('REBUILD_REMOTE_DEPENDENCY', null, null, null)
> ```
>
> Repair all objects in `<schema>` by executing the following procedure:
>
> ```
> CALL CHECK_CATALOG ('REPAIR_REMOTE_DEPENDENCY', '<schema>', null, null)
> ```
>
> Repair the object `<schema>.<view>` by executing the following procedure:
>
> ```
> CALL CHECK_CATALOG ('REPAIR_REMOTE_DEPENDENCY', '<schema>', '<view>', null)
> ```

Related Information

Open Communication From Target to Source System [page 688]
Set Up Trust Relationship Between Target and Source Systems [page 686]
Create Credential for Authenticated Access to Source System [page 690]
Enable and Configure Cross-Database Access [page 92]
Database-Specific Configuration Parameters

11.1.3 Copying a System Using System Replication

SAP HANA system replication can be used to create a copy of an SAP HANA database in a quick and simple way.

You can register another SAP HANA database for replication in one of the following two system replication scenarios:

- As a secondary for a standalone SAP HANA database
- As a tier 3 secondary in a tier 2 system replication landscape

System Replication Scenarios

Original Setup	Source Database	Target Database
Standalone SAP HANA Database	Primary	Secondary
Tier 2 System Replication	Tier 2 Secondary	Tier 3 Secondary

After the replication is active and in sync, a takeover to the newly added tier makes the target SAP HANA database runnable with identical data to the source database.

11.1.3.1 Copy a System Using System Replication

You can use SAP HANA system replication to create a copy of an SAP HANA database.

Prerequisites

- This process requires a source SAP HANA database which is to be copied.
- The target host(s) can also be virtual machines, but the installed SAP HANA version on the target must be either the same or higher than the version on the source.

Procedure

1. Prepare the source database for replication with the following command:

   ```
   hdbnsutil -sr_enable [--name=<siteName>]
   ```

2. Register the target database either as a secondary or tier 3 secondary depending on your original setup:

   ```
   hdbnsutil -sr_register --remoteHost=<primary master host> --
   remoteInstance=<primary instance id>
   --replicationMode=[sync|syncmem|async] --name=<sitename>
   --operationMode=[delta_datashipping|logreplay|logreplay_readaccess]
   ```

   The default operation mode is logreplay but you can specify a different mode in the command.

3. Start the newly registered target database.
4. When the system replication is active and in sync, perform a takeover on the target database.
5. After the takeover is done, the target database is running as a copy of the source database.
6. To avoid confusion with the source databases, rename the `<SID>` and change the instance number using the tool hdblcm.

11.1.4 Renaming a System

An SAP HANA system can be renamed by changing the system identifiers, like host names, SID, and instance number. Changing system identifiers can be performed with the SAP HANA database lifecycle manager (HDBLCM).

System Identifiers

System identifiers are required parameters set during SAP HANA system installation. In some cases, it is necessary to change the originally configured system identifiers. All three system identifiers - host name, SID, and instance number - can be changed together or individually from the SAP HANA database lifecycle manager graphical user or command-line interface.

> **Note**
>
> System replication must be disabled before renaming the SAP HANA system. If you need to change system identifiers for a system that is set up for system replication, you must first disable system replication, then change the system identifiers on each host and finally re-enable system replication.

The following options are available for SAP HANA database lifecycle manager in graphical user and command-line interfaces:

Task	Graphical User Interface	Command-Line Interface
Rename an SAP HANA System Host	*Rename the SAP HANA Database System* > *Define Host Properties* > *Edit Host*	`--action=rename_system --hostmap=<old host>=<new host>`
Change the SID of an SAP HANA System	*Rename the SAP HANA Database System* > *Define System Properties* > *Target System ID*	`--action=rename_system --target_sid=<new sid>`
Change the Instance Number of an SAP HANA System	*Rename the SAP HANA Database System* > *Define System Properties* > *Target Instance Number*	`--action=rename_system --number=<new instance number>`

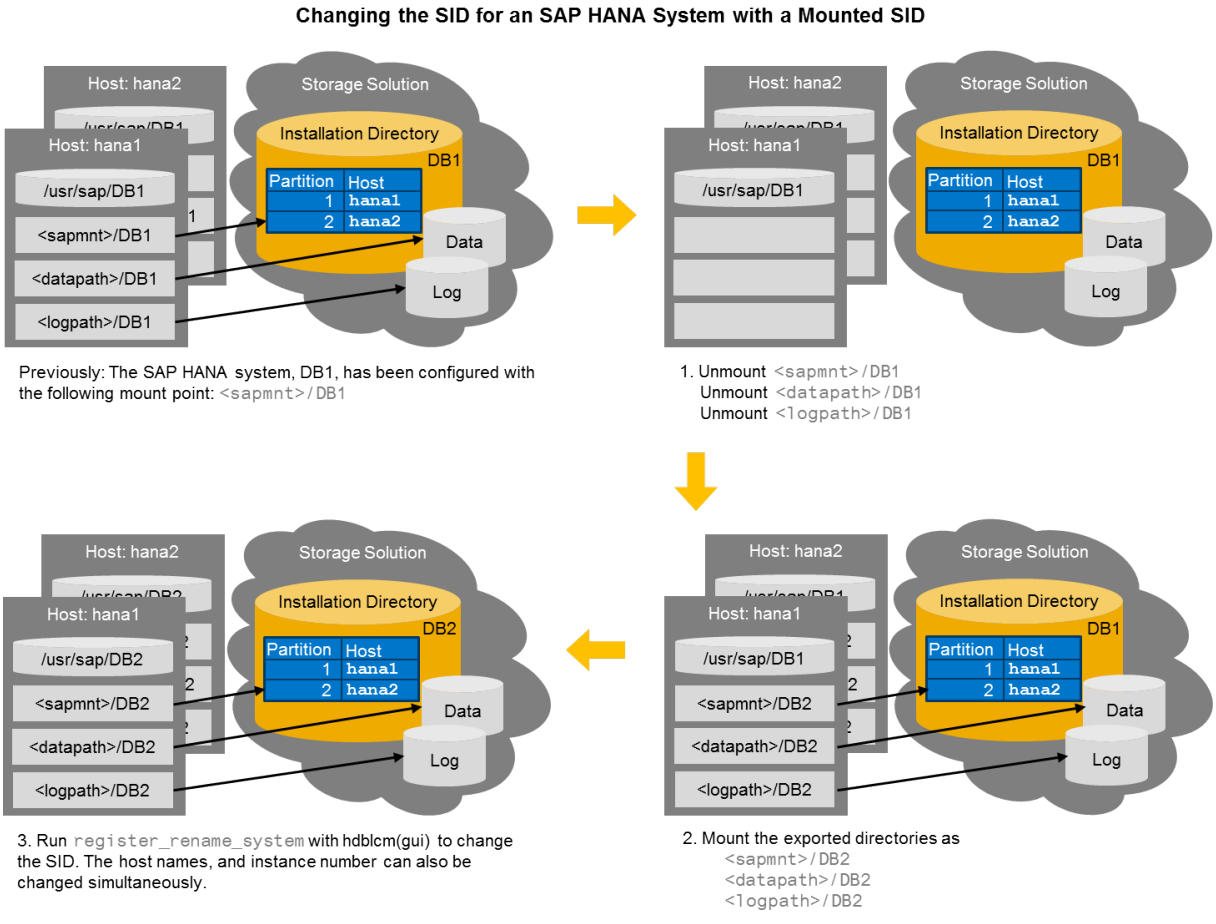
Mounted SID Preparation

If the SID is included in the mount point, and you want to change the SID, you have to take extra preparation steps.

Normally, the installation path is exported and can be shared as `/<sapmnt>`. (The default for `<sapmnt>` is `/hana/shared`) so that several SAP HANA systems are located on the same physical device. However, if you exported a directory only for an individual SAP HANA system, the shared directory (the mount point) is `/<sapmnt>/<SID>`. In this case, you need to create a shared directory with the new target SID before changing the SID of the system.

Example

In the following example, an SAP HANA system with a mounted SID is prepared for SID change:



11.1.4.1 Rename an SAP HANA System Host

You can rename an SAP HANA system host using SAP HANA database lifecycle manager (HDBLCM) resident program on the system which you want to configure.

Prerequisites

- You are logged in as root user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The host you want to rename is either reachable via both the old and new host names or the SAP HANA system is stopped.
- In case you want to change the SID as well, make sure that the target SID does not exist. However, the target operating system administrator (`<sid>adm`) user may exist. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Context

> **Note**
>
> If you rename an SAP HANA system, this usually invalidates the permanent SAP license. A temporary license is installed, and must be replaced within 28 days. For more information, see Related Information.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the rename task:
   - To rename a host using the SAP HANA database lifecycle manager command-line interface:
     - Start the command-line tool interactively:

       ```
       ./hdblcm
       ```

       and enter the index of the `rename_system` action, or

- Start the tool with the `rename_system` action specified:

  ```
  ./hdblcm --action=rename_system --hostmap=<old host>=<new host>
  ```

- To rename a host using the SAP HANA database lifecycle manager graphical user interface:
  1. Start the graphical user interface tool:

     ```
     ./hdblcmgui
     ```

     The SAP HANA database lifecycle manager graphical user interface appears.
  2. Choose *Rename the SAP HANA Database System*.
  3. Select a host and choose *Edit Host...*.
  4. Enter the new host name in the *Target Host Name* field.

3. Define the required parameters.

   For more information about parameters for the `rename_system` action, see the *SAP HANA Server Installation and Update Guide*.

   > **Note**
   >
   > When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the SAP HANA resident HDBLCM directory of the SAP HANA system, execute the following command:
   >
   > ```
   > ./hdblcm --action=rename_system --help
   > ```

4. To continue with the task proceed as follows:
   - In the command line interface: Enter *y*.
   - In the graphical interface:
     1. *Next*.
     2. To execute the configuration task, choose *Rename*. The system displays the configuration progress.
     3. After the configuration task has finished, you can:
        - View the log. To do so, choose *View Log*.
        - Exit the graphical user interface. To do so, choose *Finish*.

Related Information

Managing SAP HANA Licenses [page 137]

SAP HANA Server Installation and Update Guide

tenantmap

11.1.4.2 Change the SID of an SAP HANA System

You can change the SID of an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program on the system which you want to configure.

Prerequisites

- You are logged in as root user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- The target SID must not exist. However, the target operating system administrator (`<SID>`adm) user may exist. Make sure that you have the password of the existing `<SID>`adm user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Context

> **Note**
>
> If you rename an SAP HANA system, this usually invalidates the permanent SAP license. A temporary license is installed, and must be replaced within 28 days.
>
> An SAP HANA system has one SID for the system database and all tenants. Renaming a system changes the SID for the system database and all tenants.

Procedure

1. **In some cases**, the shared directory (mount point) includes the SID. If your mount point includes the SID, create a new shared directory before renaming the host.

   Normally, the installation path (`<sapmnt>`), the data path (`<datapath>`), and the log path (`<logpath>`) are exported and can be shared. However, if you exported shared directories only for an individual SAP HANA system, the mount points are `<sapmnt>/<current SID>`, `<datapath>/<current SID>`, and `<logpath>/<current SID>`. In this case, you need to mount the exported directories as `<sapmnt>/<target SID>`, `<datapath>/<target SID>`, and `<logpath>/<target SID>` before changing the SID of the system.

   > **Note**
   >
   > If SAP HANA Local Secure Store (LSS) is installed, the LSS shared directory (mount point) cannot include the SID. The LSS installation directory (default: `/lss/shared`) must be the shared directory (mount point) and the `<current SID>` subdirectory should not be changed to `<target SID>`.

a. Stop the SAP HANA system.

   To do this, in the SAP Host Agent perform the following operation:

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <instance number> -function StopSystem
   ```

b. Stop the sapstartsrv service by using the following SAP Host Agent operation:

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <instance number> -function
   StopService
   ```

c. Unmount `<sapmnt>/<current SID>`, `<datapath>/<current SID>`, `<logpath>/<current SID>`.

d. Mount the exported directories as `<sapmnt>/<target SID>`, `<datapath>/<target SID>`, `<logpath>/<target SID>`.

2. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.

   > **Note**
   >
   > If the mount point includes the SID, change to the SAP HANA resident HDBLCM directory of the **target SID**.

3. Start the SID change task:

   - To change the SID using the SAP HANA database lifecycle manager command-line interface:
     - Start the command-line tool interactively:

       ```
       ./hdblcm
       ```

       and enter the index of the `rename_system` action, or
     - Start the tool with the `rename_system` action specified:

       ```
       ./hdblcm --action=rename_system --source_sid=<current SID> --
       target_sid=<new SID>
       ```

   > **Note**
   >
   > If the mount point includes the SID, and you have completed the preparation in Step 1, select the *Register and Rename SAP HANA System* action in either the SAP HANA database lifecycle manager graphical user interface or command-line interface.

   - To change the SID using the SAP HANA database lifecycle manager graphical user interface:
     1. Start the graphical user interface tool:

        ```
        ./hdblcmgui
        ```

        The SAP HANA database lifecycle manager graphical user interface appears.
     2. Choose *Rename the SAP HANA Database System*.
     3. Enter the new SID in the *Target System ID* field.

4. Define the required parameters.

For more information about parameters for the `rename_system` and `register_rename_system` actions, see the *SAP HANA Server Installation and Update Guide*.

> **Note**
>
> When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the SAP HANA resident HDBLCM directory of the SAP HANA system, execute the following command:
>
> ```
> ./hdblcm --action=rename_system --help
> ```

5. To continue with the task proceed as follows:
   - In the command line interface: Enter *y*.
   - In the graphical interface:
     1. *Next*.
     2. To execute the configuration task, choose *Rename*. The system displays the configuration progress.
     3. After the configuration task has finished, you can:
        - View the log. To do so, choose *View Log*.
        - Exit the graphical user interface. To do so, choose *Finish*.

Related Information

Managing SAP HANA Licenses [page 137]
SAP HANA Server Installation and Update Guide
tenantmap

11.1.4.3 Change the Instance Number of an SAP HANA System

You can change the instance number of an SAP HANA system using SAP HANA database lifecycle manager (HDBLCM) resident program on the system which you want to configure.

Prerequisites

- You are logged in as root user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.

Context

> **Note**
>
> If you rename an SAP HANA system, this usually invalidates the permanent SAP license. A temporary license is installed, and must be replaced within 28 days. For more information, see Related Information.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the instance number change task:
   - To change an instance number using the SAP HANA database lifecycle manager command-line interface:
     - Start the command-line tool interactively:

       ```
       ./hdblcm
       ```

       and enter the index of the `rename_system` action, or
     - Start the tool with the `rename_system` action specified:

       ```
       ./hdblcm --action=rename_system --number=<new instance number>
       ```
   - To rename a host using the SAP HANA database lifecycle manager graphical user interface:
     1. Start the graphical user interface tool:

        ```
        ./hdblcmgui
        ```

        The SAP HANA database lifecycle manager graphical user interface appears.
     2. Choose *Rename the SAP HANA Database System*.
3. Define the required parameters.

   For more information about parameters for the `rename_system` action, see the *SAP HANA Server Installation and Update Guide*.

   > **Note**
   >
   > When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the SAP HANA resident HDBLCM directory of the SAP HANA system, execute the following command:
   >
   > ```
   > ./hdblcm --action=rename_system --help
   > ```
4. To continue with the task proceed as follows:

- In the command line interface: Enter *y*.
- In the graphical interface:
  1. *Next*.
  2. To execute the configuration task, choose *Rename*. The system displays the configuration progress.
  3. After the configuration task has finished, you can:
     - View the log. To do so, choose *View Log*.
     - Exit the graphical user interface. To do so, choose *Finish*.

Related Information

Managing SAP HANA Licenses [page 137]
SAP HANA Server Installation and Update Guide
tenantmap

11.2 Network Administration

Set up your SAP HANA system and the corresponding data center and network configuration in line with your organization's environment and implementation considerations.

An SAP HANA data center deployment can range from a database running on a single host to a complex distributed system with multiple hosts located at a primary and one or more secondary sites, and supporting a distributed multi-terabyte database with full high availability and disaster recovery.

How you configure your network depends on a number of considerations, including:

- Support for traditional database clients, Web-based clients, and administrative connections
- The number of hosts used for the SAP HANA system, ranging from a single-host system to a complex distributed system with multiple hosts
- Support for high availability through the use of standby hosts, and support for disaster recovery through the use of multiple data centers
- Security and performance

SAP HANA has different types of network communication channels to support the different SAP HANA scenarios and setups:

- Channels used for external access to SAP HANA functionality by end-user clients, administration clients, application servers, and for data provisioning via SQL or HTTP
- Channels used for SAP HANA internal communication within the database or, in a distributed scenario, for communication between hosts

Before you start configuring the network for SAP HANA, it's important that you understand the different types of connections to, from, and within SAP HANA and which ports to configure for them. In addition, you should be familiar with the mechanisms used for assigning and resolving host names in SAP HANA.

Security

SAP HANA supports the isolation of internal communication from outside access. To separate external and internal communication, SAP HANA hosts use a separate network adapter with a separate IP address for each of the different networks. For IBM Power systems, this might be different.

SAP HANA supports IPv6. In a mixed IP environment both IPv4 and IPv6 are present. Communication with hosts that use IP addresses is possible as well. For further information, refer to *Configuration of SAP HANA in an IPv6 Environment*.

In addition, SAP HANA can be configured to use TLS/SSL for secure communication. For more information, see the *SAP HANA Security Guide*.

Related Information

Network Zones [page 716]
Ports and Connections [page 718]
Host Name Resolution [page 743]
Configuring the Network for Multiple Hosts [page 1115]
Configuration of SAP HANA in an IPv6 Environment [page 715]

11.2.1 Configuration of SAP HANA in an IPv6 Environment

SAP HANA can be run in a dual-stack environment. In this mixed IP environment both IPv4 and IPv6 are present.

Enabling IPv6 for an Existing Installation

If you already have IPv4 installed, you can activate IPv6 additionally to IPv4 by setting the environment variable in the file `.customer.sh` that is located in the `<sid>adm $HOME` directory:

Sample Code

```
export SAP_IPv6_ACTIVE=1
```

After a restart, IPv6 is enabled for SAP HANA.

Enabling IPv6 for a New Installation

> **Note**
>
> If IPv6 is the only available IP stack, the environment variables have to be set before the installation in `/etc/profile`.

For a new installation, set the environment variables in `/etc/profile`:

> **Sample Code**

```
NI_USEIPv6=1
export NI_USEIPv6
SAP_IPv6_ACTIVE=1
export SAP_IPv6_ACTIVE
```

After a restart, IPv6 is enabled for SAP HANA.

Handling Literal IPv6 Addresses

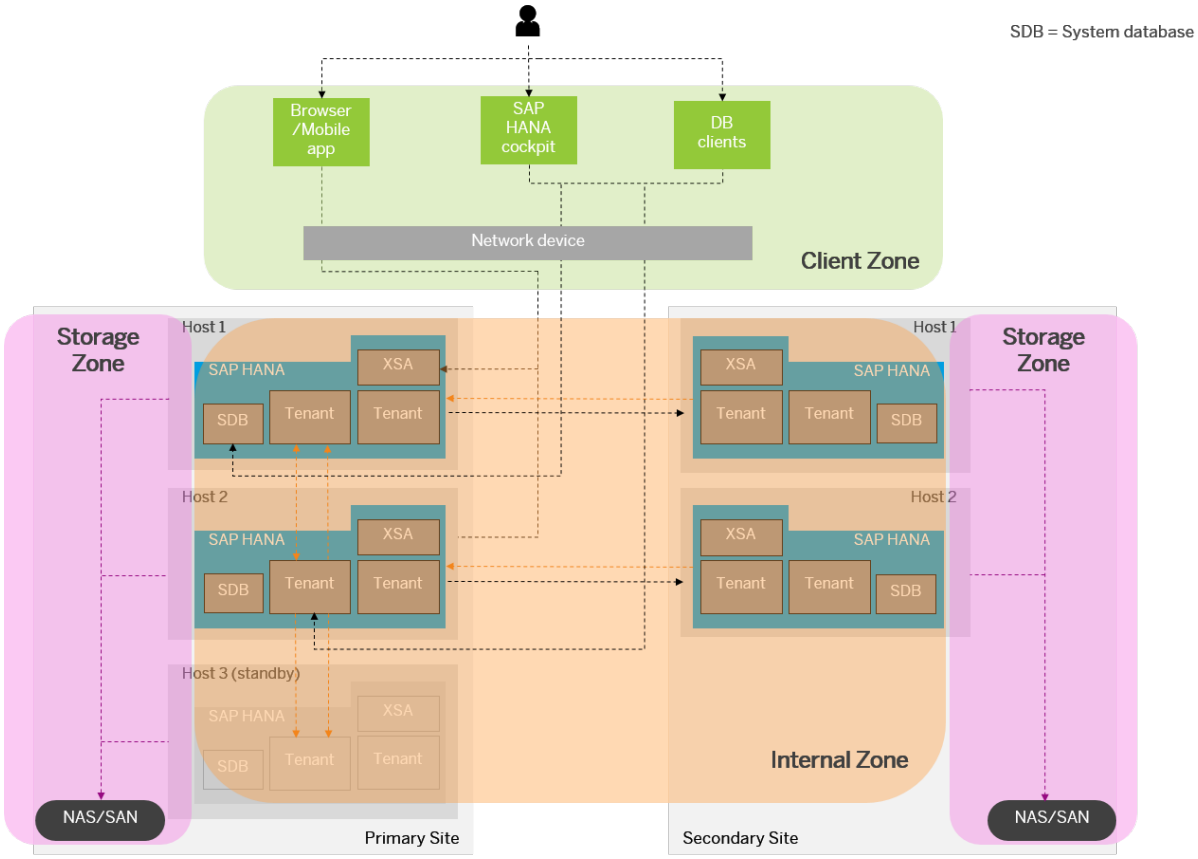
Whenever literal IPv6 addresses are involved, the host should be encapsulated with square brackets, for example [2001:DB8::1234]. This is necessary for SAP HANA to be able to distinguish the host from the port part of an address. With IPv4 this can be done by looking for the colon separator. The host part of an IPv6 address, however, contains colons instead of dots and therefore square brackets are used for distinction.

11.2.2 Network Zones

Separate network zones, each with its own configuration, allow you to control and limit network access to SAP HANA to only those channels required for your scenarios, while ensuring the required communication between all components in the SAP HANA network.

These network zones can be basically described as follows:

- Client zone
  The network in this zone is used by SAP application servers, by clients such as the SAP HANA studio or Web applications running against the SAP HANA XS server, and by other data sources such as SAP Business Warehouse.
- Internal zone
  This zone covers the interhost network between hosts in a distributed system as well as the SAP HANA system replication network.
- Storage zone
  This zone refers to the network connections for backup storage and enterprise storage.
  In most cases, the preferred storage solution involves separate, externally attached storage subsystem devices that are capable of providing dynamic mount-points for the different hosts, according to the overall landscape. A storage area network (SAN) can also be used for storage connectivity – for example, when running SAP HANA on IBM Power.



Related Information

Connections from Database Clients and Web Clients to SAP HANA [page 718]
Host Name Resolution for SQL Client Communication [page 750]
Connections for Distributed SAP HANA Systems [page 728]
Internal Host Name Resolution [page 747]
Host Name Resolution for System Replication [page 879]
SAP HANA Tailored Data Center Integration – Overview
SAP Note 1900823

11.2.3 Ports and Connections

Before you start configuring the network for SAP HANA, you'll want to get an overview of the different types of connections to, from, and within SAP HANA and which ports to configure for them.

Related Information

Connections from Database Clients and Web Clients to SAP HANA [page 718]
Connections for Distributed SAP HANA Systems [page 728]
Connections for SAP HANA Extended Application Services, Advanced Model [page 736]
Connections for Components in the Extended SAP HANA Landscape [page 739]

11.2.3.1 Connections from Database Clients and Web Clients to SAP HANA

Several types of connections between SAP HANA and external clients are possible.

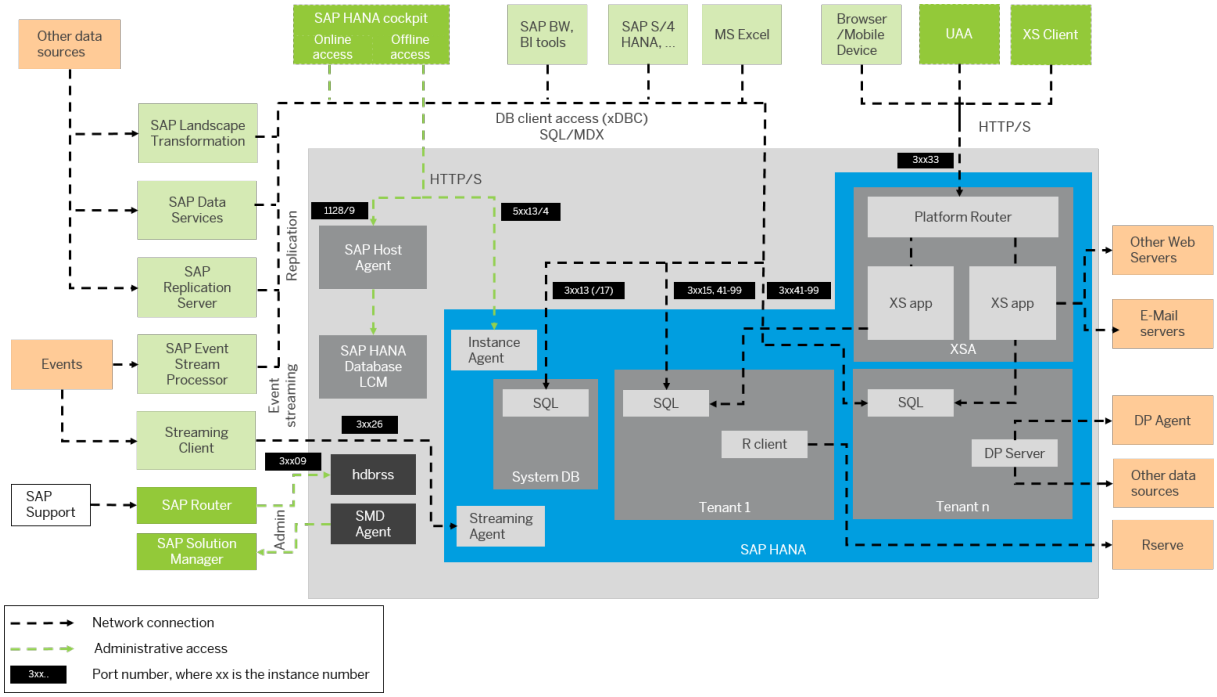
The connections between SAP HANA and external components and applications can be classified as follows:

- Connections used for administrative purposes
- Connections used for data provisioning
- Connections from database clients that access the SQL/MDX interface of the SAP HANA database
- Connections from HTTP/S clients
- Outbound connections

You can see an example of what these connections look like in the diagram below. Network connections are depicted by dashed arrows. The direction of each arrow indicates which component is the initiator (start of arrow) and which component is the listener (end point of arrow). Administrative access to and from SAP HANA is depicted by the green dashed arrows. Port numbers are shown with a black background. The "xx" in the port numbers stands for the SAP HANA instance number.

The diagram shows all the network channels used by the SAP HANA software. For purposes of illustration, the diagram shows a single-host installation with two tenant databases. However, the connections shown apply equally to a distributed scenario.

> ⓘ Note
>
> In distributed scenarios, you must also ensure that every database client can connect to every host (not shown in the diagram). Moreover, additional network channels are required in distributed scenarios for communication between the different hosts of an SAP HANA system or between the different sites. For more information, see the section on connections for distributed systems.



Connections from Database Clients and Web Clients to SAP HANA

The following tables explain the diagram and the different categories described above in more detail.

Database Client Access via ODBC/JDBC (SQLDBC)

Client	Additional Information	TCP Port
Application servers that use SAP HANA as a database Examples: SAP Business Warehouse and one or more components of SAP Business Suite	You must enable SQL/MDX access for all database clients. External and internal host names are mapped for the purposes of database client access. You can change the default mapping.	• System database: 3xx13 • First tenant database if automatically created during installation: 3xx15 • Tenant databases: 3xx41–3xx98 The port numbers of tenant databases are assigned automatically from the available port number range according to availability at the time the database is created. Administrators can also explicitly specify which port numbers to use when they create the tenant database. For more information, see the section on port assignment in tenant databases.
End-user clients that access the SAP HANA database directly Example: Microsoft Excel		

Client	Additional Information	TCP Port
SAP HANA cockpit and SAP HANA studio This connection is used for administrative purposes (for example, to access user data, configuration data or trace files).		**Note** In a new system created with one initial tenant database or an upgraded single-container system, the first tenant database is accessible on port 3xx15.

For more information about JDBC/ODBC client connections to the SAP HANA database, see the *SAP HANA Client Interface Programming Reference*.

HTTP/S Client Access

Client	Additional Information	TCP Port
XS advanced application UI (browser, mobile, and so on)	Access from applications based on SAP HANA Extended Application Services, Advanced Model (SAP HANA XS advanced)	3xx33 (hostname routing) **Note** The SAP HANA XS advanced application server supports two routing modes: port routing and hostname routing. As URLs in the hostname routing scenario are user friendly and there is only a single public port, this mode is recommended for production usage and is depicted in the figure above. If port routing is configured, all public ports of the platform router are exposed: 51000 – 51500, 3xx30, and 3xx32. For more information, see SAP Note 2245631.
XS User Account and Authentication service (browser, mobile, and so on)	Access from the application client to the xscontroller-managed Web Dispatcher (platform router) for the purposes of user authentication.	
• XSA command line client • Client library (Java) • One or more SAP HANA XS advanced model applications used, for example, for administrative and/or monitoring purposes	Access to the xscontroller-managed Web Dispatcher for purposes of data access	

Client	Additional Information	TCP Port
XS classic application UI (browser, mobile, and so on)	Access for applications based on SAP HANA Extended Application Services, classic model (SAP HANA XS classic). The SAP HANA platform itself has a number of Web applications that run on SAP HANA XS classic, for example, the SAP HANA Web-based Development Workbench and SAP HANA Application Lifecycle Management. **Note** SAP HANA XS, classic is deprecated as of SAP HANA 2.0 SPS 02. For more information, see SAP Note 2465027.	80xx/43xx **Note** This port is not depicted in the graphic above.
SAP HANA Direct Extractor Connection (DXC)	This connection is used for ETL-based data acquisition. For more information, see the *SAP HANA Direct Extractor Connection Implementation Guide*.	
UI toolkit for SAP HANA Info Access	This connection is used for the SAP HANA Info Access HTTP search service. SAP HANA Info Access provides UI building blocks for developing browser-based search apps on SAP HANA. For more information, see the *SAP HANA Search Developer Guide*.	
SAP HANA cockpit, SAP HANA studio	This is the connection to the SAP HANA database lifecycle manager via SAP Host Agent. For more information about the SAP HANA database lifecycle manager, see the section on SAP HANA Platform Lifecycle Management.	1128 1129 (SSL)

Note

SAP HANA XS advanced server components and application instances also expose ports that are not designed for external communication. For more information about these, see *Connections for SAP HANA Extended Application Services, Advanced Model* and the *SAP HANA Security Guide*.

Administrative Tasks

Client	Protocol	Additional Information	TCP Port
SAP support	Internal SAP protocol	The connection is not active by default because it is required only in certain support cases. To find out how to open a support connection, see the section on getting support.	3xx09
SAP HANA cockpit, SAP HANA studio	SQLDBC (ODBC/JDBC)	The connection to the instance agent acts as an administrative channel for low-level access to the SAP HANA instance to allow features such as starting or stopping of the SAP HANA database.	5xx13 5xx14 (SSL)

Other administrative tasks, mainly database administration, use the SQL/MDX channel of the database.

Data Provisioning

Client	Protocol	Additional Information	TCP Port
Replication systems for external data sources	SQLDBC (ODBC/JDBC)	The following replication technologies may be used: • SAP Landscape Transformation (SLT) • SAP Data Services (DS) • SAP Replication Server (not included with all licensed editions of SAP HANA)	• System database: 3xx13 • Tenant databases: 3xx41–3xx98 The port numbers of tenant databases are assigned automatically from the available port number range according to availability at the time the database is created. Administrators can also explicitly specify which port numbers to use when they create the tenant database. For more information, see the section on port assignment in tenant databases. **Note** In a new system created with one initial tenant database or an upgraded single-container system, the first tenant database is accessible on port 3xx15.
	HTTP/S	SAP HANA Direct Extractor Connection (DXC). This technology uses HTTP/S access via the SAP HANA XS classic server.	80xx/43xx **Note** This port is not depicted in the graphic above.

Client	Protocol	Additional Information	TCP Port
Streaming client	XML/RPC	This connection is used for SAP HANA Streaming Analytics (supported on Intel-based platforms only) **⚠ Caution** SAP HANA Streaming Analytics is an SAP HANA option. Be aware that you need additional licenses for SAP HANA options. For more information, see Important Disclaimer for Features in SAP HANA [page 1677].	3xx26

Outbound Connections

Connection	Additional information
From the SAP Solution Manager diagnostics (SMD) agent to SAP Solution Manager	For information about how to install the SAP Solution Manager diagnostics agent, see SAP Note 1858920.
Calls from SAP HANA Extended Application Services to external servers	Examples: a Web server or an e-mail server (depends on what applications your company has deployed)
Smart data access from SAP HANA to external data sources for data federation purposes	For more information, see the section on smart data access.
From SAP HANA to the R environment	Only required for scenarios which use the R integration supported by SAP HANA. For more information, see the *SAP HANA R Integration Guide*.
From the data provisioning (DP) server of the SAP HANA database to the data provisioning agent and, depending on the type of adapter used, to the external data source(s)	This connection is used for SAP HANA smart data integration in scenarios where SAP HANA is deployed on premise. For more information, see the *Installation and Configuration Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality*. SAP HANA with the DP server can run on IBM Power. However, the data provisioning agent needs to be hosted on an Intel machine. It is possible to connect between the two. **⚠ Caution** SAP HANA smart data integration is an SAP HANA option. Be aware that you need additional licenses for SAP HANA options. For more information, see Important Disclaimer for Features in SAP HANA [page 1677].

Related Information

Connecting to SAP HANA Databases and Servers (Client Interface Guide)
Host Name Resolution for SQL Client Communication [page 750]
Connections for SAP HANA Extended Application Services, Advanced Model [page 736]
Connections for Components in the Extended SAP HANA Landscape [page 739]
SAP Note 2245631
SAP HANA Smart Data Access [page 1519]
SAP Note 1858920
SAP Note 2465027

11.2.3.1.1 Port Assignment in Tenant Databases

Every tenant database has its own ports and connections for internal and external communication.

Every tenant database has dedicated ports for SQL and internal communication. There is also a dedicated port for HTTP-based client communication via the SAP HANA XS classic server, which runs by default as an embedded service in the index server.

However, there are no standard port number assignments. Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.

The only exceptions to this are the tenant database that is automatically created when you **install** a single-tenant system and when you **convert** a single-container system to a tenant database system. This database retains the port numbers of the original single-container system: 3`<instance>`03 (internal communication), 3`<instance>`15 (SQL), and 3`<instance>`08 (HTTP via SAP HANA classic server). The ports of any subsequently added tenant database are automatically assigned according to availability at the time.

The default port number range for tenant databases is 3`<instance>`40—3`<instance>`99. This means that the maximum number of tenant databases that can be created per instance is 20. However, you can increase this by reserving the port numbers of further instances. In the cockpit, a dialog will prompt you to do this, or you can configure the property `[multidb] reserved_instance_numbers` in the `global.ini` file. The default value of this property is 0. If you change the value to 1, the port numbers of one further instance are available (for example, 30040—30199 if the first instance is 00). If you change it to 2, the port numbers of two further instances are available (for example, 30040—30299 if the first instance is 00). And so on.

> **Note**
>
> The port number of the **system database** are fixed: 3`<instance>`01 (internal), 3`<instance>`13 (SQL), and 3`<instance>`14 (HTTP via XS classic server). If restricted SQL access is enabled, port 3`<instance>`17 is used for SQL requests to the system database.

Let's look at some simple examples.

Example

Example 1:

You perform a default SAP HANA system installation. This results in the automatic creation of a single tenant database. This tenant database has the following port numbers: 3<instance>03 (internal communication), 3<instance>15 (SQL), 3<instance>08 (HTTP via XS classic server).

Then, you create two additional tenant databases. Each of these tenant databases is automatically assigned port numbers for the following connection types:

- Internal communication
- SQL
- HTTP (This is the port of the XS classic server embedded in the index server.)

The second database is assigned ports 3<instance>40—42, and the third 3<instance>43—45.

Example 2:

You install a new SAP HANA system without creating an initial tenant, by running the SAP HANA database lifecycle manager (HDBLCM) with the `create_initial_tenant` flag set to `off`. Then, you create three tenant databases. Each of these tenant databases is automatically assigned the following port numbers:

The first tenant database is assigned port numbers 3<instance>40—42, the second ports 3<instance>43—45, and the third 3<instance>46—48.

Example 3:

You install a new SAP HANA system without creating an initial tenant, by running the SAP HANA database lifecycle manager (HDBLCM) with the `create_initial_tenant` flag set to `off`. Then, you create a tenant database. The same port numbers as above are assigned: 3<instance>40 (internal communication), 3<instance>41 (SQL), and 3<instance>42 (HTTP via XS classic server). Next, you add a separate xsengine service to the first database. This service is automatically assigned the next three available port numbers: 3<instance>43—45. Finally, you create a second tenant database. This tenant database is automatically assigned the next three available port numbers: 3<instance>46—48.

Example 4:

You convert a single-container system to a tenant database system. This results in the automatic creation of one tenant database. This tenant database has the same port numbers as the original single-container system: 3<instance>03 (internal communication), 3<instance>15 (SQL), 3<instance>08 (HTTP via XS classic server). Then, you add a second index server to the tenant database. It is automatically assigned port numbers 3<instance>40—42. Finally, you create a second tenant database. It is automatically assigned ports the next three available port numbers: 3<instance>43—45.

Example 5:

You can run multiple services of the same type in a single tenant on the same host. The service ports must be within the same port range and must not span across multiple port ranges. For example, you can assign port 30**10**0 to index server A and port 30**10**3 to index server B. You cannot assign port 30**10**0 to index server A and port 30**20**3 to index server B.

Note

All of the above examples refer to single-host systems and are based on automatic port number assignment.

You can determine the ports used by a particular tenant database by querying the M_SERVICES system view, either from the tenant database itself or from the system database.

- From the tenant database: `SELECT SERVICE_NAME, PORT, SQL_PORT, (PORT + 2) HTTP_PORT FROM SYS.M_SERVICES WHERE ((SERVICE_NAME='indexserver' and COORDINATOR_TYPE= 'MASTER') or (SERVICE_NAME='xsengine'))`
- From the system database: `SELECT DATABASE_NAME, SERVICE_NAME, PORT, SQL_PORT, (PORT + 2) HTTP_PORT FROM SYS_DATABASES.M_SERVICES WHERE DATABASE_NAME='<DBNAME>' and ((SERVICE_NAME='indexserver' and COORDINATOR_TYPE= 'MASTER') or (SERVICE_NAME='xsengine'))`

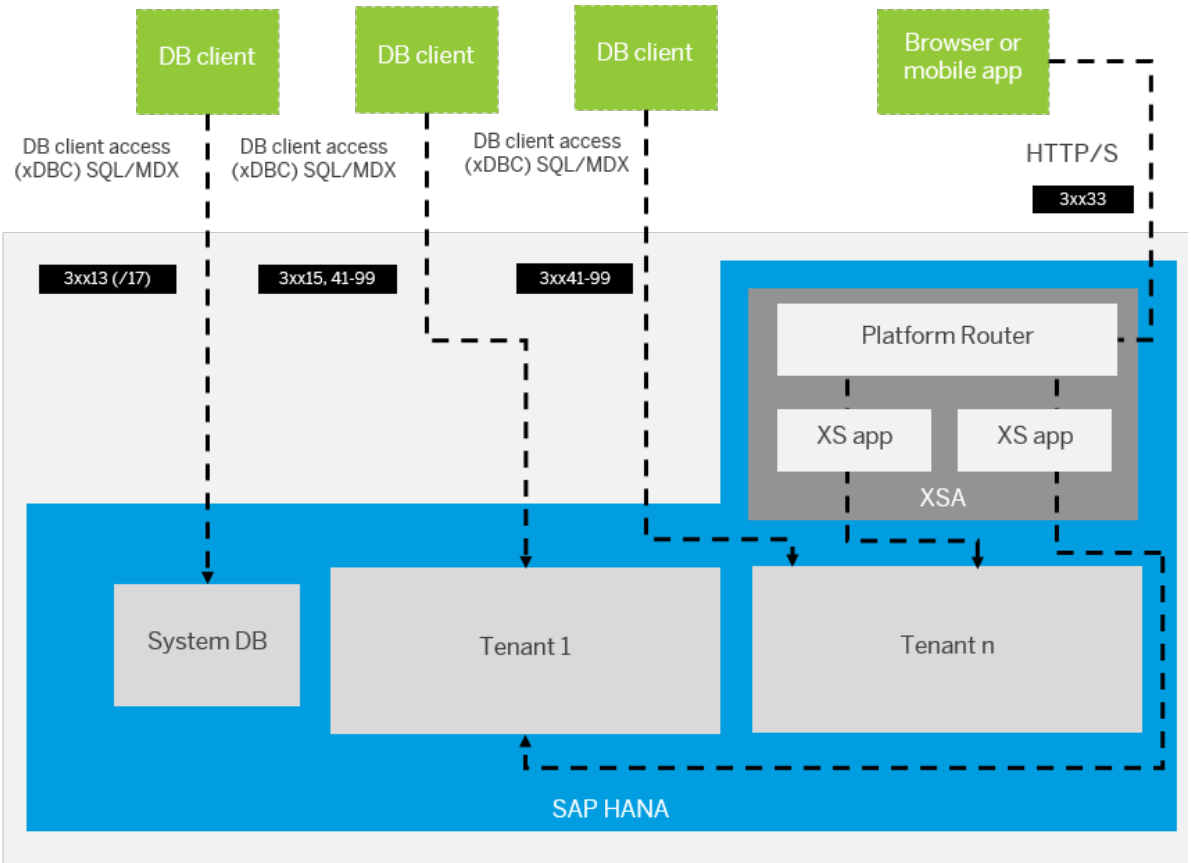
→ Remember

If your system was converted from single-container mode to a tenant database system, the HTTP port number of the first tenant database is **always** 3`<instance>`08 and not the port number returned using the above queries.

ⓘ Note

System privilege DATABASE ADMIN or CATALOG READ is required to read the M_SERVICES system view.

The following diagram shows an example of the connections and ports used in a system with two tenant databases, installed on a single host. The SAP HANA XS advanced runtime is used to run Web applications.



> **Note**
>
> The SAP HANA XS advanced application server supports two routing modes: port routing and hostname routing. As URLs in the hostname routing scenario are user friendly and there is only a single public port, this mode is recommended for production usage and is depicted in the figure above. If port routing is configured, all public ports of the platform router are exposed: 51000 – 51500, 3xx30, and 3xx32. For more information, see SAP Note 2245631.

Restricted SQL Access

If tenant databases need to be accessible from an external network, you can open an additional SQL port to prevent SQL access on port 3<instance>13. This prevents the exposure of the system database SQL administration port to the external network. You enable this feature by setting the property `[multidb] systemdb_separated_sql_port` to `true` in the `global.ini` file.

This opens port 17 for SQL requests to the system database and restricts access through port 3<instance>13 for database mapping. The connection through port 3<instance>13 is re-routed to 3<instance>17 if a connection to the system database is required. Make sure that port 17 is not exposed to the external network.

Related Information

Creating and Configuring Tenant Databases [page 65]

SAP Note 2245631

Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]

11.2.3.2 Connections for Distributed SAP HANA Systems

SAP HANA systems can be distributed across multiple hosts for the purposes of scalability and availability.

An SAP HANA system is identified by a system ID (SID). Tenant databases are identified by the SID and their database name. Both are perceived as discrete units from the administration perspective. Some tasks are performed at system level (for example, installation, update, stop, and start) and others at database level (for example, database configuration, schema and table management). The system database is used for central system administration. The services of each database share the same metadata, and requests from client applications are transparently dispatched to the different services in the database. Servers that do not persist data, such as the compile server and the preprocessor server, run on the system database and serve all tenant databases.

A **distributed SAP HANA system** is a system that is installed on more than one host. An **SAP HANA instance** is a set of components of a distributed system that are installed on one host. Tenant databases can run indvidually on a single host or be distributed across several hosts.

Connections for Internal Communication Between Hosts and Sites

In addition to external network connections, SAP HANA uses separate, dedicated connections exclusively for internal communication. There are two types of internal communication in distributed systems:

- Communication between hosts in a multiple-host system (scale-out)
  Internal network communication takes place between the hosts of a distributed system on one site. SAP HANA hosts contain a separate network interface card that is configured as part of a private network, using separate IP addresses and ports. For IBM Power systems, this might be different.

  > ### Note
  >
  > In single-host scenarios, the same communication channels are used for communication between the different processes on a single host and the internal IP addresses/ports are by default bound to the `localhost` interface.

  There are a number of ways to isolate internal network ports from the client network. The preferred method depends on the data center configuration, on hardware vendor delivered options, and on the high availability implementation. Applying network separation for the internal communication prevents unauthorized access from outside networks. For additional security, it is possible to encrypt the internal communication using TLS/SSL. For more information, see the *SAP HANA Security Guide*.

  Ports for Multiple-Host System

Client	TCP Port	Service	Note
Hosts of a distributed system on one site	3xx00	daemon	
	3xx01	nameserver	System database port only
	3xx02	preprocessor	System database port only
	3xx03	indexserver	Port used by the indexserver of the inital tenant database created in a new installation or an upgraded single-container system.
	3xx40 - 3xx97	indexserver	Tenant database ports Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.

Client	TCP Port	Service	Note
	3xx04	scriptserver	Port used by the script server of the first tenant database created in a new installation or an upgraded single-container system (optional)
	3xx40 - 3xx97	scriptserver	Tenant database ports (optional) Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.
	3xx40 - 3xx97	docstore	Tenant database ports (optional) Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.
	3xx05	diserver	Tenant database ports Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.

Client	TCP Port	Service	Note
	3xx25	diserver	Port used by the diserver of the inital tenant database created in a new installation or an upgraded single-container system.
	3xx40	diserver	Tenant database ports Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.
	3xx06	webdispatcher	Port used by the webdispatcher of the inital tenant database created in a new installation or an upgraded single-container system.
	3xx07	xsengine	Port used by the xsengine of the inital tenant database created in a new installation or an upgraded single-container system.
	3xx08	xsengine	Port used by the xsengine of the inital tenant database created in a new installation or an upgraded single-container system.
	3xx42 - 3xx99	xsengine	Tenant database ports Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.

Client	TCP Port	Service	Note
	3xx10	compileserver	System database port only
	3xx11	dpserver	Port used by the dpserver of the inital tenant database created in a new installation or an upgraded single-container system.
	3xx33	xscontroller	System database port only, if hostname routing is configured The SAP HANA XS advanced application server supports two routing modes: port routing and hostname routing. As URLs in the hostname routing scenario are user friendly and there is only a single public port (3xx33), this mode is recommended for production usage and is depicted in the figure below. If port routing is configured, all public ports of the platform router are exposed: 51000 – 51500, 3xx30, and 3xx32. For more information, see SAP Note 2245631.

> **Note**
>
> The "xx" in the port numbers is the SAP HANA instance number.

- Communication between sites in a system replication configuration (high availability)
  Internal network communication for system replication takes place between a primary site and a secondary site. In a multitier setup, this communication takes place between the tier-1 primary system and tier-2 secondary system as well as, asynchronously, between the tier-2 and tier-3 secondary systems. For more information about system replication and multitier setups, see the section on high availability. System replication is configured for the system as a whole. This means that the system database and all tenant databases are part of the system replication.
  System replication connections must be secured using TLS/SSL. In this case, landscape topology communication on the one hand, and data replication and log replication channels on the other, are secured in separate steps. For more information about configuring SSL for internal communication as well as securing communication between sites in system replication scenarios, see the *SAP HANA Security Guide*.

Ports for System Replication

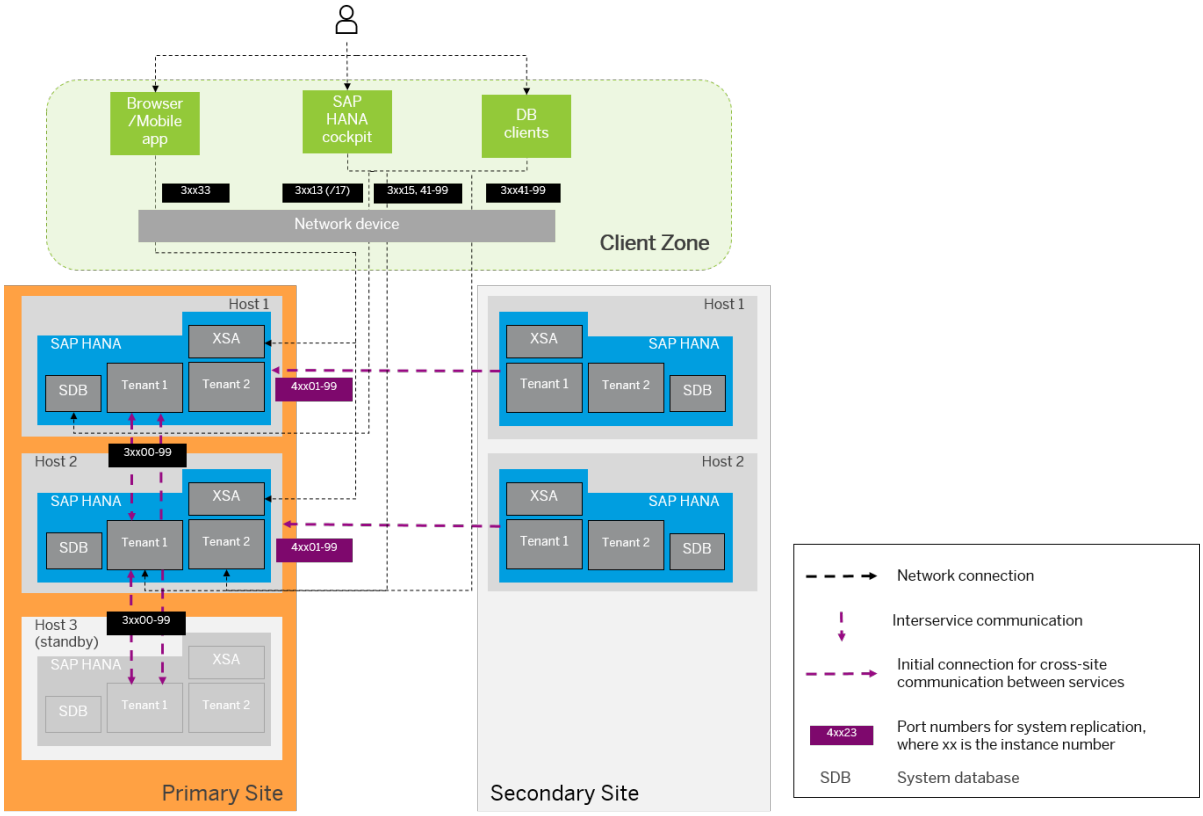
Client	TCP Port	Service	Used For
Hosts on primary and secondary sites	4xx01	nameserver	Log and data shipping (system database)
	4xx02	nameserver	Unencrypted metadata communication (system database)
	4xx06	nameserver	TLS/SSL encrypted metadata communication (system database)
	4xx40 - 4xx97	indexserver	Log and data shipping (tenant databases)
	4xx40 - 4xx97	scriptserver	Log and data shipping (optional, tenant databases)
	4xx40 - 4xx97	docstore	Log and data shipping (optional, tenant databases)

> **Note**
>
> A port offset value of 10,000 is used to reserve ports for system replication communication. This shifts the ports from the 3<instance number>00 to the 4<instance number>00 port range for services.

> **Note**
>
> SAP HANA internal communication has sometimes been unofficially referred to as TREXNet communication. However, the term TREXNet is not valid in the context of SAP HANA.

Example 1

The following diagram shows a multiple-host SAP HANA system with two active hosts and an extra standby host, fully system replicated to a secondary site to provide full disaster recovery support.



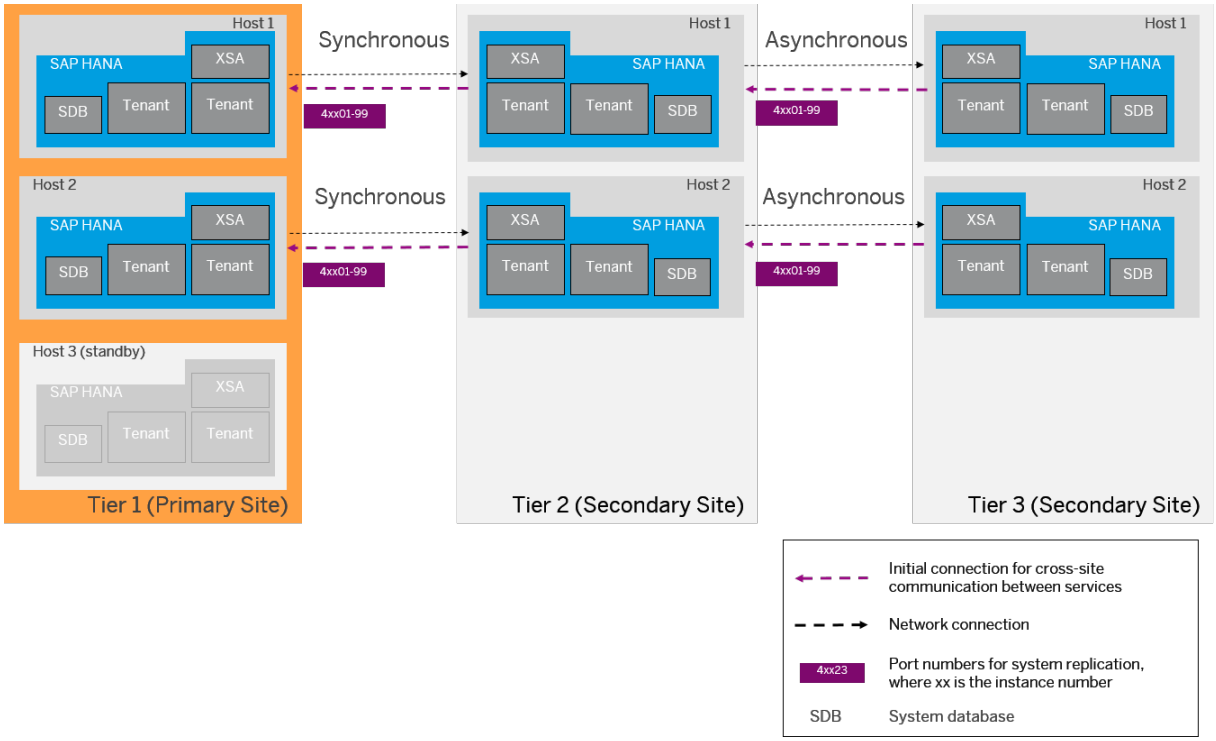
The vertical purple dashed lines show the communications between the services of a database; all instances communicate with all other instances of a multiple-host system on one site. The horizontal purple dashed lines show the initial connection for system replication communication between services on hosts on the primary site and the corresponding services on hosts of the secondary site (typically over a high-performance fiber network). The details of system replication configuration depend on the specific network setup of your company.

One of the most critical aspects of the network design of a highly available distributed system is the question of how the different clients manage to reconnect to the system when its topology changes due to the recovery operations following a failure or disaster. Two additional components can be used to handle client reconnection:

- A **network device** (router and/or switch), which can be used in conjunction with DNS or virtual IP redirection
- An **HTTP load balancer** (such as SAP Web Dispatcher) acts as a reverse proxy for HTTP connections and exposes a consistent external network address to the client network. The HTTP load balancer can also be used to provide load-balanced access to multiple distributed SAP HANA Extended Application Services (XS advanced) servers. For more information, see SAP Note 2300936.

Example 2

The following diagram shows an example of multi-tier system replication:



Related Information

Internal Host Name Resolution [page 747]
Connections from Database Clients and Web Clients to SAP HANA [page 718]
Host Name Resolution for System Replication [page 879]
SAP HANA System Replication with Tenant Databases [page 769]
Setting Up the XS Advanced Runtime Behind a Reverse Proxy
Host Auto-Failover Setup with XS Advanced Run Time [page 902]
SAP Note 2245631
SAP Note 2300936

11.2.3.3 Connections for SAP HANA Extended Application Services, Advanced Model

Additional ports and connections are required if you are using SAP HANA extended application services, advanced model (SAP HANA XS advanced).

Client	Service	TCP Port	Use
Platform Router	`xsuaaserver`	3xx31	Internal HTTP(S) This port is used for the connection from the platform router to `xsuaaserver` for purposes of user authentication.
Platform Router	`xscontroller`	One dynamic port, in the range 51000-51500	Internal HTTP(S) This port range is used for the connection from the platform router to the xscontroller for purposes of data access.
Application UI (browser, mobile, and so on)	Platform Router	Dynamic, in the range 51000-51500	Client HTTP(S) This port range is used for the connection from the client to the platform router for access to the application instance.
Platform Router	Application Instances	Dynamic, in the range 50000-50999	Internal HTTP(S) This port range is used in **single-host scenarios** for the connection from the platform router to the application instances.
Platform Router	Host-specific Web Dispatcher	Dynamic, in the range 50500-50999	Internal HTTP(S) This port range is used in **multiple-host scenarios** for the connection from the platform router to the host-specific Web Dispatcher.

Client	Service	TCP Port	Use
Host-specific Web Dispatcher	Instances	Dynamic, in the range 50000-50499	Internal HTTP(S) This port range is used in **multiple-host scenarios** for the connection from the host-specific Web Dispatcher to the application instance.
`xsexecagent`	`xscontroller`	3xx29	Internal HTTP(S) These ports are used for the connection between the xs execution agent and the `xscontroller`.
`xscontroller`	`xsexecagent`	System	

Port-Routing Mode

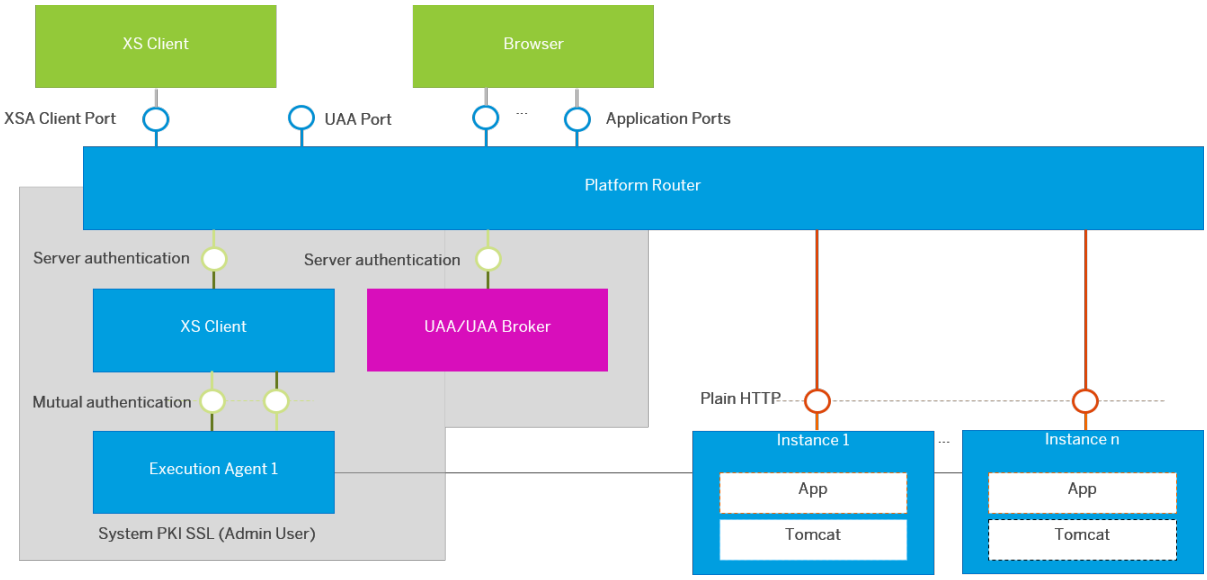
Client	Service	TCP Port	Use
Application UI (browser, mobile, and so on)	`Platform Router`	3xx32	Client HTTP(S) This port is used for the connection from the client to the platform router for purposes of user authentication.
• Command line client • SAP Web IDE • XSA Cockpit • Deploy Service • Monitoring applications	`Platform Router`	3xx30	Client HTTP(S) This port is used for the connection to the platform router for purposes of data access.

Hostname-Routing Mode

Client	Service	TCP Port	Use
Application UI (browser, mobile, xs client, SAP Web IDE, XSA Cockpit, monitoring tools and so on)	Application instances	3xx33	Web Dispatcher HTTP(S) This port is used for the platform router where routing is done by host names instead of ports. In this case, the `xscontroller` is available with URL `https://api.<example.com>:3xx33` and the xsuaaserver is available with URL `https://uaa-server.<example.com>:3xx33`. For more information, see SAP Note 2245631.

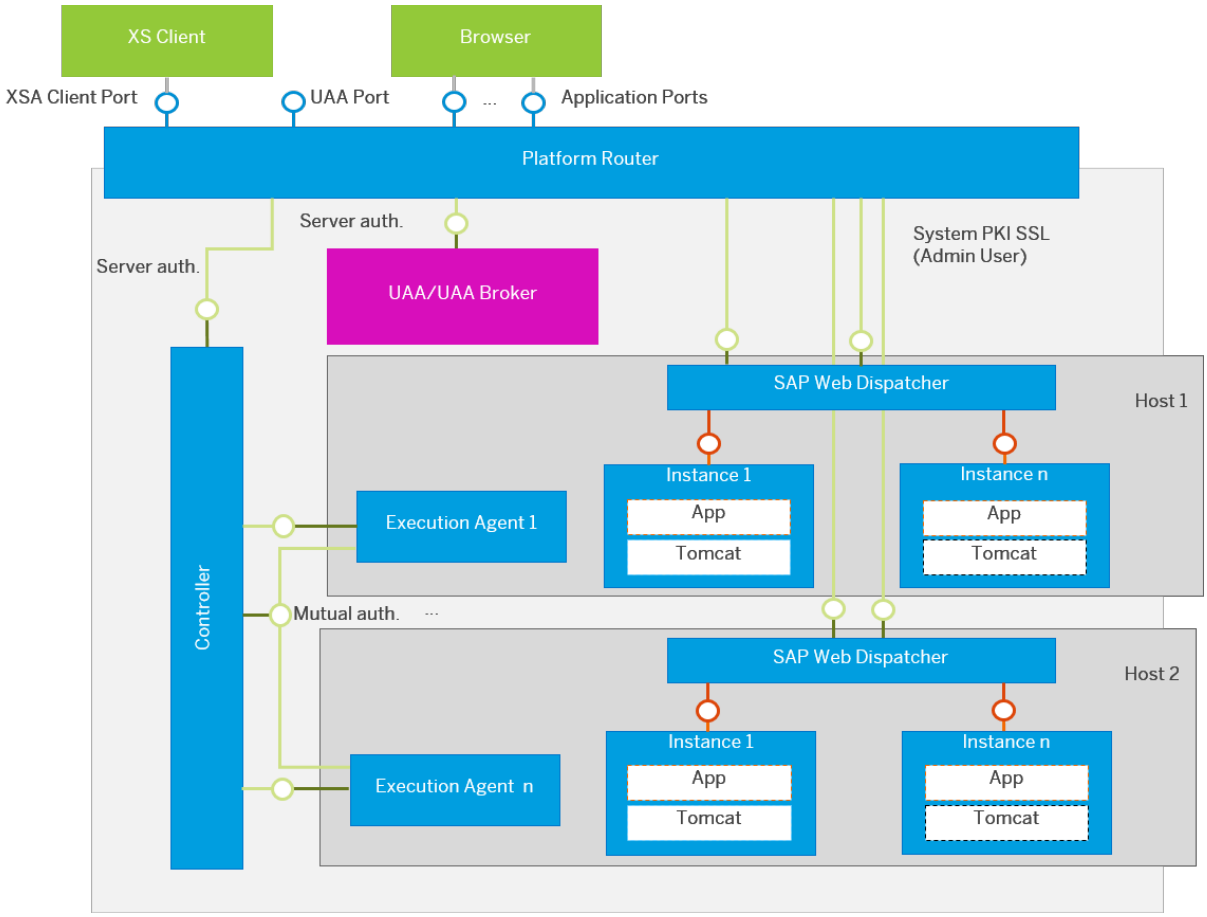
The following diagrams provide an overview of the ports and connections used by the SAP HANA XS advanced server infrastructure in single- and multiple-host scenarios. For more information about network and communication security, see the *SAP HANA Security Guide*.

XS Advanced Ports and Connections (Single-Host Scenario)



XS Advanced Ports and Connections (Single-Host Scenario)

XS Advanced Ports and Connections (Multiple-Host Scenario)



XS Advanced Ports and Connections (Multiple-Host Scenario)

Related Information

SAP Note 2245631
SAP HANA System Architecture Overview [page 15]
SAP HANA XS and Development Infrastructure
Network and Communication Security with SAP HANA XS Advanced
Setting Up the XS Advanced Runtime Behind a Reverse Proxy

11.2.3.4 Connections for Components in the Extended SAP HANA Landscape

Dedicated ports are used to connect components in the extended SAP HANA landscape.

- Connections for SAP HANA Dynamic Tiering (Deprecated) [page 740]
- Connections for SAP HANA Smart Data Integration [page 741]
- Connections for SAP HANA Streaming Analytics [page 741]

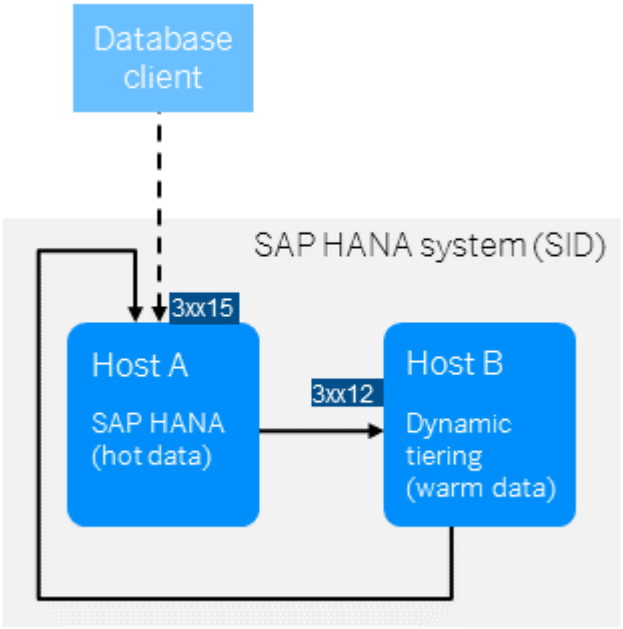
- Connections for SAP HANA Accelerator for SAP ASE [page 742]

11.2.3.4.1 Connections for SAP HANA Dynamic Tiering (Deprecated)

No additional manual configuration of connections and ports is required in the SAP HANA software for SAP HANA dynamic tiering.

> **Note**
>
> Dynamic tiering is deprecated. SAP HANA Native Storage Extension replaces it. See SAP Note 3414255.

When an external client sends a request for warm data, it connects to the SAP HANA host which passes the request to the dynamic tiering host. The dynamic tiering host listens on internal port 3xx12. There is no direct connection between external components and the dynamic tiering host. The connection back from the dynamic tiering host to the SAP HANA host is through the SQL port of the SAP HANA host.



Related Information

SAP HANA Dynamic Tiering
Important Disclaimer for Features in SAP HANA

11.2.3.4.2 Connections for SAP HANA Smart Data Integration

The connections between the components for SAP HANA smart data integration may differ depending on whether SAP HANA is deployed on premise, in the cloud, or behind a firewall.

For more information, see the *Installation and Configuration Guide* for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality.

Related Information

SAP HANA Smart Data Integration and SAP HANA Smart Data Quality

11.2.3.4.3 Connections for SAP HANA Streaming Analytics

The internal connections and ports for SAP HANA streaming analytics are set up automatically. None of the ports are configurable.

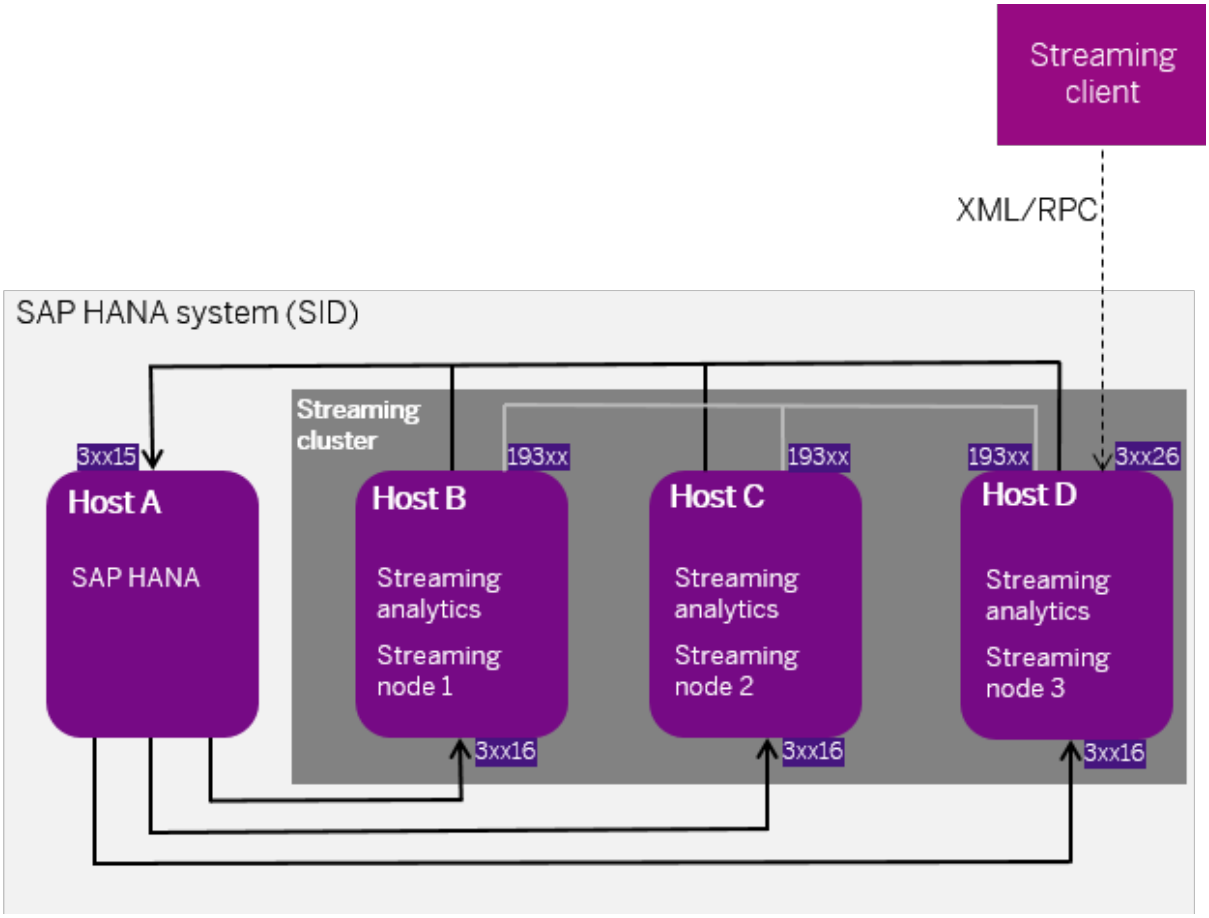
> **Note**
>
> SAP HANA streaming analytics is supported on Intel-based platforms only.

The SAP HANA server connects to one or more streaming analytics servers on internal port 3xx16. Through this connection, SAP HANA gathers streaming analytics statistics. The connection is triggered by the SAP HANA cockpit monitoring views.

The streaming analytics hosts connect to the SAP HANA server on the SQL port of the tenant database. The streaming hosts retrieve the streaming license information and the streaming cluster configuration (which is stored on the SAP HANA database). If thestreaming analytics project has an SAP HANA adapter or a generic database adapter that connects to SAP HANA, it also uses the SQL port connection.

Any streaming clients that run outside the SAP HANA system (such as custom-built external adapters) connect to a streaming node via the XML/RPC protocol on port 3xx26.

In a multi-node setup, the 193xx port is used for interserver communication between streaming hosts. This port is for internal use, but you may want to make a note of it for firewall settings.



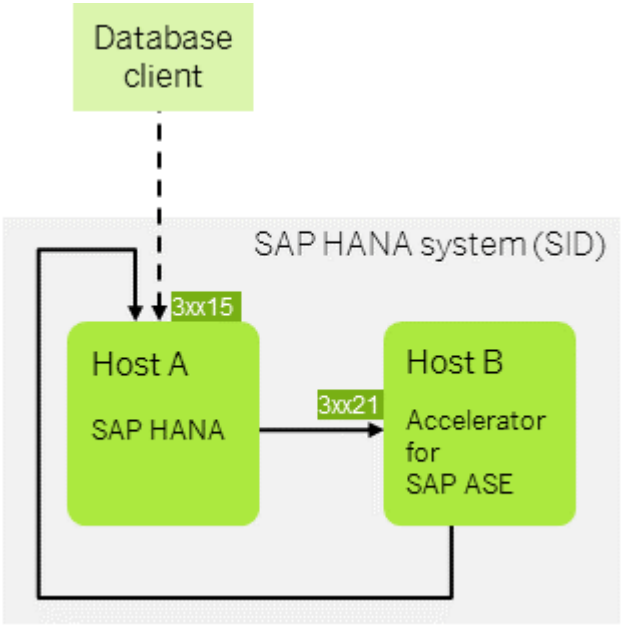
Related Information

SAP HANA streaming analytics
Important Disclaimer for Features in SAP HANA

11.2.3.4.4 Connections for SAP HANA Accelerator for SAP ASE

The internal connections and ports for SAP HANA accelerator for SAP ASE are set up automatically.

When an external client sends a request for warm data, it connects to the SAP HANA host which passes the request to the accelerator for SAP ASE host. The accelerator for SAP ASE host listens on internal port 3xx21. The connection back from the accelerator for SAP ASE host to the SAP HANA host is through the SQL port of the SAP HANA host. Any SAP ASE clients that run outside the SAP HANA system can connect to an accelerator for SAP ASE node on port 3xx21 directly.



Related Information

SAP HANA Accelerator for SAP ASE
Important Disclaimer for Features in SAP HANA

11.2.4 Host Name Resolution

Understand the mechanisms used for assigning and resolving host names in SAP HANA.

Related Information

Default Host Names and Virtual Host Names [page 744]
Internal Host Name Resolution [page 747]
Host Name Resolution for System Replication [page 879]
Host Name Resolution for SQL Client Communication [page 750]

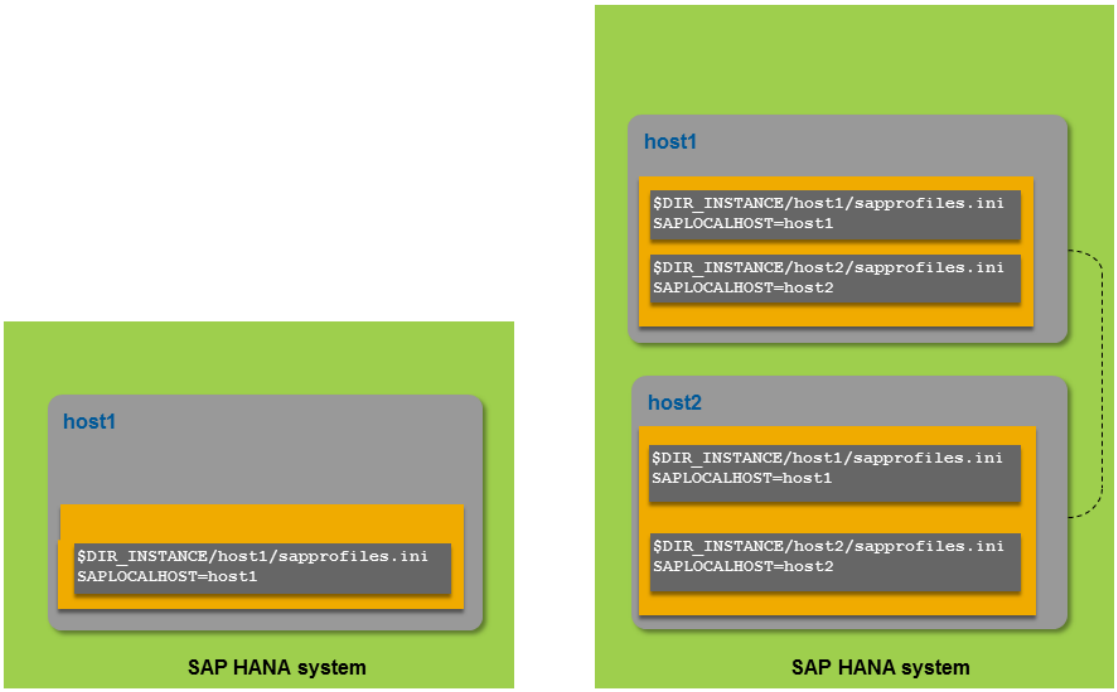
11.2.4.1 Default Host Names and Virtual Host Names

The assignment of multiple host names to the same host supports performance optimization as well as the security of your SAP HANA system. Moreover, some cluster managers and third-party backup tools as well as SAP Landscape Management work on the basis of virtual host names or IP aliases.

Default Host Names

The default host names, if nothing else is configured during the installation of SAP HANA, are the host names defined at operating system level. The installation extracts the host names known to the operating system (that is, the names of the SAP HANA instances) and stores them in the sapstart service profiles, that is, in the following files:

```
/usr/sap/sapservices
/usr/sap/<SID>/HDB<instance_number>/<hostname>/sapprofile.ini
```



Example of Default Host Names for SAP HANA

These host names are then used for all internal communication between SAP HANA services (`nameserver`, `indexserver`, and so on) and the SAP start service (`sapstartsrv`). In addition, SAP HANA system views with a HOST column show these host names.

Virtual Host Names

Another approach is to specify alternative host names during installation. These are referred to as virtual host names. Virtual host names must also be unique across multiple SAP HANA systems if more than one data center or site is used.

Host names specified in this manner must be resolvable during installation time, as well as when SAP HANA is in operation. This is achieved, for example, by adding an `<ip> <hostname>` line to the operating system file `/etc/hosts` that contains the hostname-to-IP address mappings for the TCP/IP subsystem. Here is an example of what this might look like at operating system level for one host:

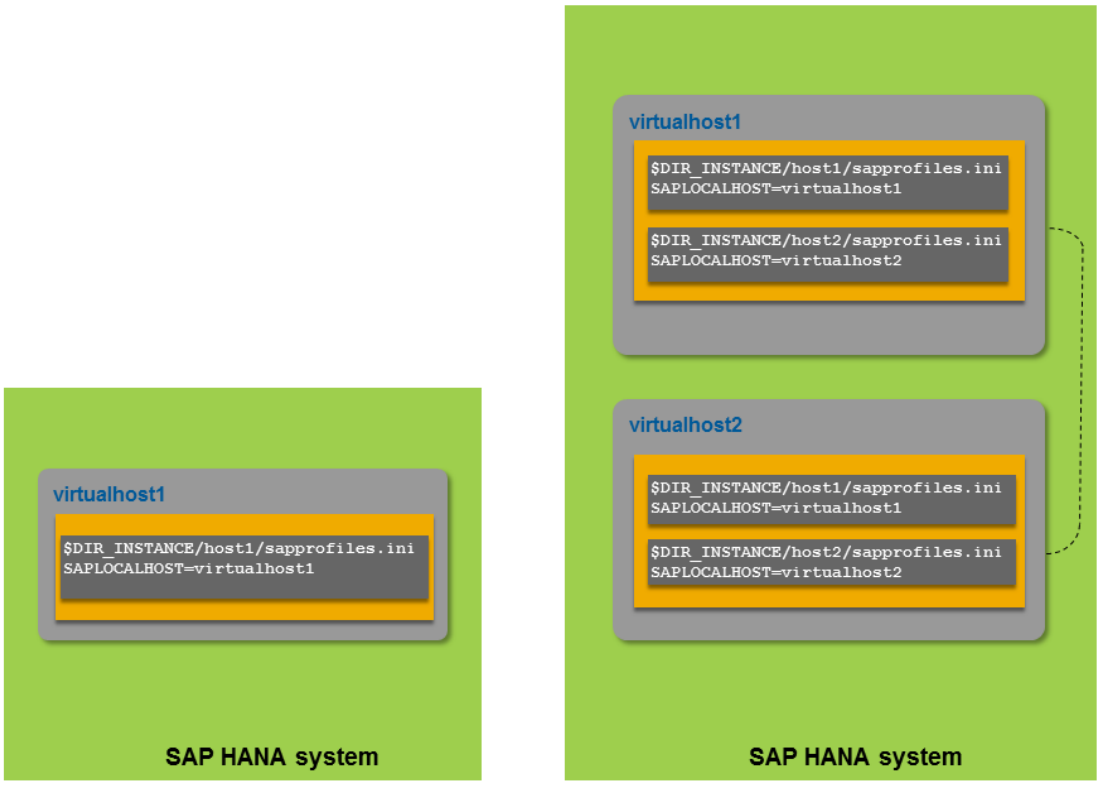
```
127.0.0.1       localhost
10.68.91.226    virtualhost1.wdf.sap.corp virtualhost1
```

> **Note**
>
> The hosts file has a strictly defined structure. The three elements of each line are: IP address, fully qualified domain name, hostname. All lines and elements in the file must be unique. Modifications to the hosts file are effective after the next OS restart, or after a restart of the network service and the diagnostic agent.

Virtual host names are assigned as part of the installation process with the platform LCM command-line tool `hdblcm` using the `hostname` parameter. For more information about using the command-line tool or the `hostname` parameter, see the *SAP HANA Server Installation and Update Guide*.

The `<virtualhostname>` is then stored as the internal host name in the `sapstart` service profiles and shows up in the HOST column of any system view.

It is also possible to assign virtual host names once the system is up and running by using the platform LCM action `system_rename` with the `hostmap` parameter. For more information about mapping hosts, see *Rename an SAP HANA System Host*.



Example of Virtual (Internal) Host Names for SAP HANA

Distributed Landscapes

In multiple-host systems used for scale-out, the host names of all hosts must be known to each SAP HANA host. The `/etc/hosts` file for each host must include the corresponding lines:

```
host1
127.0.0.1       localhost
10.68.91.226    virtualhost1.wdf.sap.corp virtualhost1
10.68.91.227    virtualhost2.wdf.sap.corp virtualhost2
```

```
host2
127.0.0.1       localhost
10.68.91.226    virtualhost1.wdf.sap.corp virtualhost1
10.68.91.227    virtualhost2.wdf.sap.corp virtualhost2
```

Further Information

An SAP Community blog *How to configure HANA network communication channels – Part 1: Public network* gives further details of these features.

Related Information

Use the Command-Line Interface to Perform Platform LCM Tasks [page 588]
hostname
Rename an SAP HANA System Host [page 708]
Internal Host Name Resolution [page 747]
Host Name Resolution for System Replication [page 879]
Host Name Resolution for SQL Client Communication [page 750]
SAP Community Blog: How to configure HANA network communication channels – Part 1: Public network

11.2.4.2 Internal Host Name Resolution

SAP HANA services use IP addresses to communicate with each other. Host names are mapped to these IP addresses through internal host name resolution, a technique by which the use of specific and/or fast networks can be enforced and communication restricted to a specific network.

Single Host Versus Multiple Hosts

For single-host systems, no additional configuration is required. The services listen on the loopback interface only (IP address 127.0.0.1). In the `global.ini` files, the `[communication] listeninterface` is set to `.local` as follows:

```
global.ini
[communication]
listeninterface=.local
```

In a distributed scenario with multiple hosts, the network needs to be configured so that interservice communication is operational throughout the entire landscape. In this setup, the host names (these could be virtual host names) of all hosts must be known to each other and thus to the SAP HANA system. This can be achieved by manually adding all hosts to each `/etc/hosts` file on the operating system of each host.

A distributed system can run with or without a separate network definition for interservice communication.

Distributed System Without a Separate Internal Network

If no separate network is defined for internal communication, SAP HANA services listen on all available network interfaces. In the `global.ini` file, the listening interface is set to `.global` as follows:

```
global.ini
[communication]
listeninterface=.global
```

> ⚠ Caution
>
> If the `listeninterface` parameter is set to **`.global`**, we strongly recommend that you secure the SAP HANA servers with additional measures such as a firewall and/or TLS/SSL. Otherwise, the internal service ports of the system are exposed and can be used to attack SAP HANA.

Distributed System with a Separate Internal Network

A distributed system can be configured with a dedicated internal network in one of the following ways:

- At installation time, using the **HDBLCM command line option** as in the following example:

  ```
  /<installation medium>/DATA_UNITS/HDB_LCM_LINUX_X86_64/hdblcm --
  internal_network=10.66.128.0/20
  ```

- Post installation, using the **resident HDBLCM** from the GUI, command-line, or Web user interface. The following example, in command-line mode, binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry:

  ```
  /<sapmnt>/<SID>/hdblcm/hdblcm --action=configure_internal_network --
  listen_interface=internal --internal_address=10.66.8/21
  ```

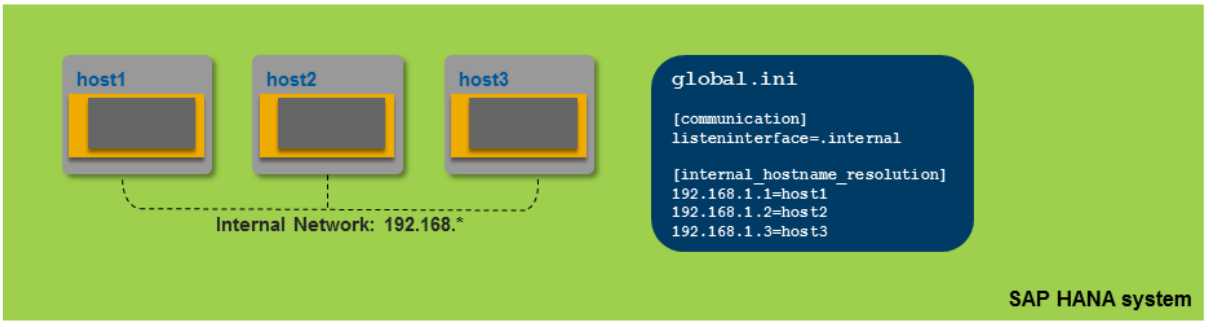
  For more information, see *Configuring SAP HANA Inter-Service Communication*.
  Post-installation configuration as described here is to be done by an SAP HANA system administrator with root credentials or on SAP HANA systems where SSH is configured. If root privileges or SSH are not available, you can still perform network configuration, but you need to use a host-by-host approach, also known as decentralized execution. In this case, see SAP Note 2048681.

SAP HANA automatically chooses on each host a network interface within the allowed network mask. If the network interface is defined as **`.internal`** in the `global.ini` file as described above, the SAP HANA services listen on this interface only:

```
global.ini
[communication]
listeninterface=.internal
```

Only the SAP start service (`sapstartsrv`) still listens on all interfaces, to accept start and stop commands, for example, from outside the SAP HANA system.

The following figure shows a simple example of how a separate internal network might be configured for an SAP HANA database with three hosts:



Simple Example of a Separate Internal Network for a Distributed SAP HANA System

For a more complex example, see *Host Name Resolution for System Replication*.

Further Information

For more information about configuring the network for multiple hosts, see the section on scaling SAP HANA.

For information about the security of internal networks, see the *SAP HANA Security Guide*.

> **Note**
>
> SAP HANA internal communication has sometimes been unofficially referred to as TREXNet communication. However, the term TREXNet is not valid in the context of SAP HANA.

An SAP Community blog *How to configure HANA network communication channels – Part 2: Internal network* gives further details of these features.

Related Information

Host Name Resolution for System Replication [page 879]
Configuring the Network for Multiple Hosts [page 1115]
Configuring SAP HANA Inter-Service Communication [page 1116]
Configure SAP HANA Inter-Service Communication Using the Command-Line Interface [page 1119]
SAP Note 2048681
Secure Internal Communication
Server-Side TLS/SSL Configuration Properties for Internal Communication
SAP Community Blog: How to configure HANA network communication channels – Part 2: Internal network

11.2.4.3 Host Name Resolution for SQL Client Communication

Client applications communicate with SAP HANA servers from different platforms and types of clients via a client library (such as SQLDBC, JDBC, ODBC, DBSL, ODBO or ADO.NET) for SQL or MDX access.

In distributed systems, the application has a **logical connection** to the SAP HANA system: that is, the client library may in fact use multiple connections to different servers or change to a different underlying connection. The client library supports load balancing and minimizes communication overhead by:

- Selecting connections based on load data
- Routing statements based on information about the location of data

> ⓘ Note
>
> Communication with SAP HANA hosts from a Web browser or a mobile application is requested using the HTTP protocol, which enables access to SAP HANA Extended Application Services, classic model (SAP HANA XS classic).

Public Host Name Resolution

An SQL client library always connects to the first available host specified in the connect string. From this host, the client library then receives a list of all the hosts. During operations, statements may be sent to any of these hosts.

By default, the IP address of the primary network interface is returned to the clients, as configured in the following parameter:

```
global.ini
[public_hostname_resolution]
use_default_route=ip
```

This works as long as there is only one external network. If a hostname or IP address is unresolvable, the client library falls back on the host names in the connect string:

- In single-host systems, the user doesn't normally notice this. In rare cases, the connection attempt does not fail immediately but waits for a TCP timeout, making the first statement run very slowly.
- In distributed systems, performance is impaired because statements must first be sent to the initial host and then forwarded on the server side to the right host.

Connect String with Multiple Host Names

In a distributed SAP HANA system consisting of more than one host, a list of hosts (host:port) is specified in the SQL client library connect string.

The connect string for JDBC, for example, could look like this:

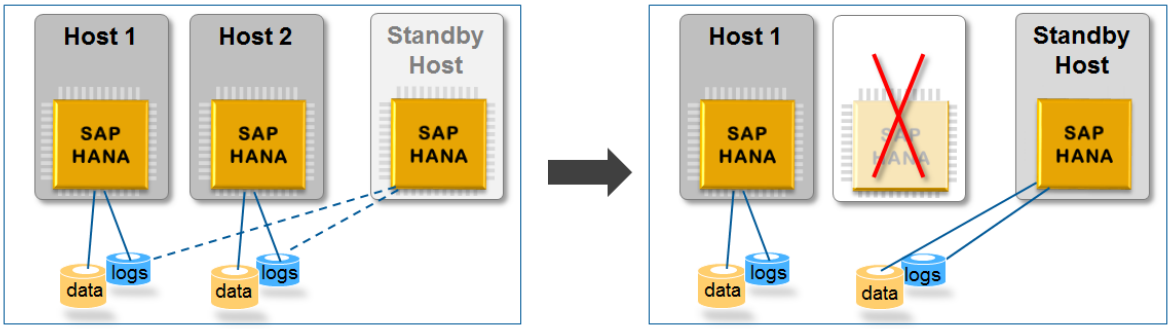
```
jdbc:sap://host1:30015;host2:30015;host3:30015/
```

All hosts that could become the active master, because they are one of the three configured master candidates, must be listed in the connect string to allow an initial connection to any of them in the event of a host auto-failover. A host auto-failover is an automatic switch from a crashed host to a standby host in the same system. One (or more) standby hosts are added to a SAP HANA system and configured to work in standby mode. As long as they are in standby mode, these hosts do not contain any data and do not accept requests or queries. When an active (worker) host fails, a standby host automatically takes its place.

Inclusion of the standby hosts in the connect string is mandatory if they are master candidates, otherwise optional.

The client connection code (ODBC/JDBC) uses a "round-robin" approach to reconnection, ensuring that the clients can always access the SAP HANA database, even after failover.

The following figure illustrates how host auto-failover works. An active host fails (in this example, Host 2), and the standby host takes over its role by starting its database instance using the persisted data and log files of the failed host.



Example of Auto Host-Failover

One way to look up the master candidates in your distributed SAP HANA database is to use the following SQL statement:

```
select HOST
from SYS.M_LANDSCAPE_HOST_CONFIGURATION
where NAMESERVER_CONFIG_ROLE like 'MASTER%'
order by NAMESERVER_CONFIG_ROLE
```

For more information, see the section on configuring clients for failover.

Connect String for SAP HANA System Replication

If system replication is used, we recommend that you do **not** specify physical host names in the SQL client connect string. Otherwise, you would have to reconfigure all of your applications after a takeover. Instead, use a **virtual host name** or **virtual IP address**, and manage it using an external cluster manager. This virtual host name or IP address must point to the active master host on the active primary site.

Further Information

System replication takeover hooks can be implemented to provide notification about the takeover. For more information about takeover hooks and client connection recovery, see the section on 'Implementing a HA/DR Provider' in System Replication.

An SAP Community blog *How to configure HANA network communication channels – Part 1: Public network* gives further details of these features.

Related Information

Mapping Host Names for Database Client Access [page 752]
SQL Connection Information for New Clients [page 754]
Configuring Clients for Failover [page 897]
Client Connection Recovery After Takeover [page 817]
Implementing a HA/DR Provider [page 905]
SAP Note 1780950
SAP Note 1876398
SAP Community Blog: How to configure HANA network communication channels – Part 1: Public network

11.2.4.3.1 Mapping Host Names for Database Client Access

Clients communicate with the database through external host names or external IP addresses. A default mapping of external host names to internal host names enables statement routing and automatic reconnection in the event of a failover.

By default, the IP address of the primary network interface is used but there may be situations where you need to change this configuration, such as for certain firewall configurations, network address translation (NAT) types, or multiple external networks. For this purpose, a `[public_hostname_resolution]` section in the `global.ini` file is used with:

```
use_default_route = ip # values: no,ip,name,fqdn
optional pattern mapping: map_<internal-prefix>* = <public-prefix>*<public-
suffix>
optional exact mapping: map_<internal-name> = <public-name>
```

If optional mappings exist, they are always considered regardless of the `use_default_route` parameter value. Exact mappings have higher priority than pattern mappings.

Each host identifies the network interface and thus the default route for the connection:

Description	Parameter	Example
IP address of the interface	*use_default_route* = **`ip`**	10.4.2.71

Description	Parameter	Example
Host name of the interface	*use_default_route* = **name**	lnd8520
Fully qualified name of the interface	*use_default_route* = **fqdn**	lnd8520.lnd.abc.corp **Example** For connections to tenant databases, certificate validation may not work due to how SAP HANA handles host name resolution. If this is the case, setting the value of the parameter `use_default_route` to **fqdn** on the SYSTEM layer ensures that SAP HANA uses the FQDN and that certificate validation for secured JDBC/ODBC connections is allowed.
Disable feature and use internal host name	*use_default_route* = **no**	hananode01

In most cases, you do not need to configure anything. If you do need to configure something, see if you can use one of the default route mechanisms. You need to specify your own mapping only if the default route mechanisms do not fit your network requirements.

Example

Here are some examples of how you might customize this parameter:

```
[public_hostname_resolution]
map_hananode* = myservername*
```

```
[public_hostname_resolution]
map_hananode* = hananode*.lnd.abc.corp
```

```
[public_hostname_resolution]
map_hananode01 = 10.4.2.71
map_hananode02 = 10.4.2.72
map_hananode03 = 10.4.2.73
map_hananode04 = 10.4.2.74
```

```
[public_hostname_resolution]
map_hananode0* = 10.4.2.7*
map_hananode1* = 10.4.2.8*
```

Related Information

Host Name Resolution for SQL Client Communication [page 750]

11.2.4.3.2 SQL Connection Information for New Clients

It can be convenient for new SQL clients to be able to query the connectivity information of an existing client.

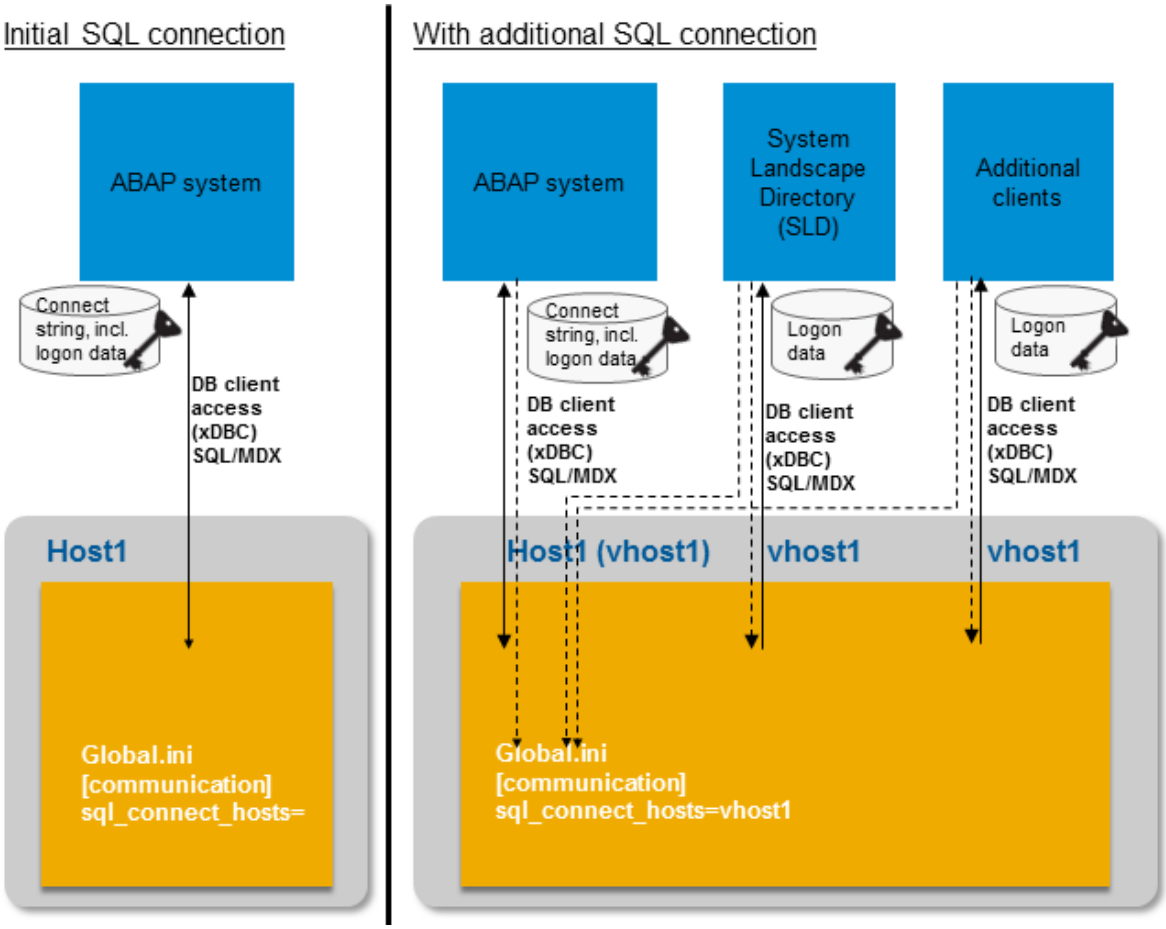
The connect string of the existing client was stored in the secure store and cannot be accessed. However, you can use the `global.ini/[communication]/sql_connect_hosts` parameter to record the connectivity information in the SAP HANA server so that it is available for the database connection from new clients. This information is a list of host names or IP addresses, which could be virtual host name or IP addresses, separated by commas.

The SAP HANA server does not use this parameter. It is used by applications and components that connect to SAP HANA. If the parameter is not filled, the application needs to consume the host values as follows:

```
select HOST
from SYS.M_LANDSCAPE_HOST_CONFIGURATION
where NAMESERVER_CONFIG_ROLE like 'MASTER%'
order by NAMESERVER_CONFIG_ROLE
```

In the following example, an ABAP system is installed on SAP HANA and the connection information is stored on the client side in the connect string including the logon data. This is the standard case. The ABAP client then sets the `sql_connect_hosts` parameter on the SAP HANA server. A System Landscape Directory (SLD) is subsequently installed on the same SAP HANA system. The SLD agent is able to look up the parameter in SAP HANA to find out the connection information. If the parameter values are missing, SLD uses the above SQL statement. If more clients are added, they follow the same procedure.

The example shows a single host but the parameter can also be useful in scenarios with multiple hosts.


Initial SQL connection
ABAP system
Connect string, incl. logon data
DB client access (xDBC) SQL/MDX
Host1
Global.ini [communication] sql_connect_hosts=
With additional SQL connection
ABAP system
System Landscape Directory (SLD)
Additional clients
Connect string, incl. logon data
Logon data
Logon data
DB client access (xDBC) SQL/MDX
DB client access (xDBC) SQL/MDX
DB client access (xDBC) SQL/MDX
Host1 (vhost1)
vhost1
vhost1
Global.ini [communication] sql_connect_hosts=vhost1

12 Availability and Scalability

SAP HANA provides comprehensive fault and disaster recovery support, as well as high availability for business continuity.

Related Information

High Availability for SAP HANA [page 756]
SAP HANA Database Backup and Recovery [page 920]
Scaling SAP HANA [page 1079]

12.1 High Availability for SAP HANA

High availability is the name given to a set of techniques, engineering practices, and design principles that support the goal of business continuity and also ensure that data and services are available to authorized users when needed.

SAP HANA is fully designed for high availability. It supports recovery measures ranging from faults and software errors, to disasters that decommission an entire data center. High availability is achieved by eliminating single points of failure (fault tolerance), and providing the ability to rapidly resume operations after a system outage with minimal business loss (fault resilience). Fault recovery is the process of recovering and resuming operations after an outage due to a fault. Disaster recovery is the process of recovering operations after an outage due to a prolonged data center or site failure. Preparing for disasters may require backing up data across longer distances, and may thus be more complex and costly.

The key to achieving high availability is redundancy, including hardware redundancy, network redundancy and data center redundancy. SAP HANA provides several levels of defense against failure-related outages:

- Hardware Redundancy: SAP HANA appliance vendors offer multiple layers of redundant hardware, software and network components, such as redundant power supplies and fans, enterprise grade error-correcting memories, fully redundant network switches and routers, and uninterrupted power supply (UPS). Disk storage systems use batteries to guarantee writing even in the presence of power failure, and use striping and mirroring to provide redundancy for automatic recovery from disk failures. Generally speaking, all these redundancy solutions are transparent to SAP HANA's operation, but they form part of the defense against system outage due to single component failures.
- Software: Linux distributions certified for SAP HANA include security pre-configurations (for example, minimal network services). Additionally, the SAP HANA system software also includes a watchdog function, which automatically restarts configured services (index server, name server, and so on), in case of detected stoppage (killed or crashed).
- Persistence: SAP HANA persists transaction logs, savepoints and snapshots to support system restart and recovery from host failures, with minimal delay and without loss of data.

- Standby and Failover: Separate, dedicated standby hosts are used for failover, in case of failure of the primary, active hosts. This improves the availability by significantly reducing the recovery time from an outage.

Related Information

SAP HANA High Availability Support [page 757]
Operation Modes for SAP HANA System Replication [page 773]
Replication Modes for SAP HANA System Replication [page 771]
Active/Active (Read Enabled) [page 852]
Configuring SAP HANA System Replication [page 792]
Performing a Takeover [page 815]
Performing a Failback [page 832]
Disabling SAP HANA System Replication [page 833]
Setting Up Host Auto-Failover [page 896]
SAP HANA System Replication with Tenant Databases [page 769]
Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 885]
Monitoring Secondary Systems [page 867]

12.1.1 SAP HANA High Availability Support

As an in-memory database, SAP HANA is not only concerned with maintaining the reliability of its data in the event of failures, but also with resuming operations with most of that data loaded back in memory as quickly as possible.

Downtime is the consequence of outages, which may be intentional (for example, for system upgrades) or caused by unplanned faults. A fault can be due to equipment malfunction, software or network failures, or due to a major disaster such as a fire, a regional power loss or a construction accident, which may decommission the entire data-center.

Fault Recovery is the process of recovering and resuming operations after an outage due to a fault. Disaster Recovery is the process of recovering operations after an outage due to a prolonged datacenter or site failure. Preparing for disasters may require backing up data across longer distances, and may thus be more complex and costly.

SAP HANA supports the following recovery measures from failures:

- Disaster recovery support:
  - Backups: Periodic saving of database copies in safe place.
  - Storage replication: Continuous replication (mirroring) between primary storage and backup storage over a network (may be synchronous).
  - System replication: Continuous update of secondary systems by primary system, including in-memory table loading.
- Fault recovery support:

- Service auto-restart: Automatic restart of stopped services on host (watchdog).
- Host auto-failover: Automatic failover from crashed host to standby host in the same system.
- System replication: Continuous update of secondary systems by primary system, including in-memory table loading and read-only access on the secondary.

System replication is flexible enough that it can also be used for both fault and disaster recovery to achieve high availability. The data pre-load option can be used for fault recovery to enable a quicker takeover than with Host Auto-Failover. You can build a solution with single node systems and do not need a scale out system and the additional storage and associated costs.

SAP HANA supports system replication for tenant databases on the system level, this means the tenant database system as a whole including all tenant databases. An SAP HANA system installed in multiple-container mode always has exactly one system database and any number of tenant databases (including zero). For more information on tenant databases see *Creating and Configuring Tenant Databases*.

Using Secondary Servers for Non-Production systems

With SAP HANA system replication, you can use the servers on the secondary system for non-production SAP HANA systems under the following conditions:

- Table pre-load is turned off in the secondary system.
- The secondary system uses its own disk infrastructure. In the case of single node systems this means, the local disk infrastructure needs to be doubled.
- The non-production systems are stopped with the takeover to the production secondary.

Related Information

Creating and Configuring Tenant Databases [page 65]
SAP Note 1999880
SAP Note 2183363
SAP Note 2300936
SCN Documents
SAP HANA Academy System Replication Videos
White paper "Introduction to High Availability for SAP HANA"
How to Perform System Replication for SAP HANA

12.1.1.1 Backups

Backups are one of the key disaster recovery features offered by SAP HANA.

SAP HANA uses in-memory technology, but of course it fully persists any transaction that changes the data, such as row insertions, deletions and updates, so it can resume from a power-outage without loss of data. SAP HANA persists two types of data to storage: transaction redo logs, and data changes in the form of savepoints.

A transaction redo log is used to record a change. To make a transaction durable, it is not required to persist the complete data when the transaction is committed; instead it is sufficient to persist the redo log. Upon an outage, the most recent consistent state of the database can be restored by replaying the changes recorded in the log, redoing completed transactions and rolling back incomplete ones.

A savepoint is a periodic point in time, when all the changed data is written to storage, in the form of pages. One goal of performing savepoints is to speed up restart: when starting up the system, logs need not be processed from the beginning, but only from the last savepoint position. Savepoints are coordinated across all processes (called SAP HANA services) and instances of the database to ensure transaction consistency. By default, savepoints are performed every five minutes, but this can be configured.

Savepoints normally overwrite older savepoints, but it is possible to freeze a savepoint for future use; this is called a snapshot. Snapshots can be replicated in the form of full data backups, which can be used to restore a database to a specific point in time. This can be useful in the event of data corruption, for instance. In addition to data backups, smaller periodic log backups ensure the ability to recover from fatal storage faults with minimal loss of data.

Savepoints, can be saved to local storage, and the additional backups, can be additionally saved to backup storage. Local recovery from the crash uses the latest savepoint, and then replays the last logs, to recover the database without any data loss. If the local storage was corrupted by the crash, it is still possible to recover the database from the data and log backups, possibly with loss of some data. Regularly shipping backups to a remote location over a network or via couriers can be a simple and relatively inexpensive way to prepare for a disaster. Depending on the frequency and shipping method, this approach may have a recovery time ranging from hours to days.

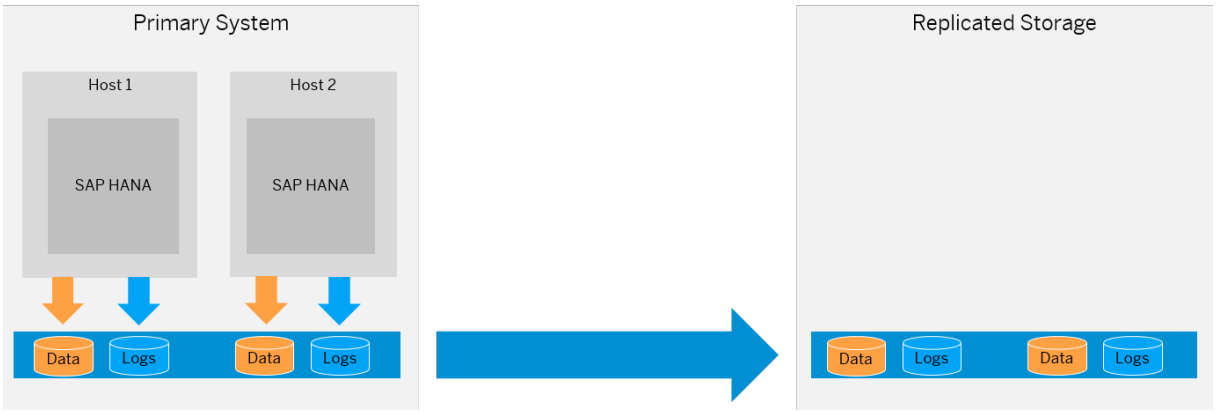
Related Information

SAP HANA Database Backup and Recovery [page 920]

12.1.1.2 Storage Replication

SAP HANA offers disaster recovery support for storage replication solutions provided by hardware partners.

One drawback of backups is the potential loss of data between the time of the last backup and the time of the failure. A preferred solution therefore, is to provide continuous replication of all persisted data. Several SAP HANA hardware partners offer a storage-level replication solution, which delivers a backup of the volumes or file-system to a remote, networked storage system. In some of these vendor-specific solutions, which are certified by SAP, the SAP HANA transaction only completes when the locally persisted transaction log has been replicated remotely. This is called synchronous storage replication. Synchronous storage replication can be used only where the distance between the primary and backup site is relatively short (typically 100 kilometers or less), allowing for sub-millisecond round-trip latencies.



Due to its continuous nature, storage replication (sometimes also called remote storage mirroring) can be a more attractive option than backups, as it reduces the amount of time between the last backup and a failure. Another advantage of storage replication is that it also enables a much shorter recovery time. This solution requires a reliable, high bandwidth and low latency connection between the primary site and the secondary site.

See *SAP Note 1755396 Released DT solutions for SAP HANA with disk replication*

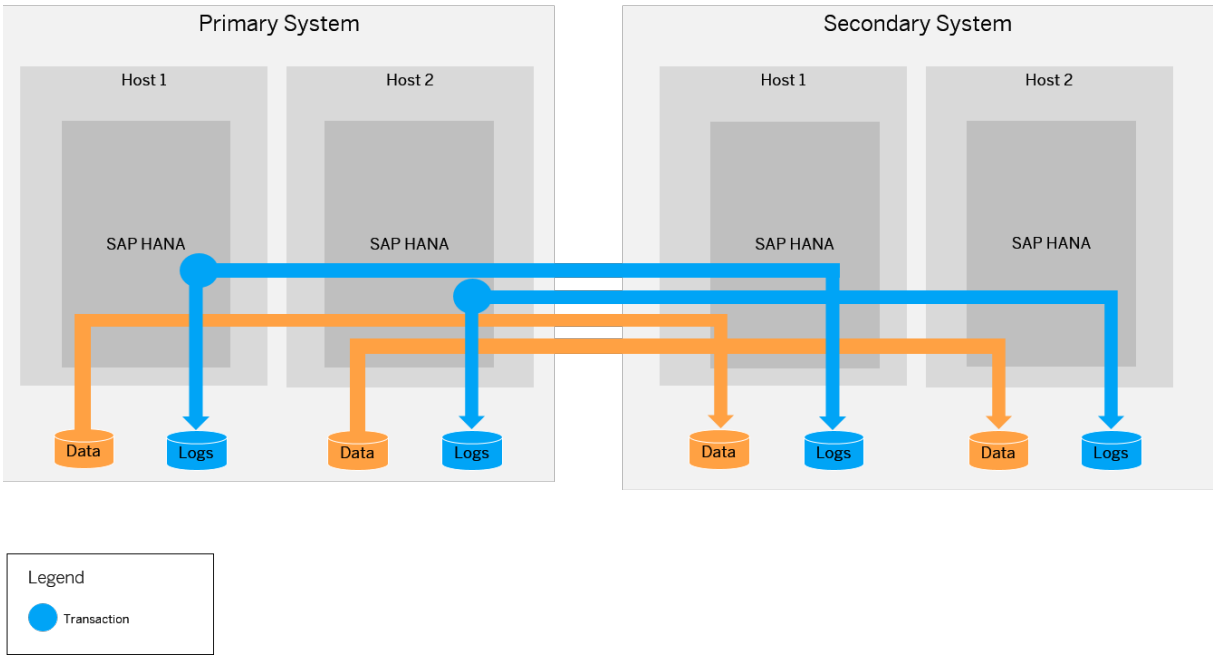
Related Information

SAP Note 1755396

12.1.1.3 System Replication

System replication is available in every SAP HANA installation offering an inherent disaster recovery support.

System replication is set up so that a secondary system is configured as an exact copy of the active primary system, with the same number of active hosts in each system. The number of standby hosts need not be identical. With multitier system replication you can have a third system attached to the first secondary making it a replication chain of three systems. Each service instance of the primary SAP HANA system communicates with a counterpart in the secondary system. With multitarget system replication the primary system can replicate data changes to more than one secondary system.



The secondary system can be located near the primary system to serve as a rapid failover solution for planned downtime, or to handle storage corruption or other local faults, or, it can be installed in a remote site to be used in a disaster recovery scenario. Also both approaches can be chained together with multitier system replication. Like storage replication, this disaster recovery option requires a reliable connection channel between the primary and secondary sites. The instances in the secondary system operate in recovery mode. In this mode, all secondary system services constantly communicate with their primary counterparts, replicate and persist data and logs, and load data to memory. The main difference to primary systems is that the secondary systems do not accept requests or queries.

When the secondary system is started in recovery mode, each service component establishes a connection with its counterpart, and requests a snapshot of the data in the primary system. From then on, all logged changes in the primary system are replicated. Whenever logs are persisted in the primary system, they are also sent to the secondary system. A transaction in the primary system is not committed until the logs are replicated. What this means in detail, can be configured by choosing one of the log replication modes. For an overview of the replication modes, see *Replication Modes for SAP HANA*.

If the connection to the secondary system is lost, or the secondary system crashes, the primary system after a brief, configurable, timeout will resume replication. The secondary system persists, but does not immediately replay the received log. To avoid a growing list of logs, incremental data snapshots are transmitted asynchronously from time to time from the primary system to the secondary system. If the secondary system has to take over, only that part of the log needs to be replayed that represents changes that were made after the most recent data snapshot. In addition to snapshots, the primary system also transfers status information regarding which table columns are currently loaded into memory. The secondary system correspondingly preloads these columns. In the event of a failure that justifies full system takeover, an administrator instructs the secondary system to switch from recovery mode to full operation. The secondary system, which already preloaded the same column data as the primary system, becomes the primary system by replaying the last transaction logs, and then starts to accept queries.

ⓘ Note

To prevent unauthorized access to the SAP HANA database, the internal communication channels between the primary site and the secondary site in a system replication scenario need to be protected. This may

include filtering access to the relevant ports and channels by firewalls, implementing network separation, or applying additional protection at the network level (for example, VPN, IPSec). We recommend routing the connection between the two sites over a special site-to-site high-speed network, which typically already implements security measures such as separation from other network access and encryption or authentication between sites. The details of security measures and implementation of additional network security measures depend on your specific environment. For more information about network and security aspects, see the *SAP HANA Master Guide* and the *SAP HANA Security Guide*.

For a complete overview on system replication or for examples, see the *SAP HANA System Replication* guide.

Related Information

Overview of Steps [page 791]
Recovery with System Replication [page 1062]
Replication Modes for SAP HANA System Replication [page 771]

12.1.1.4 Service Auto-Restart

Service auto-restart supports fault recovery for one service.

In the event of a software failure or an intentional intervention by an administrator that disables one of the configured SAP HANA services (Index Server, Name Server, and so on), the service will be restarted by the SAP HANA service auto-restart watchdog function, which automatically detects the failure and restarts the stopped service process. Upon restart, the service loads data into memory and resumes its function. While all data remains safe the service recovery takes some time.

12.1.1.5 Host Auto-Failover

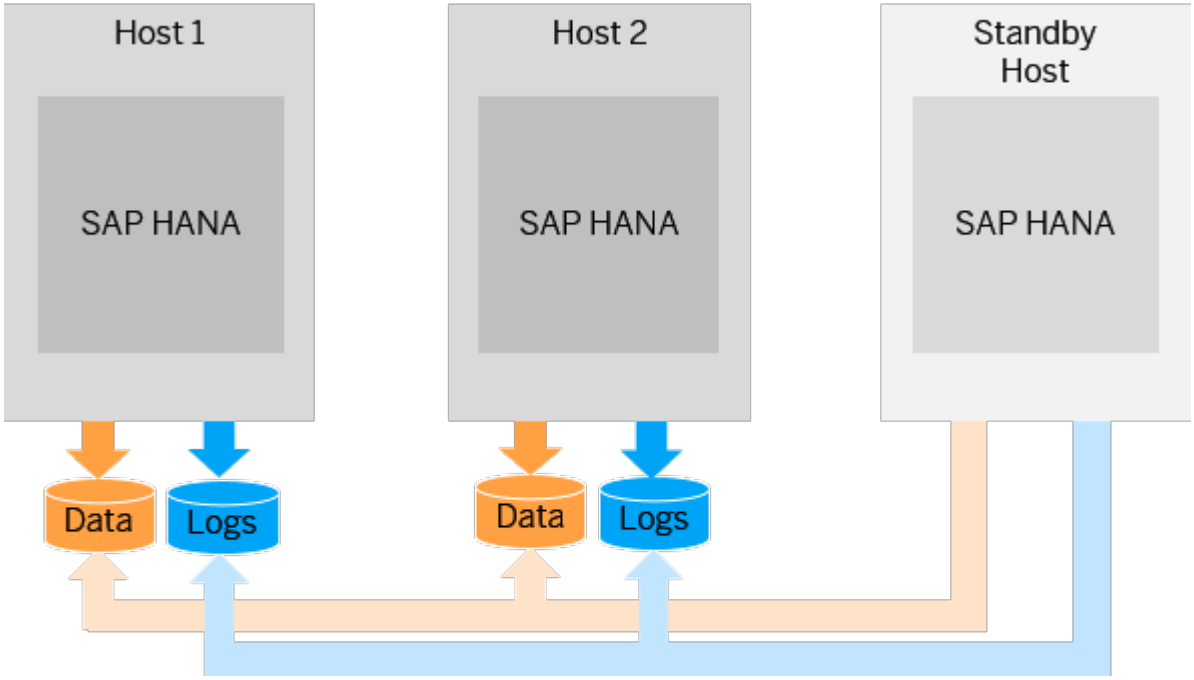
Host auto-failover supports fault recovery for a failed host.

Host auto-failover is a local fault recovery solution that can be used in addition or as an alternative measure to system replication. One (or more) standby hosts are added to an SAP HANA system, and configured to work in standby mode. As long as they are in standby mode the databases on these hosts do not contain any data and do not accept requests or queries. This means they cannot be used for other purposes such as quality or test systems.

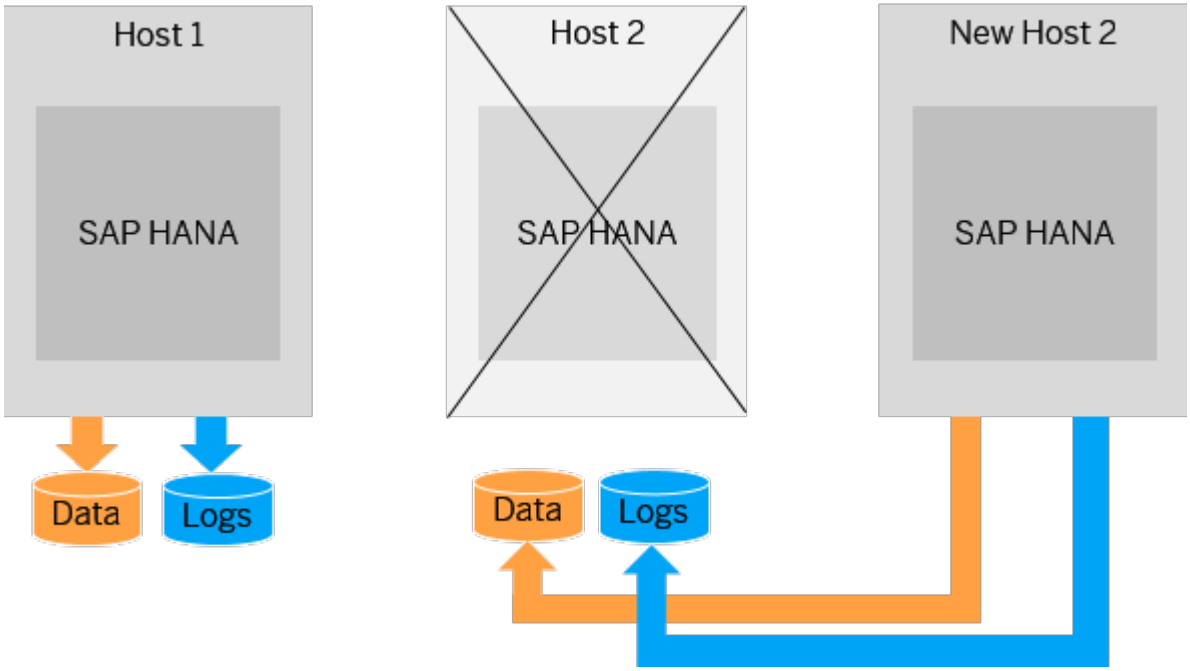
> ⓘ Note
>
> **Alternatives to Host Auto-Failover**: Host Auto-Failover requires shared network storage so that data and log files can be loaded from any server. However, Host Auto-Failover is not necessary in all systems (such as test systems or when other high availability solutions such as SAP HANA System Replication are used) and in these cases simpler storage options for the HANA data and log files can be used. Some alternative options, such as installing an NFS server on the master host, are described in a separate document in the SAP Community: *Storage alternatives for HANA scale-out without Host Auto-Failover*.

When an active (worker) host fails, a standby host automatically takes its place. If neither the name server process hdbnameserver nor hdbdaemon respond to network requests (because the instance is stopped, the OS has been shut down or powered off), a host is marked as inactive and an auto-failover is triggered. Since the standby host may take over operation from any of the primary hosts, it needs shared access to all the database volumes. This can be accomplished by a shared, networked storage server, by using a distributed file system, or with vendor-specific solutions that use an SAP HANA programmatic interface, the Storage Connector API, to dynamically detach and attach (mount) networked storage upon failover.

This scenario is illustrated in the graphic below:



Once repaired, the failed host can rejoin the system as the new standby host to reestablish the failure recovery capability:



In support of host auto-failover, database clients can be configured with the connection information of multiple hosts, optionally including the standby host. The client connection code (ODBC, JDBC, and so on) will try to connect to one of these, and upon successful connection receives the updated connection configurations. This ensures that clients can continue to reach the SAP HANA database, even after failover.

> ⓘ Note
>
> In a scale-out system, a failover from the failing master host is triggered automatically even if no standby host is configured. Then a worker host is attached to the volumes of the previous master.

> ⓘ Note
>
> It is not possible to do a seamless failover. A manual failover to a standby host can be triggered by stopping a worker host using the command HDB stop.

Related Information

Setting Up Host Auto-Failover [page 896]
SAP HANA - Host Auto-Failover (SAP Community Document)
Storage alternatives for HANA scale-out without Host Auto-Failover (SAP Community Document)
SAP Note 2407186

12.1.2 SAP HANA System Replication

SAP HANA system replication is a mechanism for ensuring the high availability of your SAP HANA system.

Through the continuous replication of data from a primary to a secondary system, including in-memory loading, system replication facilitates rapid failover in the event of a disaster. Productive operations can be resumed with minimal downtime.

The following administration activities are possible using the SAP HANA cockpit, using the SAP HANA studio, or using hdbnsutil on the command line:

- Performing the initial set-up, that is enabling system replication and establishing the connection between two identical systems
- Monitoring the status of system replication to ensure that both systems are in sync
- Triggering takeover by the secondary system in the event of a disaster and failback once the original system is available again
- Disabling system replication

System replication can also be set up or managed using SAP Landscape Management (SAP LaMa) - see related information links below.

For a complete overview of system replication including illustrations and examples, see the *SAP HANA System Replication* guide.

Related Information

SAP HANA System Replication
SAP Landscape Management (SAP LaMa) Documentation
Orchestrated nZDM for SAP HANA with a single TakeOver (Landscape Management blog)

12.1.2.1 General Prerequisites for Configuring SAP HANA System Replication

Before you configure SAP HANA system replication, several prerequisites must be fulfilled.

System Requirements

- The primary and secondary systems must both be installed and correctly configured; verify that both systems are independently up and running.
- SAP HANA systems can only be replicated as a whole, which means that the system database and all tenant databases are part of the system replication. A takeover can only be performed as a whole system. A takeover on the level of a single tenant database isn't possible.

- Create an initial data backup or storage snapshot. In multiple-container systems, the system database and all tenant databases must be backed up. For more information, see *Creating Backups*.
- The configuration of active hosts in the primary and secondary system must be the same, which means that the number of active hosts, the names of the host roles, failover groups, and worker groups must be identical in both systems. So, if there's a standby host on the primary system it need not be available on the secondary system and vice-versa.
- System replication between two systems on the same host isn't supported. Check that the host names in the primary system are different to the host names used in the secondary system. You can see the SAP HANA host name for each host in the environment variable SAP_RETRIEVAL_PATH (`/usr/sap/<SID>/HDB<InstNo>/<hostname>`) and with the python script `landscapeHostConfiguration.py`. For more information, see *Host Name Resolution for System Replication* and *Checking the Status with landscapeHostConfiguration.py* in the *SAP HANA Administration Guide*.
  If the host names of the primary and the secondary system are the same (for example, because two systems are used that have identical host names), change the host names used on the secondary system. For more information, see *Rename an SAP HANA System Host* in the *SAP HANA Administration Guide*.
- To secure the system replication communication channel between the primary and the secondary system, configure the ini parameters `[system_replication_communication] listeninterface` and `allowed_sender` as described in *Host Name Resolution for System Replication*.
- For SAP HANA system replication, a port offset value of 10000 is configured to reserve ports for system replication communication.

  > **Note**
  >
  > Note that values for port ranges do not need to be maintained manually. This can be done automatically by the SAP Host Agent which includes port reservation functionality and optimizes the relevant Linux kernel parameters. Refer to Linux Kernel Parameters in the Lifecycle Management section of the *SAP HANA Administration Guide* and the following SAP Notes:
  >
  > - *401162 - Linux: Avoiding TCP/IP port conflicts and start problems*, this describes setting up the SAP Host Agent.
  > - *2382421 - Optimizing the Network Configuration on HANA- and OS-Level*

- If the local secure store (LSS) is being used as the secure store:
  - The primary and secondary systems require completely independent installations of the LSS with different file shares.
  - If the primary system has an active key management configuration with an external key management system (KMS), it is only possible to register secondary systems that use the LSS (and not the secure store in the file system (SSFS)). After successful replication, all secondary tenant databases are automatically connected to the same KMS account as the primary with the same key management configuration. For more information, see *Using the LSS with an External Key Management System*.
- If a new tenant database is created in a running SAP HANA system replication, it must be backed up to participate in the replication. Afterward, the initial data shipping is started automatically for this tenant database. If a takeover is done while the initial data shipping is running and not finished, this new tenant database won't be operational after takeover and will have to be recovered with backup and recovery. See the *SAP HANA Database Backup and Recovery* section of the *SAP HANA Administration Guide*.
- The secondary system must have the same SAP system ID (SID) and `instance number` as the primary system.

> **Note**
>
> The primary system replicates all relevant license information to the secondary system. An additional license isn't required. For more information, see *SAP Note 2211663 The license changes in an SAP HANA database after the deregistration of the secondary site*.

- During an upgrade of the system replication landscape, the software version of the current secondary system must be equal or newer to the version of the current primary system.

> **Note**
>
> During a failback, the roles of your systems in the system replication landscape switch. Make sure in this case that your primary system doesn't have a newer software version than the secondary system.

> **Note**
>
> For Active/Active (read enabled) setups, the SAP HANA versions must be the same on the primary and the secondary system. Use this setup mainly during the upgrade process of the system replication landscape.

Endianness

In a system replication landscape the systems on all sites must run on platforms with the same byte order. System replication is supported between Intel little-endian and IBM Power little-endian systems.

In SAP HANA the following byte order is supported in the corresponding SAP HANA and Linux versions:

Supported Byte Order

SAP HANA Version	Linux OS Version	Byte Order
SAP HANA 1.0 SPS 11, SPS 12	Linux Intel SLES 11	little-endian
	Linux Power SLES 11	big-endian
SAP HANA 2.0 SPS 00	Linux Intel SLES 12	little-endian
	Linux Power SLES 12	

System replication is supported between Intel little-endian (SAP HANA 1.0 SPS 12 or SAP HANA 2.0 SPS 00) and IBM Power little-endian (SAP HANA 2.0 SPS 00).

Configuration

- All configuration steps have to be executed on the coordinator nameserver node only.
- The .ini file configuration must be similar for both systems. Any changes made manually, or by SQL commands on one system should be manually duplicated on the other system.

Automatic configuration parameter checks alert you to configuration differences between the two systems.

> **Note**
>
> To keep the ini file configuration similar on both systems, the INI parameter checker is per default configured to check for differences. Additionally, it can be configured to replicate parameter changes from the primary system to the secondary system.

- Ensure that `log_mode` is set to **`normal`** in the `persistence` section of the global.ini file. In log mode normal, the log segments are automatically backed up. This ensures that the database can be backed up to the most recent point in time.

Authorization

- You must be logged on to both systems as the operating system user (user `<sid>`adm) or you have provided its credentials when prompted.
- You need the operating system user to set up a system replication landscape, to perform a takeover and a failback, as well as to disable system replication with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* and *Connect to a Database With SSO or SAP HANA Credentials*.
- Before you configure SAP HANA system replication, you must copy the system PKI SSFS .key and the .dat file from the primary system to the secondary system:
  `/usr/sap/<SID>/SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT`
  `/usr/sap/<SID>/SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY`
  For more information, see *SAP Note 2369981 Required configuration steps for authentication with HANA System Replication*.
  If you installed XS advanced, you must also copy the XSA SSFS .key and the .dat file from the primary system to the secondary system in the following directories:
  `/usr/sap/<SID>/SYS/global/xsa/security/ssfs/data/SSFS_<SID>.DAT`
  `/usr/sap/<SID>/SYS/global/xsa/security/ssfs/key/SSFS_<SID>.KEY`
  For more information, see *SAP Note 2300936 Host Auto-Failover & System Replication Setup with SAP HANA extended application services, advanced model*.
  The copied files become active during system restart. Therefore, it's recommended to copy them when the secondary system is offline (for example, before registration).
- In preparation for maintenance tasks (for example, near zero downtime upgrades), configure a user in the SAP HANA user store (hdbuserstore) under the SRTAKEOVER key. For more information, see *Configure a User Under the SRTAKEOVER Key* in the *SAP HANA Administration Guide*.

Dynamic Tiering

- If you plan to add SAP HANA dynamic tiering to your landscape in the future, see*SAP Note 2447994* before you enable HANA system replication. SAP HANA dynamic tiering requires certain communication ports, operation modes, and replication modes.
- SAP HANA dynamic tiering isn't supported with multitarget system replication. For more information about SAP HANA system replication with SAP HANA dynamic tiering, see *SAP Note 2447994*.

Related Information

Host Name Resolution for System Replication [page 879]
Rename an SAP HANA System Host [page 708]
Secure Internal Communication Between Sites in System Replication Scenarios
SAP HANA Database Backup and Recovery [page 920]
Using the LSS with an External Key Management System [page 508]
Copying and Moving Tenant Databases [page 671]
Configure a User Under the SRTAKEOVER Key [page 884]
Operating System User <sid>adm [page 552]
Checking the Status with landscapeHostConfiguration.py
SAP Note 2211663
SAP Note 2369981
SAP Note 2300936
SAP Note 2447994
SAP Note 401162
SAP Note 2382421

12.1.2.2 SAP HANA System Replication with Tenant Databases

The usual SAP HANA system replication principles apply for tenant database systems.

SAP HANA supports system replication for tenant databases on the system level, this means the tenant database system as a whole including all tenant databases. An SAP HANA system installed in multiple-container mode always has exactly one system database and any number of tenant databases (including zero).

Before you begin preparing a replication strategy for an SAP HANA system, you should be aware of the following important points:

- SAP HANA systems can only be replicated as the whole system. This means that the system database and all tenant databases are part of the system replication. A takeover can only be performed as a whole system. A takeover on the level of a single tenant database is not possible.
- Tenant management actions are synchronized over the whole replication topology so that an action (such as create, drop, start, stop) executed on a tenant on the primary is also performed on the secondary.
- If an active tenant database is stopped in a running SAP HANA system replication, it is stopped on the secondary site as well. If a takeover is done while tenant databases (which were part of the system replication) are stopped, they will be in the same state after takeover as they were on the primary site when they were stopped. The tenant databases must be started to complete the takeover.
- If a new tenant database is created in a configured SAP HANA system replication, it must be backed up to participate in the replication. Afterwards, the initial data shipping is started automatically for this tenant database. If a takeover is done while the initial data shipping is running and not finished, this new tenant database will not be operational after takeover and will have to be recovered with backup and recovery (see the *SAP HANA Database Backup and Recovery* section of the *SAP HANA Administration Guide*).

- If SAP HANA system replication runs in replication mode SYNC with the full sync option enabled, and if the connection to the secondary site is interrupted, no write operations on the primary site are possible. The operation of creating a tenant database, for example, will wait until the connection to the secondary is reestablished or the SQL statement times out.
- With SAP HANA systems, the services needed are generated automatically in the tenant databases.
- For SAP HANA system replication, a port offset value of 10000 is configured to reserve ports for system replication communication.

  > ### Note
  >
  > Values for port ranges do not need to be maintained manually. This can be done automatically by the SAP Host Agent which includes port reservation functionality and optimizes the relevant Linux kernel parameters. Refer to 'Linux Kernel Parameters' in the Lifecycle Management section of the *SAP HANA Administration Guide* and the following SAP Notes:
  >
  > - *401162 - Linux: Avoiding TCP/IP port conflicts and start problems*, this describes setting up the SAP Host Agent.
  > - *2382421 - Optimizing the Network Configuration on HANA- and OS-Level*

- For SAP HANA systems running with the HIGH isolation level, the system PKI SSFS data and key file must be copied from the primary system to the same location on the secondary system(s). For more information, see *Increase the System Isolation Level* in the *SAP HANA Administration Guide*.

For more information on the individual points, see the *Availability and Scalability* section of the *SAP HANA Administration Guide*.

Related Information

Availability and Scalability [page 756]
SAP HANA Database Backup and Recovery [page 920]
Copying and Moving Tenant Databases [page 671]
Increase the System Isolation Level [page 78]
Linux Kernel Parameters [page 609]
SAP Note 401162
SAP Note 2382421

12.1.2.3 Replication Modes for SAP HANA System Replication

While registering the secondary system, you need to decide which replication mode to use.

SAP HANA offers different modes for the replication of the redo log:

Replication modes

Log Replication Mode	Description
Synchronous in-memory (SYNCMEM)	The primary system commits the transaction after it gets a reply that the log was received by the secondary system and stored in memory. The delay for the transaction in the primary system is smaller, because it only includes the time for transmitting the data. The disk I/O speed on the secondary system doesn't influence the primary's performance. When the connection to the secondary system is lost, the primary system continues the transaction processing and writes the changes only to the local disk. Data loss can occur in the following situations: • When the primary and the secondary system fail at the same time while the secondary system is connected. The data is not written to disk – neither on the primary nor on the secondary system. • When a takeover is executed while the secondary system is unavailable. The data that arrived on the secondary is outdated compared to the data on the primary. This option provides better performance because it is not necessary to wait for disk I/O on the secondary system, but it is more vulnerable to data loss.

Log Replication Mode	Description
Synchronous on disk (SYNC)	The primary system waits with committing the transaction until it gets a reply that the log is persisted in the secondary system. This option guarantees immediate consistency between both systems, at a cost of delaying the transaction by the time for data transmission and persisting in the secondary system. When the connection to the secondary system is lost, the primary system continues the transaction processing and writes the changes only to the local disk. No data loss occurs in this scenario as long as the secondary system is connected. Data loss can occur, when a takeover is executed while the secondary system is disconnected. Additionally, this replication mode can run with a full sync option. This means that log write is successful when the log buffer has been written to the log file of the primary and the secondary system. When the secondary system is disconnected (for example, because of network failure), the primary system suspends the transaction processing until the connection to the secondary system is reestablished. No data loss occurs in this scenario. You can set the full sync option for system replication with the parameter `[system_replication]/enable_full_sync`. **Note** If SAP HANA system replication runs in the SYNC replication mode with the full sync option enabled, and if the connection to the secondary site is interrupted, no write operations on the primary site are possible. The operation of creating a tenant database, for example, will wait until the connection to the secondary is reestablished or the SQL statement times out. For more information about how to enable the full sync option, see *Full Sync Option for System Replication*.
Asynchronous (ASYNC)	The primary system commits the transaction after sending the log without waiting for a response. Here we have no delay because the data transmission is asynchronous to the transaction in the primary system. This option provides better performance because it is not necessary to wait for log I/O on the secondary system. Database consistency across all services on the secondary system is guaranteed. However, it is more vulnerable to data loss. Data changes may be lost during takeover.

Note

If you plan to add SAP HANA dynamic tiering to your landscape in the future, please check supported replication modes in *SAP Note 2447994* before you enable SAP HANA system replication.

The replication mode can be changed without going through a full data shipping from the primary system to the secondary system afterwards. For more information, see *Changing the Replication Mode*.

Related Information

Full Sync Option for SAP HANA System Replication [page 800]
Changing the Replication Mode [page 797]

12.1.2.4 Operation Modes for SAP HANA System Replication

While registering the secondary system, you need to decide in which operation mode to run SAP HANA system replication.

System replication can be run in three operation modes: `delta_datashipping`, `logreplay` or `logreplay_readaccess`. Depending on the configured operation mode, the database sends different types of data packages to the secondary system. For more information, see *Data Transferred to the Secondary System*.

Operation Mode	Description
delta_datashipping	This mode establishes a system replication where occasionally (per default every 10 minutes) a delta data shipping takes place in addition to the continuous log shipping. The secondary system persists the received log entries but it does not replay them until it has to take over. To shorten the log replay time, data snapshots are transmitted from time to time from the primary to the secondary system. The data snapshots are transferred asynchronously as differential backups (data backup deltas) triggered by the secondary system, which asks for a data backup delta with changes since the last one. During takeover the redo log needs to be replayed up to the last arrived delta data shipment.
logreplay	In this operation mode, a redo log shipping is done after system replication was initially configured with one full data shipping. The redo log is continuously replayed on the secondary system immediately after arrival making this step superfluous during takeover. Since the log is continuously replayed, the secondary system can take over immediately, if the primary system fails. With continuous log replay, the log entries are sent from the redo log buffers in memory. When the secondary system is temporarily disconnected, the primary system must not claim and overwrite the log segments that have not been replicated yet. This is achieved by retaining these log segments up to a configurable maximum retention size. When the maximum retention size is reached, the log segments are reclaimed and overwritten with new log to prevent a standstill of the primary system. After such a situation, a full data snapshot needs to be transferred again, when the secondary system is connected again. Because this operation mode does not require delta data shippings, the amount of data that needs to be transferred to the secondary system is reduced.
logreplay_readaccess	This mode is required for replication to an Active/Active (read enabled) secondary system. Using this operation mode while configuring your system replication, read access becomes possible on the secondary system by establishing a direct connection to the secondary system or by providing a SELECT statement from the primary system with a HINT. For more information , see also *Client Support for Active/Active (Read Enabled)* and SAP HANA SQL Reference Guide. This operation mode is similar to the `logreplay` operation mode regarding the continuous log shipping, the redo log replay on the secondary system, as well as the required initial full data shipping and the takeover. As with the logreplay operation mode, the redo log is replicated to the secondary system and continuously replayed to keep the secondary system synchronized.

Limitations

Before you begin preparing a replication strategy for an SAP HANA system, consider the following important aspects regarding the operation modes `logreplay` and `logreplay_readaccess`:

- Registering a secondary with operation mode `logreplay` against a primary running on an SAP HANA revision less than or equal to SPS10 will not work, because the primary does not yet support this feature. Furthermore, for operation mode `logreplay_readaccess` the primary must be running on a revision SAP HANA 2 SPS00 or higher.
- Only the operation mode `delta_datashipping` will work when registering the original primary (failback) after upgrade of the secondary during a near zero downtime upgrade from an SAP HANA revision less than or equal to SPS 10 to SPS 11, because the former primary's version does not yet support `logreplay`.
- If the connection to the secondary is not available, the primary system will keep writing the redo log segments in the online log area to be prepared for the delta log shipping after the connection is reestablished. These log segments are marked as *RetainedFree* until the secondary is in sync again. In this case there is a risk that the log volume may run full. To prevent this:
  - If a secondary is not used anymore, it must be unregistered (sr_unregister).
  - If a takeover to the secondary was done, the former primary should be disabled (sr_disable).

  For more information, see *How to Avoid Log Full Situations* in *LogReplay: Managing the Size of the Log File*.
- In a multitier or multitarget system replication it is not possible to combine the `logreplay` and `delta_datashipping` operation modes.
- For multitarget system replication only the `logreplay` and `logreplay_readaccess` modes are supported.
- The logreplay operation modes do not support history tables.

> **Note**
>
> If you plan to add SAP HANA dynamic tiering to your landscape in the future, please check supported operation modes in *SAP Note 2447994* before you enable SAP HANA system replication.

> **Note**
>
> When selecting an operation mode, keep in mind that the selected operation mode impacts the network throughput requirements of the communication channel used in SAP HANA system replication. For more information about this, see *Network Recommendations*.

For information about how to change the operation mode, see *Changing the Operation Mode*.

Related Information

Data Transferred to the Secondary System [page 775]
Active/Active (Read Enabled) [page 852]
SAP HANA System Replication Command Line Reference [page 811]
LogReplay: Managing the Size of the Log File
Network Recommendations
Changing the Operation Mode [page 797]

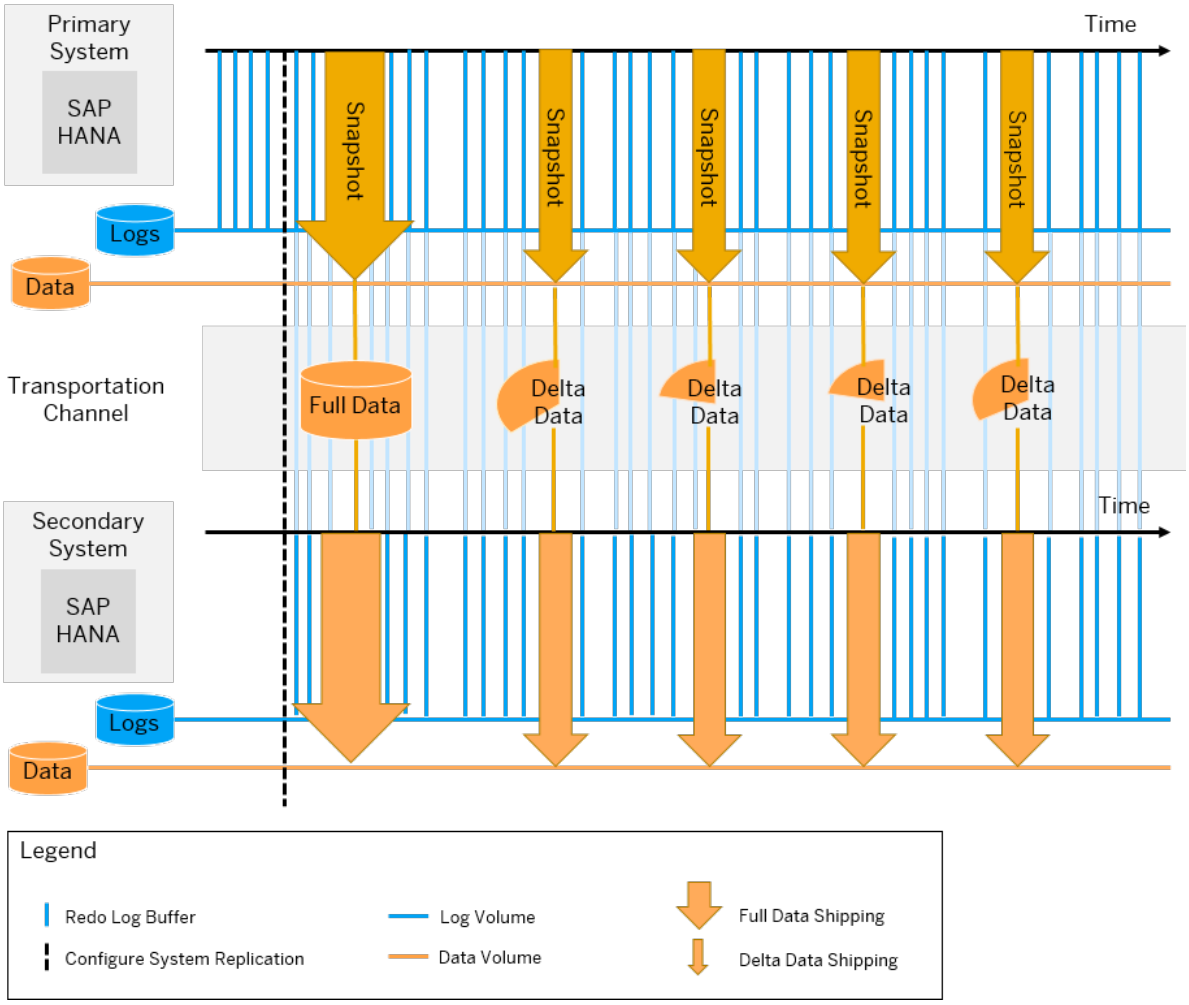
12.1.2.4.1 Data Transferred to the Secondary System

The selected operation mode determines what types of data packages are sent to the secondary system.

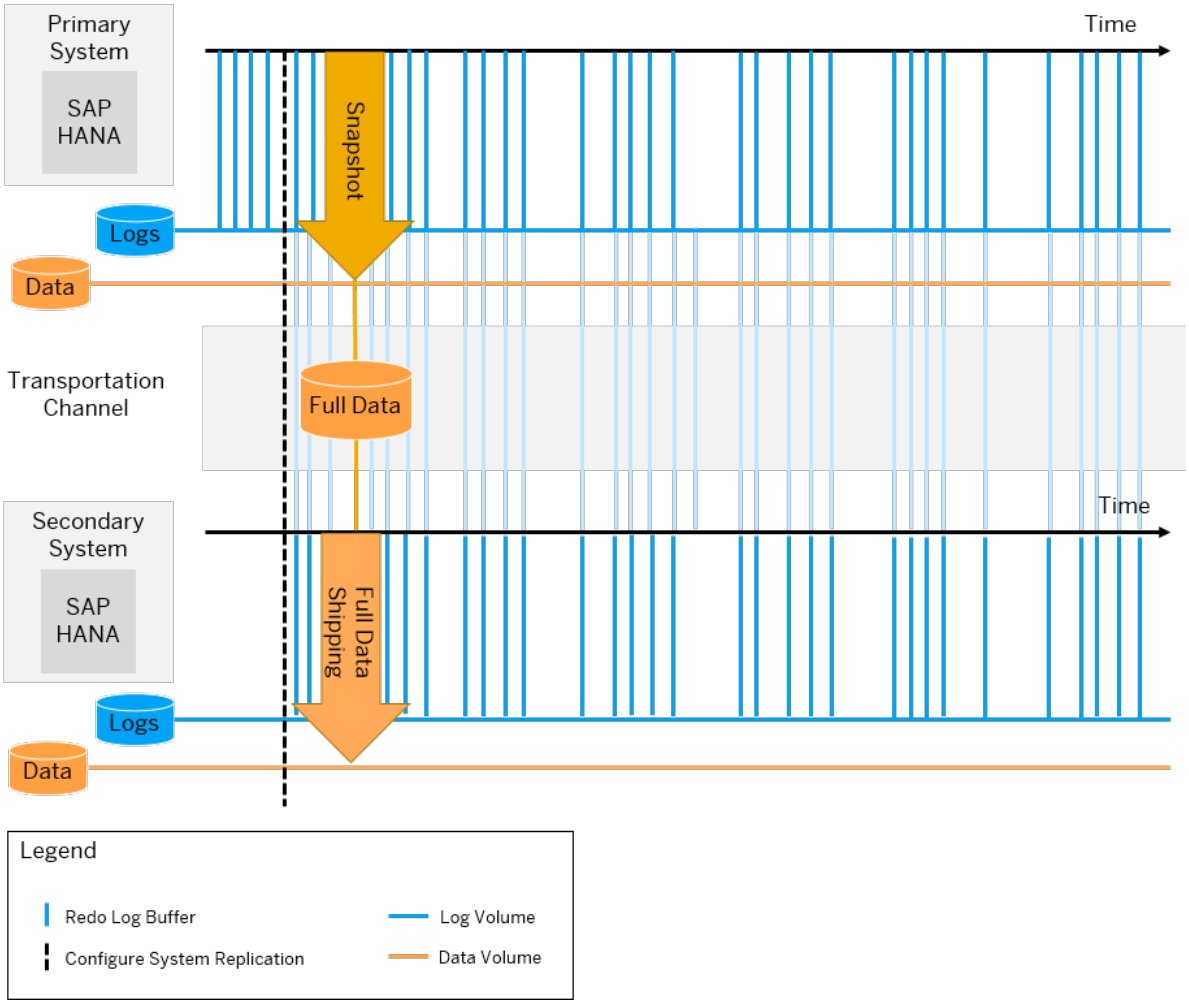
When system replication is configured, the following types of data packages can be sent to the secondary system:

- Initial full data shipping
  When system replication is configured, a full set of data created as an SAP HANA in-place snapshot on the disk of the primary system is initially sent.
- Delta data shipping
  When using the `delta_datashipping` operation mode, the data that has changed since the last full or the last delta data shipping is transported from the data area of the primary system to the data area of the secondary system. The default time is every 10 minutes.
  When using `logreplay` and `logreplay_readaccess`, delta data shippings are not required.
- Continous redo log shipping
  Every committing write transaction on the primary system generates redo log buffers, which are continuously sent to the secondary system.

The following graphic illustrates this traffic on the transportation channel between the primary and the secondary system for the `delta_datashipping` operation mode:



The following graphic illustrates this traffic on the transportation channel between the primary and the secondary system for the `logreplay` and `logreplay_readaccess` operation modes:



> **Note**
>
> With the ini file parameter `datashipping_parallel_channels` (default 4), the full and the delta data shipping are done using parallel network channels. You can change it on the secondary system in the `global.ini` section `[system replication]`. For more information about configuration parameters, see *SAP HANA System Replication Configuration Parameters*.

Related Information

SAP HANA System Replication Configuration Parameters [page 801]
Troubleshoot System Replication
System Replication Network Connection

12.1.2.4.2 Data Volume Encryption with System Replication

Encryption for data volume, redo log, and data and log backups can be applied to all replicated systems.

Enabling Encryption

You can enable (or disable) encryption in the primary system and the setting will be propagated to all secondary systems which are running and replicating. On replicated sites data volume encryption can be set either by parameter replication, or by directly applying manual settings.

In general, if you enable encryption in an operational database only the pages in use in the data volumes are encrypted. Pages in data volumes that are not in use may still contain old content, and are only overwritten and encrypted over time. In addition, only redo log entries that are created after encryption is enabled are encrypted. See also 'Enable Encryption' in the Security section.

From a replication perspective, using logreplay / logreplay_readaccess operation modes, when you start encryption on the primary the complete persistence of all replicating systems is encrypted.

Using the delta_datashipping mode only new sites are encrypted. In this case, if you want to ensure that previously created data pages are also encrypted it is necessary to re-register a site using the `force_full_replica` parameter to enable encryption. Carry out the following steps:

1. Stop the secondary site:

```
sapcontrol –nr <instance_number> -function StopSystem HDB
```

2. Re-register secondary site:

```
hdbnsutil -sr_register --force_full_replica
```

3. Start the secondary system to start replication:

```
sapcontrol –nr <instance_number> -function StartSystem HDB
```

See also SAP Note 2767023 *Data Volume Encryption is Only Propagated to Secondary Site for Newly Written Pages*.

Checking the Encryption Status

On the primary site you can check whether that site is encrypted by running the following query:

```
select * from M_PERSISTENCE_ENCRYPTION_STATUS
```

On the secondary you can run the following hdbcons command to get information about the encryption status:

```
hdbcons "encryption status"
```

To identify a specific tenant, include the process ID of the tenant's indexserver:

```
hdbcons -p <PID> "encryption status"
```

Return values are: active: 0/1 or mayContainUnencryptedData: 0/1.

The same hdbcons command can be run on the replication site (SYSTEMDB logged in as <sid>adm) using the remote execution functionality to check the encryption status on the secondary or tertiary sites:

```
hdbcons -e hdbnameserver "distribute exec <hostname>:<port> encryption status”
```

See also SAP Note 2767023 *How to check the encryption status of a replication site in SAP HANA*.

Related Information

Enable Encryption [page 533]
SAP Note 2754379
SAP Note 2767023

12.1.2.5 Resync Optimization

Whenever the primary and the secondary system are disconnected, SAP HANA system replication is out of sync. To get in sync again, a shipping of the missing data is initiated.

The system tries to avoid a full data shipping and to achieve a resync with a delta data or a log shipping. To get the primary and the secondary system in sync again, their persistencies (that is, the data and log volumes) must be compatible. The system that is to be registered as the secondary system checks if its persistence is compatible with the primary system. If this check succeeds, a delta shipping can be carried out instead of requesting a full data shipping from the primary system.

A maximum of three checks are executed by the secondary system in the following order:

1. Check if the newest savepoint is compatible.
   The to-be secondary system checks if its newest savepoint is compatible. This check most likely succeeds if the secondary system has just been shut down for a short time.
2. Check if the newest replication snapshot is compatible:
   Replication snapshots are written on the system replication primary and secondary system while the replication is up and running. The replication snapshots are created on the secondary system each time a savepoint is written. On the primary system, the replication snapshots are created periodically (time and volume-based) to preserve a state that is known to be shipped to the secondary system. As the snapshot verification takes some time, a replication snapshot that is not yet verified to be shipped may have been created on the primary system.
   This check most likely succeeds after a test takeover on the secondary system because this state has to be available also on the primary system.
3. Check if the active replication snapshot is compatible:
   The active replication snapshot is a special replication snapshot created on the primary system and verified to be shipped to the secondary system. This check most likely succeeds during a failback operation because it's created on the old primary and the snapshot is verified to be shipped.

The first savepoint or snapshot that is compatible with the primary system will be used for delta data shipping.

For the two logreplay operation modes if the persistence is found to be not compatible then a full data shipping will be automatically carried out, this is determined by the configuration parameter `full_replica_on_failed_delta_sync_check` on the secondary. Automating this step avoids the need for intervention by an administrator but if you wish to prevent automatic resynchronization you can set the parameter to FALSE. Note that starting with SPS 06 the default value of this parameter was changed to TRUE so that a full data shipping is automatically carried out.

> **Note**
>
> If system replication is out of sync and you need to register the initial secondary system again, use the command `hdbnsutil –sr_register` (refer to the topic 'SAP HANA System Replication Command Line Reference' for details of the parameters and usage of this command). It is not necessary to unregister the secondary system before registering it again; unregistering the initial secondary would prevent an optimized resync and would trigger a full data shipping (see also SAP Note 1945676 - Correct usage of hdbnsutil -sr_unregister).

Depending on the chosen operation mode, two different techniques are in place to achieve a resync: data retention (for mode delta_datashipping) and log retention (for the logreplay modes). For more information, see *Data Retention* or *Log Retention*

Related Information

Data Retention [page 780]
Log Retention [page 781]
SAP HANA System Replication Command Line Reference [page 811]
SAP Note 1945676

12.1.2.5.1 Data Retention

Data retention is the technique used to resync the disconnected systems when using the `delta_datashipping` operation mode.

With `delta_datashipping` the availability of the last snapshot that was successfully received by the secondary system, determines the type of the data shipping:

- If the last snapshot successfully received by the secondary system is still available on the secondary, the secondary system will request the incremental data to get in sync again.
- If it is no longer available, a full data shipping is necessary to get in sync again. The full data shipping is triggered automatically when the secondary reconnects with the primary and attempts to resynchronize.

To reduce the need for full data shipping after takeover, data snapshots are retained on the primary (or the new primary after takeover) for a given period of time. The `datashipping_snapshot_max_retention_time` parameter with a default of 300 minutes specifies for how long the primary system will keep the snapshot. Takeover takes place automatically but failback (registering the secondary) is a manual action which must be done within the snapshot retention period otherwise a full data shipping will be necessary.

Scenarios where resynchronization is possible

In many cases, when attempting to reconnect the secondary system after a takeover, the snapshot is available and resynchronization is possible via delta data or delta log shipping, as illustrated in the following scenarios:

- The failback scenario
  After a takeover the original primary system that was offline is registered as a new secondary. In this case a valid snapshot should be available on the new secondary from the time when this system was the primary and therefore resynchronization can be performed without the need for full data shipping.
- The re-register scenario (this is a non-standard case which could occur in a test environment)
  After a takeover the original secondary is again (without a long delay) registered as a secondary to the original primary. In this case the snapshots should still be available on the original secondary.
- Reconnection without takeover
  If the connection to an already initialized secondary system is temporarily lost but no takeover takes place then a valid snapshot will still be available on the secondary and replication can continue once the connection is restored.

Actions which may necessitate full data shipping

These takeover scenarios might not apply if any of the following actions have been taken which would cause the snapshot to get lost. In these cases, if the snapshot is not available then full data shipping would be triggered:

- Disabling the primary system before registering it as secondary
- Running the system in the role of primary for more than the snapshot retention time.
- Unregistering a secondary so that it becomes a standalone system.

Related Information

SAP HANA System Replication: Takeover and Failback
SAP HANA System Replication Configuration Parameters [page 801]

12.1.2.5.2 Log Retention

With the `logreplay` and `logreplay_readaccess` operation modes, log segments can be marked as `retained` so that they can sync a secondary system after being disconnected.

With continuous log replay, delta data shipping cannot be used to sync a secondary system anymore. This is because although the primary's and secondary's persistence are logically compatible, they are no longer physically compatible. This means the data that is contained in the persistence is the same, but the layout of the data on pages can be different on the secondary system. Therefore, a secondary system can sync via delta log shipping only. This happens, for example, in the following use cases:

- The secondary system has been disconnected for some time (for example, because of a network problem or temporary shutdown of the secondary system).
- A former primary system has been registered for failback.

The secondary system only uses the log of the online log area of the primary system for re-syncing. The log must be retained for a longer time period than in the `delta_datashipping` mode to be able to sync the

secondary system. If getting in sync again doesn't work with delta log shipping (for example, because the log has been reused), a full data shipping becomes necessary. To avoid this, the concept of log retention has been introduced.

For more information on log retention in different scenarios, see *Log Retention for Secondary Disconnect*, *Log Retention for Failback*, and *Log Retention and Multitarget System Replication*.

The following parameters are significant for log retention:

- Use the `enable_log_retention` parameter to enable or disable log retention.
- Use the `logshipping_max_retention_size` parameter to specify how the system behaves when many log segments of the type `RetainedFree` are created.

For a full description of the parameters, see *SAP HANA System Replication Configuration Parameters*.

Related Information

Log Retention for Secondary Disconnect [page 782]
Log Retention for Failback [page 783]
Estimating the Maximum Retention Time
Log Retention and Multitarget System Replication [page 784]
SAP HANA System Replication Configuration Parameters [page 801]

12.1.2.5.2.1 Log Retention for Secondary Disconnect

When a secondary system configured with the operation mode `logreplay` or `logreplay_readaccess` is disconnected, the primary system will not reuse the log segments in the online log area that are required to sync the secondary system using delta log shipping.

These log segments are marked as `RetainedFree` until the secondary has successfully synced again. If a secondary system is stopped, the log volume will grow on the primary system until the log volume has filled up with log segments. Once the secondary system reconnects and has synced the missing log, these log segments are then set to *Free* and can be reused after that. This behavior is automatically turned on, if a secondary system with the operation mode `logreplay` or `logreplay_readaccess` is registered.

Log segments are retained on the primary as long as the secondary system is registered, but not connected to the primary system.

> ### Note
>
> Therefore, if a secondary system is shut down and not used for a longer period of time, unregister it first, to prevent log volumes from accumulating on the primary system. However, in such a case a full data shipping will be necessary when the system reconnects.

> ### Note
>
> Log full means that no more log segments can be created in the log volume, because the log segment directory is full. Currently the number of log segments is roughly 10000. Disk full means that the disk is full,

which is not necessarily the case in the log full situation. Log retention usually deals with disk full situations, because with 10000 log segments having each a size of 1 GB you can create 10 TB of log segments.

To understand for how long the SAP HANA system replication landscape will survive before running into a log full situation, see *Estimating the Maximum Retention Time*.

The `logshipping_max_retention_size` parameter determines if a full log volume can be prevented at the price of a possibly necessary full data shipping when the system reconnects. The value of this parameter (default is 1 TB per log volume) should not exceed the size of the file system reserved for all log volumes.

Related Information

Estimating the Maximum Retention Time

12.1.2.5.2.2 Log Retention for Failback

On the secondary system, log retention is required to do a failback with optimized data transport.

The primary system periodically creates persistence snapshots during replication and provides the log position information to the secondary system. After takeover, when the old primary is started again as a secondary, the most recent snapshot is opened on the old primary system and the missing log is requested from the new primary system.

Log retention can occur in two situations:

- While replication is active
  During replication, the secondary system keeps all log starting from the last snapshot position provided by the primary system. The old log is automatically released after a new snapshot has been created on the primary system. This behavior is turned on by default and it ensures that during replication only a few `RetainedFree` segments are kept online. They are needed to fill the gap between the primary snapshot and the current potential takeover log position.
- After a takeover
  After takeover, the new primary has to keep the log until a new secondary system is registered and has synced the missing log. Because syncing can take some time, this behavior has to be turned on explicitly on the new primary system as follows:
  `global.ini/[system_replication]/enable_log_retention = on`

  → Recommendation

  If you have a setup in which there will be frequent failbacks between two systems, we recommend that you set the following parameter on both system to simplify the configuration: `global.ini/[system_replication]/enable_log_retention = on`

  ⓘ Note

  If the old primary system will not be reused as the new secondary system (failback), it should be disabled after the takeover using `hdbnsutil -sr_disable` to prevent log volumes from

accumulating on the new primary system. You can disable it with SAP HANA cockpit, SAP HANA studio, or using the command line.

12.1.2.5.2.3 Log Retention and Multitarget System Replication

When the primary system replicates data changes to more than one secondary system, you should use force log retention and log retention propagation to reach an optimized re-sync and avoid a full data shipping after takeover or other disconnect situations.

Force log retention

Force log retention is used on a system to retain log until it's actively disabled. To use force log retention, enter the value **`force_on_takeover`** for the `enable_log_retention` configuration parameter.

If `enable_log_retention = force_on_takeover` is configured, the log will be retained during replication for all direct secondaries until a takeover is executed. During takeover, the parameter is set to **`force`**. This means the log will be retained independently of any secondary system.

> Example
>
> A typical scenario is described in the following steps:
>
> 1. Configure all systems with `[system_replication]/enable_log_retention = force_on_takeover`
> 2. During takeover on a secondary system, if **`force_on_takeover`** is set, the value is changed to `enable_log_retention = force`. This means that starting from the takeover, the log is retained until it's explicitly disabled.
> 3. Re-register all required systems until the landscape is fully functional again.
> 4. Reset `[system_replication]/enable_log_retention = force_on_takeover` on the system on which takeover has been executed before re-establishing the original configuration.

The configuration must be done manually (for example, by the administrator or using setup scripts) because the SAP HANA system doesn't know when the system landscape has been completely reconfigured.

Log retention propagation

Log retention propagation is used to retain the log based on the smallest savepoint log position in the whole system replication landscape. Log retention propagation should be enabled if you want to re-order your systems in a complex system replication setup.

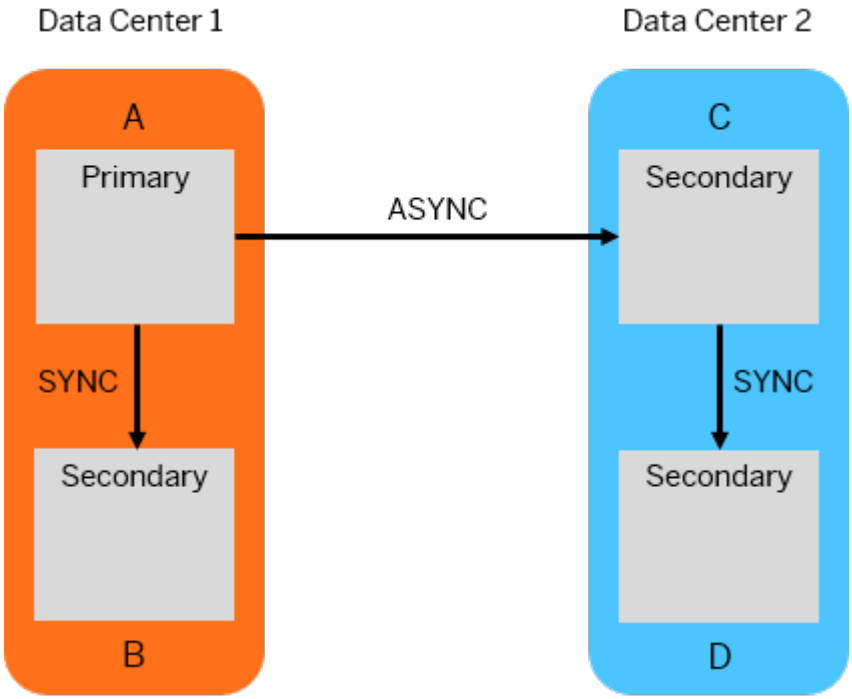
This can be done by setting the following parameter in global.ini: `[system_replication]/propagate_log_retention = on`. If you want to propagate log retention in a system replication landscape between all systems, this parameter should be set on all systems in the landscape.

When you set this parameter on a system, it behaves as follows:

- The system sends to its source system the minimum log position of its own savepoint and the retained log position it gets from all direct secondaries as retained position.
- The system sends the minimum log position of its own savepoint and the retained log position it gets from all direct secondaries to the secondaries as retained log position.
- The system uses the minmum log position it gets from all direct secondaries and its source system (if not primary) as own retained log position.

Example

To explain these concepts we are using the setup described in *Multitarget System Replication*. In this setup, primary system A replicates data changes to secondary system B located in the same data center. Primary system A also replicates data changes to the secondary system C located in another data center. Secondary system C is a source system for a further secondary system D located in the same data center with system C. For a quick overview, use the graphic below:



If there is a takeover on secondary system B, you must register system C to B and system A to B to recreate the original configuration. To avoid a full data shipping for both systems, system B must retain all the log until systems A and C have synced again. This can't be accomplished by setting `global.ini/[system_replication]/enable_log_retention = on` because system B doesn't know how many systems must be re-attached until the landscape is back in its functional state.

Force log retention should be used on system B until systems A and C are registered again and synced.

If you want to re-order your systems, enable log retention propagation. Log retention without propagation only affects the direct neighbors. For example, if system D is stopped in this setup, system C retains log for D, but not for A and B. If system D is re-attached to systems B or A and propagation is not turned on, log could be missing because systems A and B do not retain their log with respect to D.

Related Information

SAP HANA Multitarget System Replication [page 846]
Disaster Recovery Scenarios for Multitarget System Replication [page 848]

12.1.2.5.3 Data and Log Compression

Data and log compression can be configured to reduce the amount of traffic between systems, especially over long distances (for example, when using the ASYNC replication mode).

Data and log compression can be used for the initial full data shipping, the sub sequential delta data shipping, as well as for the continuous log shipping.The following types of compression for log and data shipping are supported:

- Log
  - Log buffer tail compression
  - Log buffer content compression
- Data
  - Data page compression

Log Buffer Tail Compression

All log buffers are aligned to 4kb boundaries by a filler entry. With log buffer tail compression the filler entry is cut off from the buffer before sending it over the network and added again when the buffer has reached the secondary system. So, only the net buffer size is transferred to the secondary system.

The size of the filler entry is less than 4kb, this is the maximum size reduction per sent log buffer. If the size of the log buffers is large, the compression ratio is limited. Log buffer tail compression is turned on by default.

Log Buffer and Page Content Compression

Log buffers and data pages shipped to the secondary system can be compressed using a lossless compression algorithm (lz4). By default content compression is turned off. You can turn it on by setting the following configuration parameters on the secondary system in the system_replication section of the global.ini file:

- `enable_log_compression = true`
- `enable_data_compression = true`

Log buffer content compression works also in combination with log buffer tail compression. So, only the content part of the log buffer is compressed, without considering the filler entry.

Related Information

External link to LZ4

12.1.2.6 SAP HANA System Replication Setup for XS Advanced Runtime

In a system replication setup, all the data – including XS advanced runtime system data and application data – is replicated to a secondary system.

XS advanced services and applications run only on the currently active system. On the secondary system, XS advanced services are in an idle state until the takeover takes place. This means that while a system is in "secondary" status, applications are not started and login to XS Advanced is not possible. After the takeover, all XS advanced services and applications are started on the `xs_worker` hosts on the former secondary system.

XS Advanced integrates into the system replication mechanism provided by SAP HANA. However, routing of requests to XS advanced services and applications needs special care. In order to setup XS Advanced in combination with system replication, choose one of the following options:

- Use System Replication with an External Domain
- Use System Replication without an External Domain

Use System Replication with an External Domain

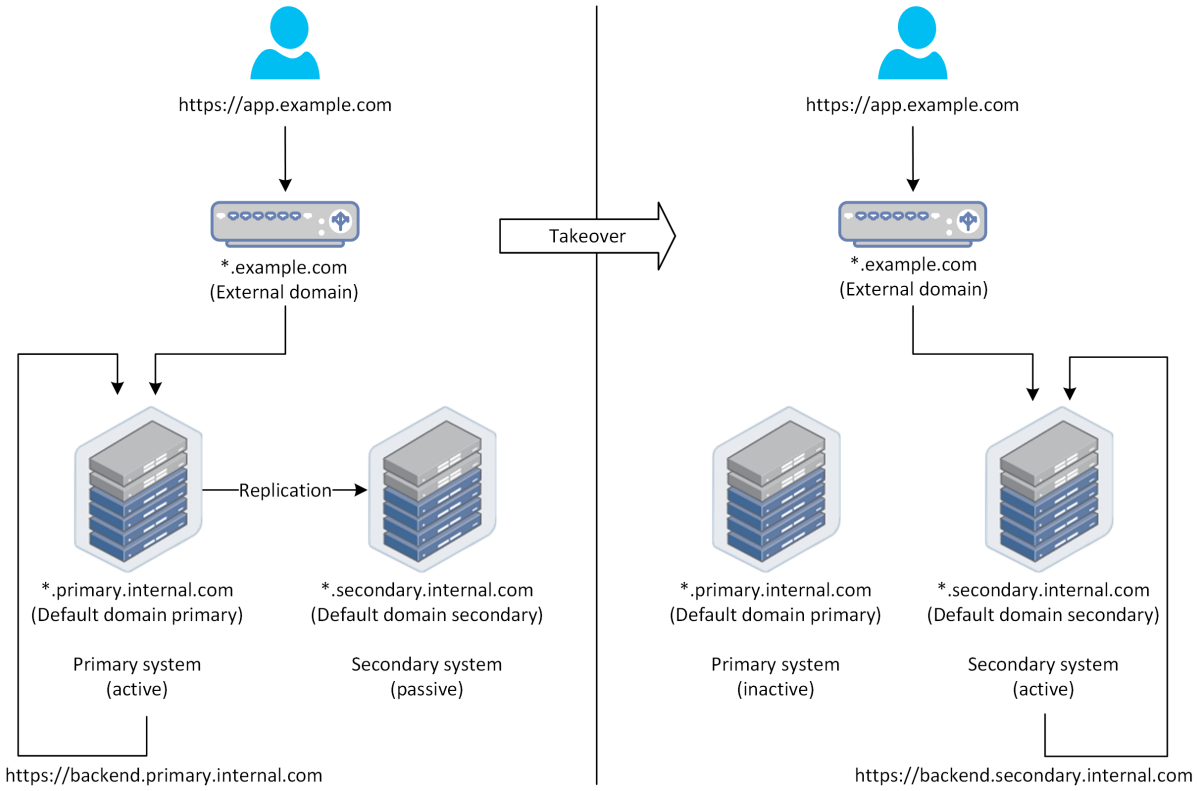
It is recommended to use SAP HANA System replication with an external domain in XS Advanced in order to correctly route incoming requests. With this setup each system can have a distinct default domain, which allows the systems to communicate internally independently. To use this option, you need to perform the following high-level steps:

1. Setup the primary and secondary systems separately, use distinct default domain settings when configuring XS Advanced.
2. On the primary system, use the command `xs set-certificate` to apply a default domain certificate that includes the wildcard default domain names of all secondary systems (for example, `*.secondary.internal.com`) in the Subject Alternative Name (SAN) section of the certificate.
3. For the primary system, follow the steps described in *Using A Reverse Proxy with an External Domain* in *Related Information* below to configure a failover router or virtual hostname.
4. Activate system replication, which is described in *General Prerequisites for Configuring SAP HANA System Replication* in *Related Information* below.

The external route configuration is automatically replicated to the secondary system, so no manual configuration is necessary on the secondary system. In this scenario, the default domain setting is not replicated to the secondary system and stays stable on both primary and secondary systems even after failover.

Example Setup

In the following example, an external domain `example.com` is configured to route incoming external requests. All requests for `*.example.com` are routed via a reverse proxy, which forwards the requests to the default domain of the currently active system. The default domain configuration of the primary and secondary system is `primary.internal.com` and `secondary.internal.com`, respectively.



As each system uses a separate and stable default domain, the systems can communicate independently of each other even if the external route is not available.

Performing a Takeover

To perform a takeover, you need to perform the following steps:

1. Use SAP HANA database tools to perform a takeover.

   > **Note**
   >
   > For more information, see *Performing a Takeover* in *Related Information* below.

2. To ensure that XS advanced applications are available on the external route again, configure the reverse proxy to forward requests on the external domain (`*.example.com`) to the former secondary system (for example, `secondary.internal.com`). If you use a virtual hostname instead of a reverse proxy, point the virtual hostname to the former secondary system.

Disabling System Replication

After disabling system replication, primary and secondary systems still share the same external domain settings. To make each system available via individual external domains, the external domain configuration must be unique per system. To reconfigure the external domain setting for a system, follow these steps:

1. Log in with the `xs` cli as XS advanced administrator using the default domain of the affected system, for example:

   {·} Output Code

   ```
   xs login -a https://api.primary.internal.com -u XSA_ADMIN
   ```

2. Delete the existing external domain, for example, with the following command:

   {·} Output Code

   ```
   xs delete-domain example.com
   ```

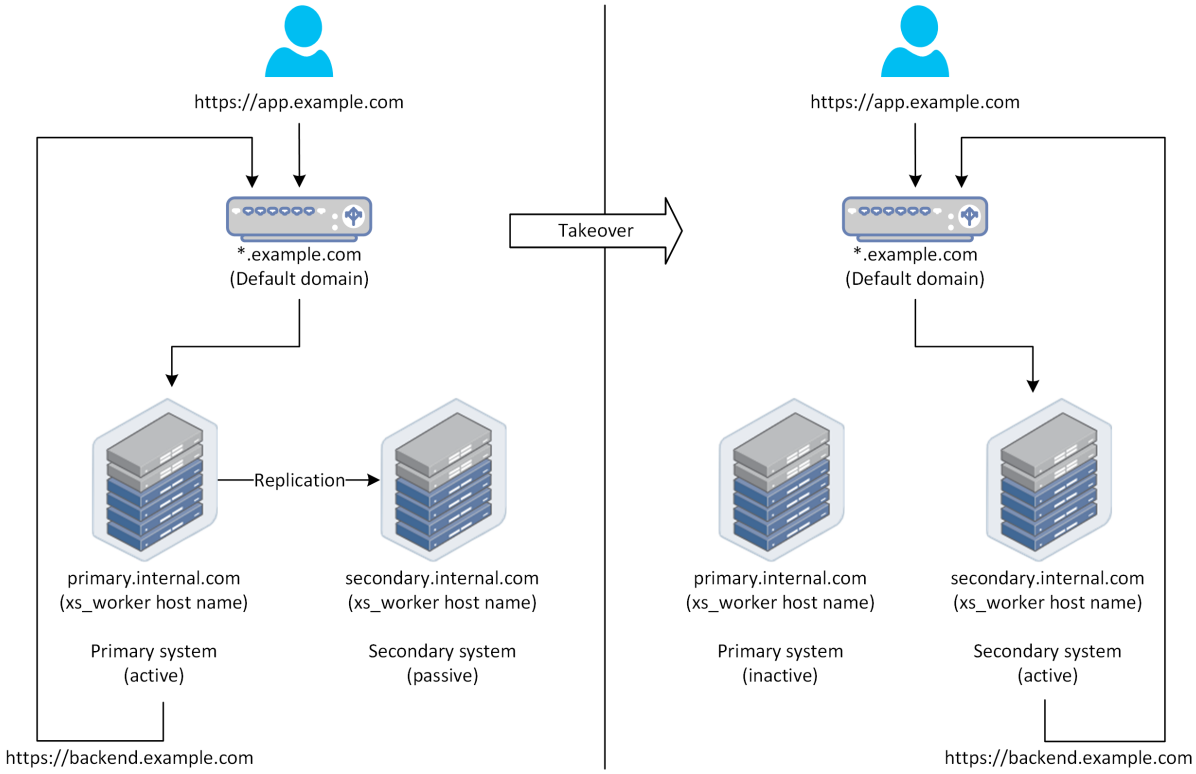
   → Tip

   You can continue to use the system without an external domain. Alternatively, you can configure a new domain as described in *Using A Reverse Proxy with an External Domain* in *Related Information* below.

Use System Replication without an External Domain

If you set up system replication without an external domain, XS advanced applications must be reached by the default domain, independently of the currently active system. To enable this, you need to use a common

default domain that is shared between primary and secondary systems and points to a failover router, as shown in the following diagram:



In the scenario immediately above, the primary and secondary systems share a common default domain setting `example.com`, which points to the router and is automatically replicated to the secondary system. The router forwards incoming requests to the physical host name of the currently active `xs_worker` host.

> → Tip
>
> The default domain is also used by XS advanced components to communicate internally. This is why the secondary system is only able to function after a takeover once the failover router forwards requests to the secondary host. For more information, see SAP Note 2300936.

Additional Information

Bear in mind the following important points:

- If you use applications that make use of the file-system-service (for example, the SAP WebIDE for SAP HANA), see *Configuring the XS Advanced File-System Service* in *Related Information* below, when setting up system replication.
- If you rotate technical passwords on the primary system, for example by means of the command `XSA renew-passwords-for-technical-users`, make sure to copy the XS SSFS (Secure Store in the File System) to all secondary systems as described in *General Prerequisites for Configuring SAP HANA System Replication* in *Related Information* below.

- On the primary system, set up certificates to include all `xs_worker` host names of all secondary systems. For more information, see *Creating an External Domain* in *Maintaining Domains in XS Advanced* in *Related Information* below.

Related Information

Host Auto-Failover Setup with XS Advanced Run Time [page 902]
SAP Note 2300936
Configuring the XS Advanced File-System Service [page 1501]
General Prerequisites for Configuring SAP HANA System Replication [page 765]
Maintaining Domains and Certificates in XS Advanced [page 1324]
Using A Reverse Proxy with an External Domain in XS Advanced (SAP HANA Server Installation & Update Guide)
Performing a Takeover [page 815]

12.1.2.7 Overview of Steps

This topic provides an overview of the steps involved in setting up system replication between two systems, taking over to a secondary system, failing back to a primary system, and disabling system replication.

Procedure

1. Set up system replication on primary and secondary systems:
   a. Start the primary system.
   b. Create an initial data backup or storage snapshot on the primary system. The system database and all tenant databases must be backed up.
   c. Enable system replication on the primary system (sr_enable).
   d. Prepare the secondary system for authentication by copying the system PKI SSFS .key and the .dat file from the primary system to the secondary system. For more information see *General Prerequisites for Configuring System Replication* and see SAP Note 2369981 *Required configuration steps for authentication with HANA System Replication*.
   e. Register the secondary system with the primary system (sr_register).
   f. Start the secondary system.
2. During takeover, the secondary system takes over from the primary system:
   a. Stop primary system in data center A.
   b. Secondary system in data center B takes over from primary in data center A (sr_takeover).
   c. When the primary system is available again, register it with the secondary system (sr_register).

      The roles are switched, the original primary is registered as a secondary system. The original secondary is the production system.

d. Start the system in data center A.

To avoid any risk of having multiple primary systems active at the same time you can also start a takeover with handshake. Using this option it is not necessary to stop the primary first and it also ensures that all the sent redo logs are written to disk on the secondary system before the takeover. For more information see *Takeover with Handshake*.

3. Restore the role of the original primary system:
   a. Stop the system in data center B.
   b. Send a takeover command from the system in data center A (sr_takeover).
   c. Register the system in data center B as secondary again (sr_register).
   d. Start the system in data center B.
4. Disable system replication:
   a. Unregister the secondary system.
   b. Disable system replication on the primary system.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 765]
Configuring SAP HANA System Replication [page 792]
Performing a Takeover [page 815]
Takeover with Handshake [page 824]
Performing a Failback [page 832]
Disabling SAP HANA System Replication [page 833]
Rename an SAP HANA System Host [page 708]
Configuring the Network for Multiple Hosts [page 1115]
Implementing a HA/DR Provider [page 905]
Enable Encryption [page 533]
Data and Log Volume Encryption
Encryption Key Management
SAP Note 2211663
SAP Note 2369981

12.1.2.8 Configuring SAP HANA System Replication

You can configure system replication using SAP HANA cockpit, SAP HANA studio, or hdbnsutil.

You can configure system replication using the following tools:

- SAP HANA cockpit
  For more information, see the SAP HANA cockpit guide*Configure SAP HANA System Replication with the SAP HANA Cockpit*.
- SAP HANA studio
  For more information, see the SAP HANA studio guide *Configuring SAP HANA System Replication*.

- hdbnsutil
  For more information, see *Configure System Replication with hdbnsutil*.

Related Information

Configure SAP HANA System Replication with the SAP HANA Cockpit
Configure SAP HANA System Replication with the SAP HANA Studio
Configure SAP HANA System Replication with hdbnsutil [page 793]
Full Sync Option for SAP HANA System Replication [page 800]

12.1.2.8.1 Configure SAP HANA System Replication with hdbnsutil

You can configure SAP HANA system replication with the hdbnsutil command line tool.

Prerequisites

You have considered all the general prerequisities needed to configure system replication. For more information, see *General Prerequisites for Setting Up SAP HANA System Replication*.

> **Note**
>
> For SAP HANA tenant database systems all databases must be backed up using hdbnsutil via the database name option:
>
> - for the system database `-d SystemDB`
> - for the tenant databases `-d <tenantDBName>`

Procedure

1. Enable system replication on the primary system as follows:
   a. Ensure that the configuration parameter `log_mode` is set to `normal` in the `persistence` section of the `global.ini` file.

      Log mode `normal` means that log segments must be backed up. Log mode `overwrite` means that log segments are freed by the savepoint (therefore only useful for test installations without backup and recovery).
   b. Do an initial data backup or create a storage snapshot. In multiple-container systems, the system database and all tenant databases must be backed up. For more information, see *Creating Backups*.

c. As `<sid>adm` on the command line enable the primary for system replication and give it a logical name with the following command. The primary system must be online at this time:

```
cd /usr/sap/<sid>/HDB<instancenr>/exe
```

```
./hdbnsutil -sr_enable --name=<siteName>
```

Option Name	Value	Description
--name	<primary_alias>	Alias used to represent your primary system and assign it as the primary system for system replication

To check if the system has been successfully enabled for system replication with hdbnsutil run:

```
cd /usr/sap/<sid>/HDB<instancenr>/exe
```

```
./hdbnsutil -sr_state
```

2. Stop the secondary system:

```
sapcontrol -nr <instance_number> –function StopSystem HDB
```

If you are running SAP HANA 2.0, you will need to copy the system PKI SSFS key and data file from the primary system to the secondary before registering the secondary system. The corresponding files can be found on the primary system in the following locations:

- `/usr/sap/<SID>/SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT`
- `/usr/sap/<SID>/SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY`

If you are running XS advanced, you will also need to copy both the SSFS key and data files for XS advanced from the primary system to the secondary system. The corresponding files can be found on the primary system in the following locations:

- `/usr/sap/<SID>/SYS/global/xsa/security/ssfs/data/SSFS_<SID>.DAT`
- `/usr/sap/<SID>/SYS/global/xsa/security/ssfs/key/SSFS_<SID>.KEY`

3. Register the secondary system as follows:
   a. Enable system replication on the secondary system as user `<sid>`adm with the following command:

```
hdbnsutil -sr_register --name=<secondarySiteName>
 --remoteHost=<primary_host> --remoteInstance=<primary_systemnr>
 --replicationMode=[sync|syncmem|async]--operationMode=[delta_datashipping|
logreplay|logreplay_readaccess]
```

hdbnsutil -sr_register Call Options

Option Name	Value	Description
--name	<secondarySiteName>	Alias used to represent the secondary system
--remoteHost	<primary_host>	Name of the primary host that the secondary registers with

Option Name	Value	Description
--remoteInstance	<primary_instancenr>	Instance number of primary
--replicationMode	[sync|syncmem|async]	Log replication modes
--operationMode	[delta_datashipping|logreplay|logreplay_readaccess]	Log operation mode
--online	N/A	If the system is running you can use this parameter to automatically perform a system restart. Not relevant if the system is shut down.
--force_full_replica	N/A	Use this parameter to initiate a full data shipping. Otherwise a delta data shipping is attempted
--withAllSecondaries	N/A	Use this option to re-attach existing secondaries to a new source system within an existing system replication setup

To check if the system has been successfully enabled for system replication with hdbnsutil run:

```
cd /usr/sap/<sid>/HDB<instancenr>/exe
```

```
./hdbnsutil -sr_state
```

b. Start the secondary system to reinitialize it with the following command:

As <sid>adm:

```
/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> –function
StartSystem HDB
```

Once the secondary system is started, the replication process will start automatically.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 765]
Rename an SAP HANA System Host [page 708]
Host Name Resolution for System Replication [page 879]
Creating Backups

12.1.2.8.1.1 Example: Configure SAP HANA System Replication

This example shows you how to configure system replication with a single host system.

Context

To configure system replication with two hosts, you may have to change the host names.

In this example a single host system is used. In multi-host systems all hosts have to be renamed.

> **Note**
>
> To rename hosts in a production system replication landscape, system replication must be first deactivated. This means you have to first unregister and disable the secondary system before renaming any hosts. Once you have renamed the hosts then you can enable recovery mode again and register the secondary system with the primary system to re-activate system replication.

Procedure

1. Enable system replication on the primary system, with the hostname ej11.

   ```
   cd /usr/sap/<sid>/HDB<instancenr>/exe
   ```

   ```
   ./hdbnsutil -sr_enable --name=dcsite1
   ```

2. Stop the secondary system. The primary system can stay online.

   As `<sid>`adm

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> –function StopSystem
   HDB
   ```

3. Register the secondary system with the following command:

   ```
   cd /usr/sap/<sid>/HDB<instancenr>/exe
   ```

   ```
   ./hdbnsutil -sr_register
    --name=dcsite2
    --remoteHost=ej11
    --remoteInstance=50
    --replicationMode=sync
    --operationMode=logreplay
   ```

   Also see *SAP Note 611361 Hostnames of SAP servers*

4. Start the secondary system. This initiates the initial data transfer.

As <sid>adm

```
/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> –function StartSystem
HDB
```

Related Information

Rename an SAP HANA System Host [page 708]
SAP Note 611361

12.1.2.8.2 Changing the Replication Mode

The replication mode can be changed without going through a full data shipping from the primary system to the secondary system afterwards.

To change the replication mode, use the following command on the online or offline secondary system:

```
hdbnsutil -sr_changemode --mode=sync|syncmem|async
```

In the M_SERVICE_REPLICATION view you can check whether the replication mode was changed correctly. The following command provides this information too:

```
hdbnsutil -sr_state --sapcontrol=1
```

Related Information

M_SERVICE_REPLICATION System View

12.1.2.8.3 Changing the Operation Mode

The operation mode can only be changed by stopping and re-registering the secondary system with the desired operation mode.

You can change operation modes using the `hdbnsutil -sr_register` command and explicitly setting the new operation mode with the `-operationMode` option:

```
hdbnsutil -sr_register
--remoteHost=<primary hostname>
--remoteInstance=<instance number>
 --replicationMode=[sync|syncmem|async]
--operationMode=[delta_datashipping|logreplay|logreplay_readaccess]
--name=<siteName>
```

To start the replication with the new operation mode, start the secondary system:

```
sapcontrol -nr <instance_number> -function StartSystem HDB
```

> **Note**
>
> It is not necessary to unregister the secondary while changing the operation mode. The `hdbnsutil -sr_register` command overwrites the previous register configuration. To understand the scenarios in which you should unregister a secondary system, see *SAP Note 1945676*.

When changing the operation mode from `delta_datashipping` to `logreplay` or `logreplay_readaccess`, no full data shipping is necessary. Full data shipping is necessary, however, when switching from `logreplay` or `logreplay_readaccess` back to `delta_datashipping`.

Related Information

SAP Note 1945676

12.1.2.8.4 Initializing the Secondary

Whenever the secondary is registered with the primary system, the goal is to get the persistence (that is, the data and log volumes) on the secondary system into a consistent state to the primary system.

After initially configuring system replication, a full data shipping takes place. This happens automatically, but it can also be done manually. For more information, see *Initialize the Secondary with Storage Copy from Primary*.

When initializing the secondary the following two situations can occur:

- The secondary system is completely unrelated to the primary system
  If the secondary system is unrelated to the primary system, a full data shipping is done. An in-place snapshot created on the disk of the primary system is initially sent to the secondary system.
  This initial full data shipping can be prevented by manual intervention and the secondary system can be initialized with a binary storage copy of the primary system's persistence. For more information, see *Initialize the Secondary with Storage Copy from Primary*.
- The secondary system is related to the primary system for one of the following reasons:
  - It was already registered before as a secondary system to this primary and probably shut down for a time.
  - It is a former primary system, which will become a secondary system though failback switching the replicating direction.

  If the persistence (that is data and log volumes) of the secondary system is related to the primary system (it actually contains the persistence of the primary at a former time), the newly registered system can be synced with a delta data or log shipping avoiding a full data shipping.

After a new registration of the secondary system, a delta data or log shipping is always attempted. For more information, see *Resync Optimization*.

Related Information

Initialize the Secondary with Storage Copy from Primary [page 799]
Resync Optimization [page 779]

12.1.2.8.4.1 Initialize the Secondary with Storage Copy from Primary

The secondary system can be initialized using a binary storage copy from the primary system.

Context

For this procedure copy only the data, not the log.

Procedure

1. Create a consistent binary storage copy from the primary system for the persistence of all services. You can use the snapshot technology to create an IO consistent persistence copy. Create a full copy of the persistence using the IO consistent storage snapshots.

   If you can't use the method above, create a consistent OS copy of persistence while the primary system is stopped.
2. Shut down the secondary system.
3. Transfer or mount the full copy on the secondary system.
4. Replace the persistence of the secondary system with the storage copy from the primary system.
5. Register the secondary system without [--force_full_replica].
6. Start the secondary system.

Results

When the secondary system is started after the new registration, the initialization optimizations are carried out. The system checks if the persistence of the secondary system is compatible with the persistence of the primary system. The secondary system checks if its persistence is compatible with the persistence of the primary system. If this check succeeds, the secondary system requests only a delta data shipping.

12.1.2.8.5 Full Sync Option for SAP HANA System Replication

To reach a true Recovery Point Objective value of zero for synchronous system replication, the full sync option can be enabled for SYNC replication mode.

With the activated full sync option, the transaction processing on the primary system blocks when the secondary is currently not connected and newly created log buffers cannot be shipped to the secondary system. This behavior ensures that no transaction can be committed on the primary without shipping the log buffers to the secondary system.

The full sync option can be switched on and off using the command:

```
hdbnsutil -sr_fullsync --enable|--disable
```

This changes the setting of the parameter `enable_full_sync` in the `[system_replication]` section of the global.ini file accordingly.

> → Recommendation
>
> When configuring the full sync option, proceed as follows:
>
> 1. Configure your system replication with the SYNC replication mode. This replication mode is the prerequisite for enabling the full sync option.
> 2. Check that the system replication status is active and in sync for all services.
> 3. Enable the full sync option with `hdbnsutil -sr_fullsync --enable`

In the M_SERVICE_REPLICATION system view, the setting of the full sync option can be viewed in the column *FULL_SYNC*. It can have the following values:

- DISABLED: Full sync is not configured at all.
  The parameter `enable_full_sync = false` in the system_replication section of the global.ini file.
- ENABLED: Full sync is configured, but it is not yet active.
  In this state transactions do not block. To become active, the secondary has to connect and the replication status has to be ACTIVE.
- ACTIVE: Full sync mode is configured and active.
  In this state if the network connection to a connected secondary is closed, the transactions on the primary system will block.

If full sync is enabled when an active secondary is currently connected, the FULL_SYNC column will be immediately set to ACTIVE.

If the secondary is stopped, disable the full sync option. Otherwise the primary blocks and it is not possible to stop it.

> ⓘ Note
>
> Use the `hdbnsutil` command to resolve a blocking situation of the primary system caused by the enabled full sync option. This is important because a configuration changing command could also block in this state. You must do this also when you want to shut down the currently blocking primary system. Otherwise it is not possible to stop it.

The cluster manager that is used to operate SAP HANA system replication landscapes could provide a timeout after which the blocking situation is resolved automatically using the `hdbnsutil` command and deactivating

the full sync option. However, after the reason for the blocking situation disappears, you must activate the full sync option again (manually or automatically with the help of the cluster manager tool).

In a multitarget system replication setup, configure the full sync option on the primary system. Enter the site name used for the secondary system when registering it:

```
global.ini
[system_replication]
enable_full_sync[<secondary_site_name>] = true
```

> **Note**
>
> In a multitarget system replication setup, you can use `hdbnsutil -sr_fullsync` to turn off the full sync option.

Related Information

SAP HANA System Replication Command Line Reference [page 811]

12.1.2.8.6 SAP HANA System Replication Configuration Parameters

Several configuration parameters are available for configuring SAP HANA system replication between the primary and secondary system.

The system replication parameters are defined in the `[system_replication]` section of the `global.ini` file and have the default values shown below. The *System* column defines whether the parameter can be set on the primary, the secondary, or both.

In addition to the parameters described here, a number of parameters for managing the automatic generation of runtime dumps in specific system replication scenarios are described in detail in an SAP Note 2400007 FAQ: SAP HANA Runtime Dumps.

Parameter	`datashipping_min_time_interval`
Type	Integer
Unit	seconds
Default	600 (10 min)
System	Secondary
Description	Minimum time interval between two data shipping requests from the secondary system. If `datashipping_logsize_threshold` (see next parameter) is reached first, the data shipping request will be sent before the time interval is elapsed when the log size threshold is reached.

Parameter	datashipping_logsize_threshold
Type	Integer
Unit	bytes
Default	5*1024*1024*1024 (5 GB)
System	Secondary
Description	Minimum amount of log shipped between two data shipping requests from the secondary system. If the time defined by `datashipping_min_time_interval` (see previous parameter) has passed before reaching this threshold, the data shipping request will be sent before this threshold is reached when the time interval has elapsed.

Parameter	preload_column_tables
Type	Boolean
Default	true
System	Primary and Secondary
Description	When this parameter is set, preloading of column table main parts is activated for the secondary according to the information in the loaded table information from the primary. This parameter has no effect in operation mode delta_datashipping.

Parameter	datashipping_snapshot_max_retention_time
Type	Integer
Unit	minutes
Default	300
System	Primary

Parameter	`datashipping_snapshot_max_retention_time`
Description	Maximum retention time (in minutes) of the last snapshot that has been completely shipped to the secondary system. Shipped snapshots older than `datashipping_snapshot_max_retention_time` will be dropped automatically. Snapshots currently used in data shipping are not affected and are not dropped, if data shipping takes longer than `datashipping_snapshot_max_retention_time`. They can be dropped if data shipping has been finished. If the parameter is set to 0, snapshots are immediately dropped after data replication finishes. When roles are switched between the primary and secondary systems preparing a failback later on, the secondary can be initialized with a delta replica between this snapshot and the current persistent state on the new primary after takeover. In order to do this: • A snapshot has to exist on the new secondary when it starts for the first time as secondary. • The snapshot has to be compatible with the persistence of the new primary. It is verified, if the snapshot has been the source of the primary system before takeover. It cannot be used, if the secondary is registered with an incompatible primary system. If both conditions are true, the secondary can be initialized with a delta replica.

Parameter	`logshipping_timeout`
Type	Integer
Unit	seconds
Default	30
System	Primary
Description	Number of seconds the primary waits for the acknowledgment after sending a log buffer to the secondary system. If the primary does not receive the acknowledgment for a sent log buffer within the time defined by `logshipping_timeout`, it closes the connection to the secondary system in order to continue data processing. This is done to prevent the primary system from blocking transaction processing if there is a hang situation on the connection to the secondary system. After the timeout period for a send operation has elapsed, transactions are written only on the primary system until the secondary has reconnected. The `logshipping_timeout` does not define a blocking period for logshipping on the primary system in general. It is used to close hanging connections on the primary system that are not getting automatically closed. If the redo log cannot be sent to the secondary system within this time, the connection is temporarily closed and the primary writes the redo log locally. This can happen any time, also when the primary is currently not waiting for acknowledgments from the secondary system. Use the ful sync option, if the primary system should block whenever the connection to the secondary system is lost. In this case the primary system will stop.

Parameter	logshipping_async_buffer_size
Type	Integer
Unit	bytes
Default	67108864 (64MB)
System	Primary
Description	In asynchronous replication mode, the log writer copies the log buffers into an intermediate memory buffer first and continues processing. A separate thread reads log buffers from this memory buffer and sends them to the secondary system asynchronously over the network. This parameter determines how much log can be intermediately buffered. This buffer is especially useful in peak times when log is generated faster than it can be sent to the secondary system. If the buffer is large, it can handle peaks for a longer period of time. The behavior of buffer full situations can be controlled by the parameter logshipping_async_wait_on_buffer_full. The parameter can be changed online, but will become active the next time the secondary system reconnects.

Parameter	logshipping_async_wait_on_buffer_full
Type	Boolean
Default	true
System	Primary
Description	This parameter controls the behavior of the primary system in asynchronous log shipping mode when the log shipping buffer is full. If set to true, the primary system potentially waits until there is enough space in the log shipping buffer, so that the log buffer can be copied into it. This can slow down the primary system if there is currently high load that cannot be handled by the network connection. If the parameter is set to false, the connection to the secondary system will be temporarily closed to not impact the primary system. Later, the secondary can reconnect and sync using delta shipping.

Parameter	reconnect_time_interval
Type	Integer
Unit	seconds
Default	30
System	Secondary

Parameter	reconnect_time_interval
Description	If a secondary system is disconnected from the primary system because of network problems, the secondary system tries to reconnect periodically after the time interval specified in this parameter has passed.

Parameter	enable_full_sync
Type	Boolean
Default	false
System	Primary
Description	If set, system replication operates in full sync mode when the SYNC replication mode is set. In full sync mode, transaction processing blocks when the secondary is currently not connected and newly created log buffers cannot be shipped to the secondary system. This behavior ensures that no transaction can be locally committed without shipping to the secondary system. For more information, see *Full Sync Option for SAP HANA System Replication*.

Parameter	enable_log_compression
Type	Boolean
Default	false
System	Secondary
Description	If activated, log buffers will be compressed before sending them over the network to the secondary system. The secondary system decompresses the log buffers it receives and then writes them to disk. If the network bandwidth is the bottleneck in the system replication setup, log buffer compression can improve log shipping performance because less data is being sent over the network. The drawback to sending a compressed log buffer to the secondary system is that it requires additional time and processing power for compression and decompression. This can result in worse log shipping performance if turned on in a configuration with a fast network. The parameter has to be set on the secondary system. It can be changed online, but the secondary system has to re-connect to the primary system in order to activate the parameter change.

Parameter	enable_data_compression
Type	Boolean
Default	false
System	Secondary

Parameter	enable_data_compression
Description	If activated, data pages will be compressed before sending them over the network to the secondary system. The secondary system decompresses the data pages it receives and then writes them to disk. If the network bandwidth is the bottleneck in the system replication setup, data compression can improve log shipping performance because less data is being sent over the network. The drawback to sending compressed data pages to the secondary system is that it requires additional time and processing power for compression and decompression. This can result in worse data shipping performance if turned on in a configuration with a fast network. The parameter has to be set on the secondary system. It can be changed online, but the secondary system has to re-connect to the primary system in order to activate the parameter change.

Parameter	keep_old_style_alert
Type	Boolean
Default	false
System	Primary
Description	Before SPS 09 closed replication connections and configuration parameter mismatches were alerted with Alert 21. With SPS 09 two dedicated alerts have been introduced for both error situations. By default old style alerting is still offered for backwards compatibility. When setting this parameter to false, the old behavior is turned off and only new alerts will be generated. For more information, see *SAP HANA System Replication Alerts*.

Parameter	operation_mode
Type	enum
Values	delta_datashipping/logreplay/logreplay_readaccess
Default	logreplay
System	Secondary

Parameter	operation_mode
Description	Operation mode of the secondary site during replication. There are three different settings for this parameter: • delta_datashipping System Replication uses data and log shipping for replication. Log buffers received by the secondary system are just saved to disk, savepoints after intermediate delta data shippings truncate the log. Column table merges are not executed on the secondary system, but merged tables on the primary system are transported via delta data shippings to the secondary system. • logreplay System Replication uses an initial data shipping to initialize the secondary system. After that only log shipping is done and log buffers received by the secondary are replayed there. Savepoints are executed individually for each service and column table merges are executed on the secondary system. • logreplay_readaccess System Replication uses an initial data shipping to initialize the secondary system. After that only log shipping is done and log buffers received by the secondary are replayed there. Savepoints are executed individually for each service and column table merges are executed on the secondary system. Furthermore, read only access via SQL is possible to the secondary system. For more information, see *Operation Modes for SAP HANA System Replication*.

Parameter	enable_log_retention
Type	enum
Values	auto/off/on/force/force on takeover
Default	auto
System	Primary, Secondary

Parameter	`enable_log_retention`
Description	Enables or disables log retention on a system replication system. Log retention on the primary system is useful when the secondary should sync with the primary by re-shipping missing log after a network outage or downtime. If the missing log is not available anymore on the primary system, a data shipping is required (delta in operation mode `delta_datashipping`, full in all other operation modes). Log retention on the secondary system is needed to keep log for optimized re-sync during failback. Configuration options: • auto Log retention is automatically enabled if the secondary is in `logreplay` or `logreplay_readaccess` operation modes. For the `delta_datashipping` operation mode log retention is disabled. • on Log retention is enabled. • off Log retention is disabled. • force on takeover / force In multitarget replication use this option to retain the log for all secondaries. The value `force` is set automatically during takeover. For a detailed example of this scenario see *Log Retention and Multitarget System Replication*. When log retention is enabled and the system is configured as primary, the primary will not free log segments when the secondary system is disconnected, but keep them marked as `RetainedFree` for a potential optimized resync. When setting log retention explicitly to on or off, it should also be set for `delta_datashipping` operation mode or for failback with delta log shipping optimization. In the latter case after takeover to the secondary, the old primary can re-sync via missing log with the new primary system and no full data shipping is required for initialization.

Parameter	`logshipping_max_retention_size`
Type	Integer
Unit	MB
Default	1048576 (1TB)
System	Primary

Parameter	`logshipping_max_retention_size`
Description	Sets the maximum amount of log that will be kept for syncing a secondary system. This value only has an effect if log retention is enabled. Two situations have to be distinguished here: If `logshipping_max_retention_size` has been set to a value other than 0, when no secondary is connected log segments are not reused even if they are truncated and backed up until the max size limit has been reached or the system runs into a log full situation. If the maximum size limit is reached or in log full situations, the segments that are only kept for syncing the secondary system will be reused. This setting prevents the system from hanging on the primary system because of too many log segments that are held for syncing the secondary system. With this setting, the primary keeps running with the drawback that the secondary cannot sync anymore. If `logshipping_max_retention_size` is configured to 0, log segments required for syncing the secondary are not reused and a log full results in a system standstill on the primary system until log writing can continue. This setting allows you to configure an upper limit up to which redo log segments are kept in `RetainedFree` state on the primary system before they are overwritten for syncing with a secondary system. When the reason for the log full has been resolved, the transaction processing can continue. **Note** The default setting `logshipping_max_retention_size = 1048576 (MB)` of 1 TB means that 1 TB of size is configured for every service, which replicates data to a secondary system. That is, every service owning a persistence in form of data and log volume. **Example** If the services nameserver, two indexservers (for example, two tenant databases) and an xsengine are running in your SAP HANA system, the total configured log retention size will be 4 TB (4 x 1 TB). With this setting it can happen that the disk full is reached before the RetainedFree marked log segments are overwritten. If you want to change the default value of 1 TB, you can do this in the global.ini. Another option is to set this parameter in the service ini files individually. For example, if the value is set in the global.ini of the system database, in the global.ini of a tenant database, and in the indexserver.ini of a tenant database, the indexserver.ini setting would win and will be taken for log retention of this indexserver.

Parameter	`datashipping_parallel_channels`
Type	Integer
Default	4
System	Secondary

Parameter	datashipping_parallel_channels
Description	The parameter defines the number of network channels used by full or delta datashipping. The actual number of channels for each shipping can be adjusted by the system to reduce overhead depending on the current amount of data to be shipped. Higher parallelism can be useful when large amounts of data (above several GB at least) needs to be shipped and the utilization of network bandwidth by single network stream is low. Please note that the overall bandwidth is still limited by the I/O bandwidth because the data needs to be read from the primary system. To deactivate the parameter, change the default to 0.

Parameter	enable_takeover_with_uninitialized_volumes
Type	Boolean
Default	False
System	Secondary
Description	The parameter can be used to prevent a takeover from taking place if the databases have not been initialized. This can be applied either in strict mode (set to false) or relaxed mode (set to true). In relaxed mode takeover is possible if at least the system database has been initialized and in this case a warning message is displayed if other databases are not initialized.

Parameter	alternative_sources
Type	VARCHAR
Default	
System	Secondary
Description	Candidate source sites to register to when the current source site is unavailable. Syntax: alternative_sources=SiteA,SiteB,... or alternative_sources=SiteA:sync,SiteB:async,..

Related Information

Full Sync Option for SAP HANA System Replication [page 800]
SAP HANA System Replication Alerts [page 865]
Operation Modes for SAP HANA System Replication [page 773]
Log Retention and Multitarget System Replication [page 784]
Change a System Property in SAP HANA Studio
Log Retention [page 781]
SAP Note 2400007

12.1.2.8.7 SAP HANA System Replication Command Line Reference

This topic provides details of the supported system replication options for the command line tool hdbnsutil.

Command	**-sr_enable**
Options	[--name=<site alias>]
System	Primary
Online/Offline	Online
Description	Enables a system to serve as a system replication source system. In multitier and multitarget setups the --name= option is mandatory. Use `-sr_enable` to enable the source system for any further tier that is added to the system replication landscape.

Command	**-sr_disable**
System	Primary
Online/Offline	Online
Description	Disables system replication capabilities on the primary system.

Command	**-sr_register**
System	Secondary
Online/Offline	Offline
Description	• `--remoteHost=<primary master host>` Registers a system to a source system and creates the replication path for the system replication. • `--remoteInstance=<primary instance id>` • `--replicationMode=sync|syncmem|async` Specifies the replication mode. • `--operationMode=delta_datashipping|logreplay|logreplay_readaccess` Specifies the operation mode. • `--name=<unique site name>` Specifies the system name. • `[--online]` If the system is running you can use this parameter to automatically perform a system restart. Not relevant if the system is shut down. • `[--force_full_replica]` If a parameter is given, a full data shipping is initiated. Otherwise a delta data shipping is attempted. • `[--withAllSecondaries]` This option helps to manage reconfiguration of sites in a multitarget replication landscape. You can use this to re-attach a secondary to another source system within the landscape; any existing sub-trees will remain attached. All sites in the landscape must be online when you run the command.

Command	**-sr_unregister**
Options	[--id=<site id>|--name=<site name>]

Command	-sr_unregister
System	Primary, Secondary
Online/Offline	Secondary offline Primary online (to remove metadata)
Description	Unregisters a secondary system from its source. Use this command on the secondary that needs to be unregistered. When the secondary system is not available, this command can also run on the primary. In this case use either the –id option or the –name option to identify the secondary system. **Note** There are three scenarios in which it is necessary to unregister system replication: • When the secondary system is available, but should be de-coupled permanently You will be able to use the secondary system as a standard SAP HANA installation afterwards. • When the secondary system is not available anymore and the primary system needs to be cleaned up in order to be able to register a new system This can occur when the secondary system was uninstalled or when it cannot be recovered after a disaster. • When you want to re-establish the original setup after a takeover in a multitier system replication configuration For more information, see *Restore the Original SAP HANA Multitier System Replication Configuration*. To understand how to use the `-sr_unregister` command correctly, see *SAP Note 1945676*.

Command	-sr_initialize
Options	--database=<tenantDB>|--volume=<volume id> [--force_full_replica]
System	Primary
Online/Offline	Online
Description	Initializes a given database or specific volume for system replication. If parameter [--force_full_replica] is given, a full data shipping is initiated, otherwise a delta data shipping is attempted. See also SAP Note 2980989 - *How-To: Performing a Full Data Shipment for a Single Volume / Service*. This command is normally not required as initialization takes place automatically. It may be required in exceptional circumstances to reinitialize one specific tenant where replication needs to be restarted.

Command	-sr_fullsync
Options	--enable |--disable
System	Primary
Online/Offline	Online and offline

Command	-sr_fullsync
Description	For use with the SYNC replication mode to ensure replication consistency. Full sync mode ensures that no transaction can be committed on the primary without shipping the log buffers to the secondary. See *Full Sync Option for SAP HANA System Replication*.

Command	-sr_changeReplicationMode
Options	--mode=sync|syncmem|async
System	Secondary
Online/Offline	Online and offline
Description	Changes the replication mode of a secondary system.

Command	-sr_takeover
Options	[--suspendPrimary|--maxWriteTransactionWaitTime=<time_s>]
System	Secondary
Online/Offline	Online and offline
Description	Makes this secondary the new primary system. The parameter --suspendPrimary is used for the take-over with handshake option; in this case you can also specify for how long the primary should wait for running write transactions to complete before suspending the primary (by default there is no wait time).

Command	-sr_resumeSuspendedPrimary
System	Primary
Description	Unblock and resume a suspended primary after a takeover with handshake. In this case there is no safeguard by SAP HANA against multiple active primaries.

Command	-sr_state
System	Primary and Secondary
Online/Offline	Online and offline
Description	Shows status information about a system replication system.

Related Information

Full Sync Option for SAP HANA System Replication [page 800]
Example: Restore the Original SAP HANA Multitier System Replication Configuration [page 844]
SAP Note 1945676
SAP Note 2980989

12.1.2.8.8 Add and Remove Hosts in SAP HANA System Replication

You can add a new host to a replicated system with the SAP HANA lifecycle manager.

Context

Background information about adding hosts to a replication system is provided in the *SAP HANA Administration Guide for SAP HANA Platform*, this includes an illustrated topic 'Host Addition Concepts' showing different scenarios. Hosts must be added equally to both primary and secondary sites. You are recommended to add a host to the secondary site before adding it to the primary site, this avoids the situation where the new host saves data without first being in sync. A further recommendation if you are adding multiple hosts is to add them one at a time so that they can be correctly matched on primary and secondary.

You can review details of the configuration of the system replication topology of all replication systems using various tools as described in the section Monitoring System Replication. To confirm the details of the host mappings between sites you can use the `-sr_state` parameter of the command line tool hdbnsutil which lists details for all sites in the system. See the examples in the topic 'Monitoring SAP HANA System Replication with hdbnsutil'.

> **Note**
>
> Do not turn off system replication when adding a host.

Procedure

1. Add a host to the secondary site and start it.
2. Add a host to the primary site and start it.

   Replication begins automatically.
3. To remove a host, first remove it from the primary site and then remove the host from the secondary site.

Related Information

Host Addition Concepts [page 1086]
Add Hosts Using the Command-Line Interface [page 1092]
Remove Hosts Using the Command-Line Interface [page 1100]
Monitoring System Replication
Monitoring SAP HANA System Replication with hdbnsutil [page 873]

12.1.2.9 Performing a Takeover

The takeover process is the name for the task of switching your active system from the current primary system to the secondary system.

If your primary data center is not available, for example because of planned downtime or due to a disaster, and a decision has been made to fail over to the secondary data center, you can perform a takeover on your secondary system.

We recommend that you use third-party external tools to check if hosts, the network, and data center are still available.

To help you decide if a takeover is advisable with regard to system availability, downtime and the risk of data loss, see the decision tree in *SAP Note 2063657 SAP HANA System Replication Takeover Decision Guideline*. In addition, you can use python scripts such as `getTakeoverRecommendation.py` to help you decide when a takeover should be carried out. For a detailed description of the available python scripts, see 'Checking the SAP HANA System Replication Status' in the *SAP HANA System Replication Guide*.

The takeover does not include stopping the former primary system. If you are performing a takeover as part of a planned downtime, first make sure that the primary system has been fully stopped.

If you wish to avoid stopping the primary system you can also simply isolate it in the landscape by putting it behind a firewall. It is also not necessary to completely stop the primary if you use the command line option 'takeover with handshake' (`hdbnsutil -sr_takeover --suspendPrimary`), see *Takeover with Handshake* for details.

If necessary, you can verify that the primary is no longer available using the STONITH command which checks for the existence of a failed host. A scenario where this command is used is given in the section 'Example HA/DR Provider Implementation'. Additionally, you can activate a check to ensure that the database(s) have been initialized - refer to the configuration parameter `enable_takeover_with_uninitialized_volumes` in 'SAP HANA System Replication Configuration Parameters'.

Once the takeover command completes, the former secondary system becomes the new active primary system.

You can perform a takeover using the following tools:

- SAP HANA cockpit
  For more information, see the topic in the SAP HANA Replication Guide *Perform a Takeover with the SAP HANA Cockpit*.
- SAP HANA studio
  For more information, see the SAP HANA studio guide *Perform a Takeover with the SAP HANA Studio*.
- hdbnsutil
  For more information, see *Perform a Takeover with hdbnsutil*.

Related Information

Perform a Takeover
Perform a Takeover with the SAP HANA Studio
Perform a Takeover with hdbnsutil [page 816]
Checking the SAP HANA System Replication Status

SAP HANA System Replication Configuration Parameters [page 801]
Takeover with Handshake [page 824]
Example HA/DR Provider Implementation [page 916]
SAP Note 2063657

12.1.2.9.1 Perform a Takeover with hdbnsutil

You can perform a takeover on your secondary system with the `hdbnsutil` command line tool.

Prerequisites

- The secondary system must be fully initialized.
  You can check this in M_SERVICE_REPLICATION or in the SAP HANA studio *Administration Console* > *Landscape* > *System Replication*. The secondary system is ready for takeover if all services display *REPLICATION_STATUS ACTIVE*.
- The takeover command can be executed both when the secondary system is offline or online.

> **Note**
>
> If you are performing a takeover as part of a planned downtime, you should first make sure that the primary system has been fully stopped before performing a takeover to the secondary system.

Procedure

As `<sid>`adm enter the following command on the secondary system to enable the secondary system to take over and become the primary system:

```
cd /usr/sap/<sid>/HDB<instancenr>/exe
```

```
./hdbnsutil -sr_takeover [--comment="Your Comment"]
```

Use the `--comment` to add a reason for the takeover (for example, to distinguish between a planned or unplanned takeover). This comment is displayed in the M_SYSTEM_REPLICATION_TAKEOVER_HISTORY monitoring view in the COMMENTS column.

If the system is offline, the takeover is actually carried out when the system is started again.

Related Information

Stop a System

Monitoring SAP HANA Systems During Stop and Start
M_SERVICE_REPLICATION System View
M_SYSTEM_REPLICATION_TAKEOVER_HISTORY System View

12.1.2.9.2 Client Connection Recovery After Takeover

Connection recovery after a takeover can be done with network-based IP redirection or network-based DNS redirection.

After a takeover, the client or the application server need to be able to continuously reach the SAP HANA system, no matter which system is currently the primary system after takeover.

After a takeover, the new primary database server is not aware of previous connections which existed between clients and the former primary server. If the client application does not issue a new request and keeps waiting for a reply from the server, it will not receive an explicit request to close these connections from either of the servers and will keep waiting indefinitely. To prevent this, the SAP HANA client library supports the TCP keepalive feature provided by the operating system. This feature will lead the client to abort the invalid connection on its end and to trigger a reconnect after a specified period during which the former primary server is not reachable.

However, the default keepalive settings for the operating system (2 hours) may lead the client processes to wait for a long time before they abort the connection on their end and trigger a reconnect with the new primary system. For example, the default Linux settings leave the clients waiting for more than two hours before aborting the connection. For more information on how to configure the keepalive settings to match your needs see SAP Note 2053504 *System replication: Hanging client processes after a takeover* and the corresponding documentation for your operating system.

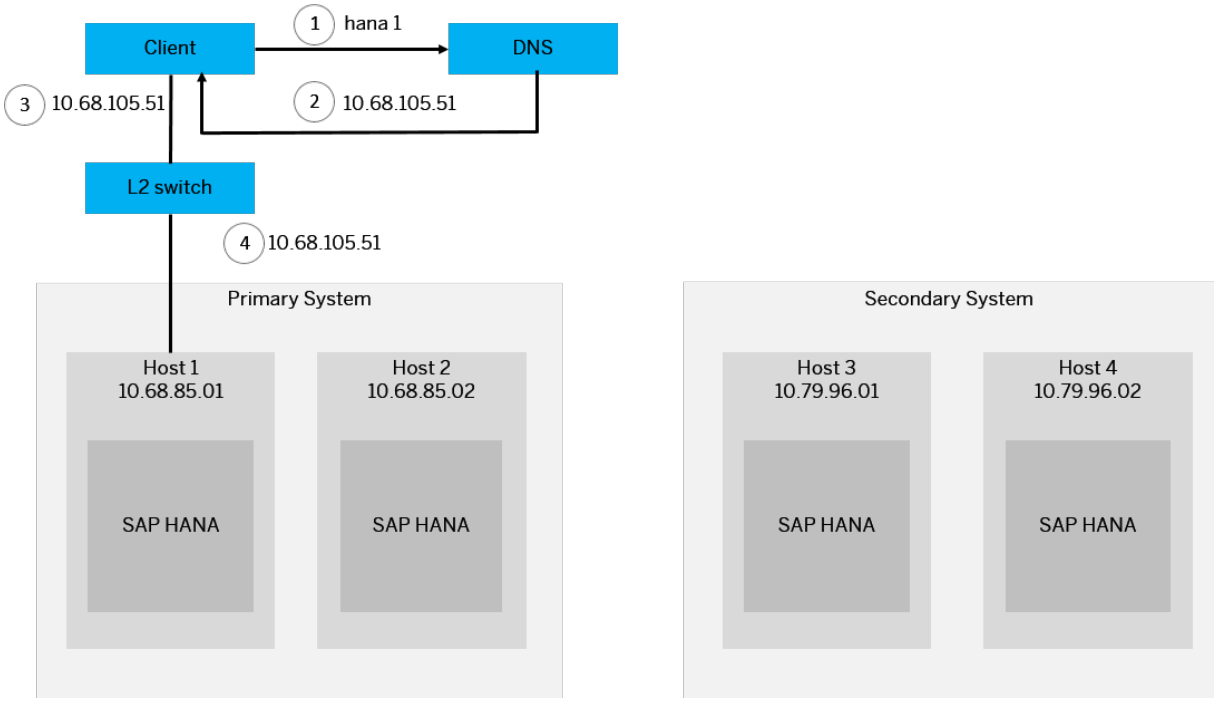
There are different possibilities for enabling client connection recovery:

1. IP redirection
   A virtual IP address is assigned to the virtual host name. In case of a takeover, the virtual IP will unbind from the network adapter of the primary system and bind to the adapter on the secondary system.
2. DNS redirection
   In this scenario the IP for the host name in the DNS will be changed from the address of the primary to the address of the secondary system.

Both methods have their advantages. If there are no existing constraints, the IP redirection has the clear benefit of being faster to process in a script rather than synchronizing changes of DNS entries over a global network.

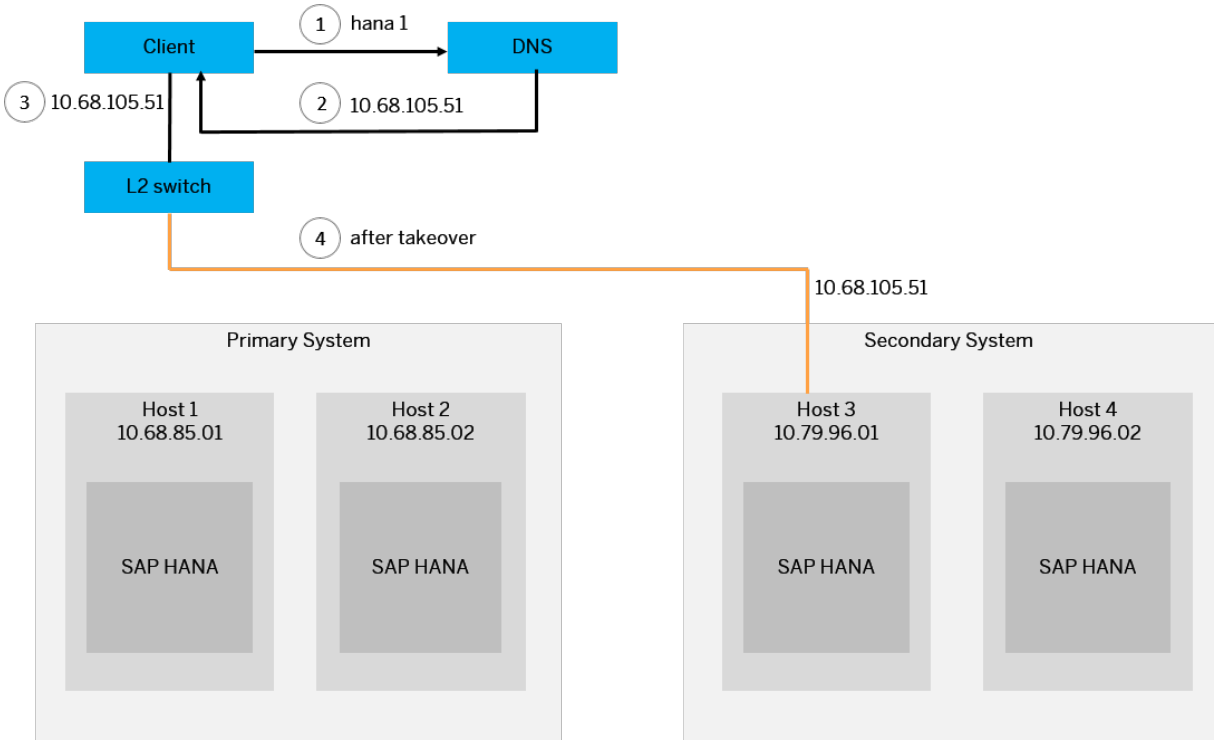
1. Network-based IP Redirection

The principle of IP redirection is to define an additional "logical" host name (hana1, in the diagram below) with its own separate logical IP address (10.68.105.51), and then map this initially to the MAC address of the original host in the primary system (by binding it to one of the host's interfaces):



As part of the takeover procedure, a script is executed which re-maps the unchanged logical IP address to the corresponding takeover host in the secondary system. This must be done pair-wise, for each host in the

primary system. The remapping affects the L2 (OSI layer 2: data link) switching, as can be seen in step 4 of the following diagram:

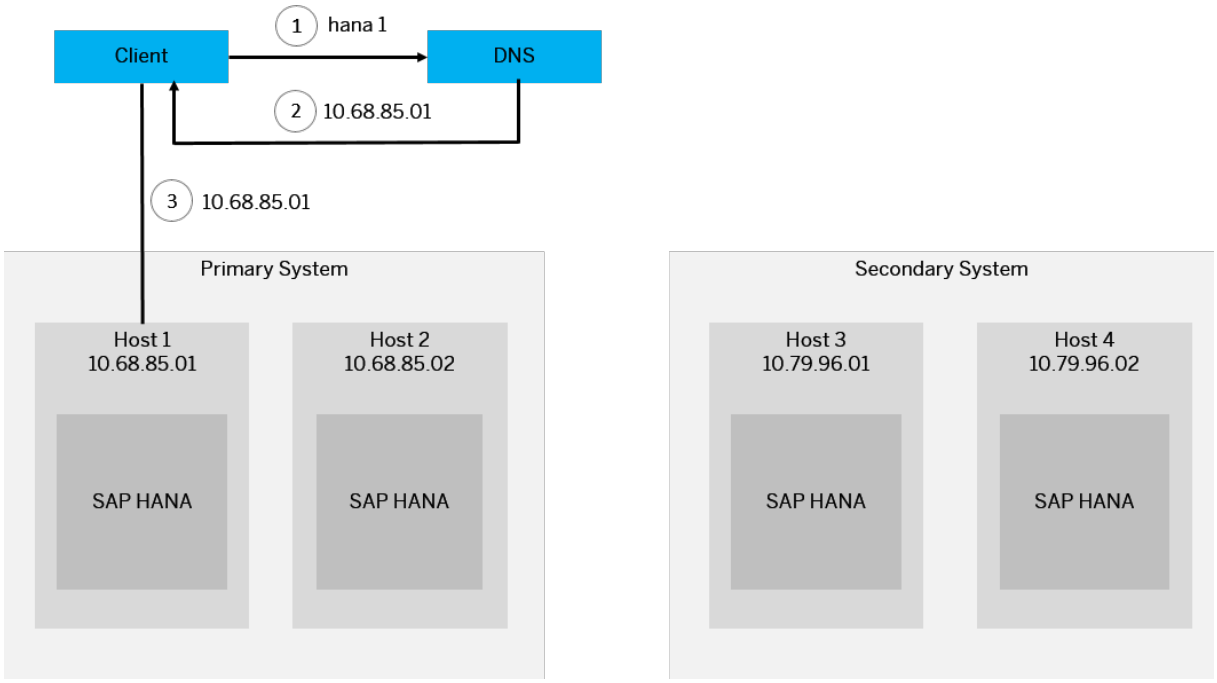


IP redirection can be implemented using a number of actual techniques, for example with the use of Linux commands which affect the network Address Resolution Protocol (ARP) tables (ip addr add/del...), by configuring L2 network switches directly, or by using cluster management software. Following the IP redirection configuration, the ARP caches should be flushed, to provide an almost instantaneous recovery experience to clients.

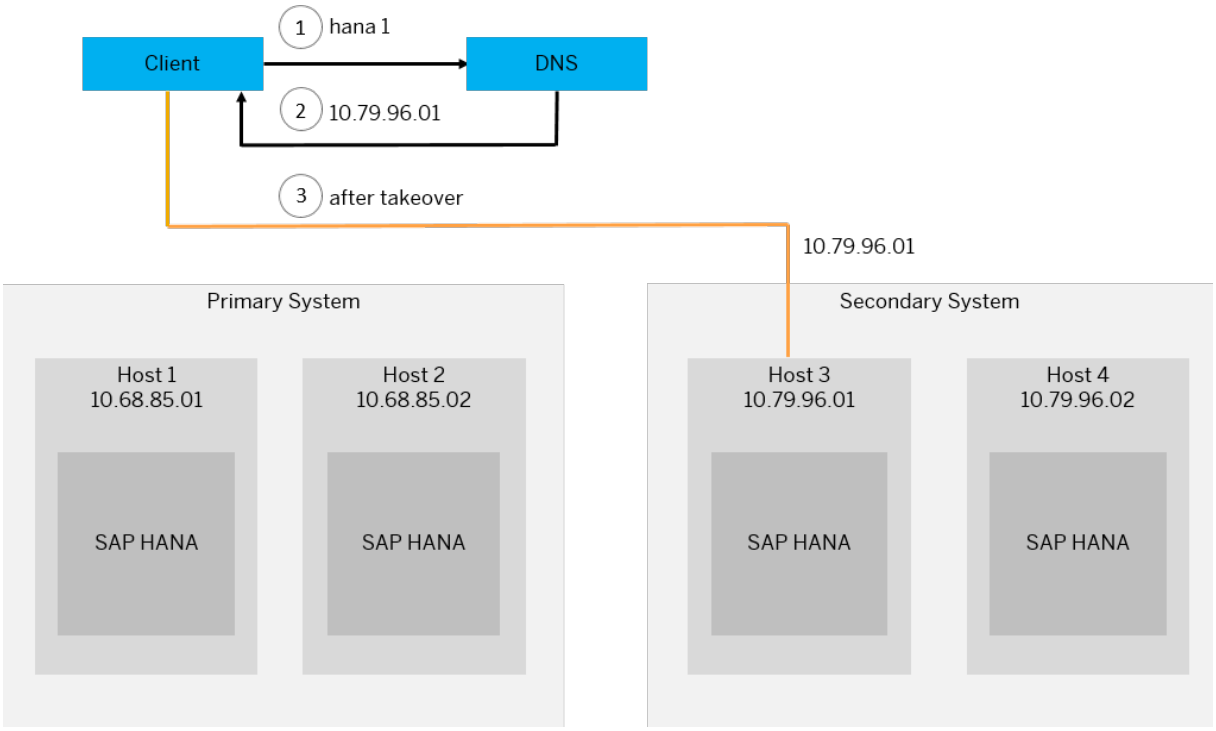
IP redirection requires that both the primary and failover host(s) are on the same L2 network. If the standby system is in a completely separate L3 network, then DNS redirection is the preferred alternative solution.

2. Network-based DNS Redirection

DNS redirection is an alternative to IP redirection. DNS is a binding from a logical domain name to an IP address. Clients contact a DNS server to obtain the IP address of the SAP HANA host (step 1 below) they wish to reach:



As part of the takeover procedure, a script is executed that changes the DNS name-to-IP mapping from the primary host to the corresponding host in the secondary system (pair-wise for all hosts in the system). From that point in time, clients are redirected to the failover hosts:



This solution shares the advantage with IP redirection that there are no client-specific configurations. Further, it supports disaster recovery configurations where the primary and secondary standby systems may be in two completely different network domains (separated by routers). One drawback of this solution is that modifying DNS mappings requires a vendor-proprietary solution. Further, due to DNS caching in nodes (both clients and intermediate network equipment), it may take a while (up to hours) until the DNS changes are propagated, causing clients to experience downtime despite the recovery of the system.

HA/DR providers can inform external entities about activities inside SAP HANA scale-out (such as Host Auto-Failover) and SAP HANA system replication setups. Actions can be defined in a Python script which should be executed before or after certain activities (such as, startup, shutdown, failover, takeover, connection changed, service state changed). One application of this is to move virtual IP addresses after takeover in SAP HANA system replication. Additionally, external cluster management software can be used to perform the client reconnect after takeover. For more information, see *Implementing HA/DR Providers*.

Related Information

Implementing a HA/DR Provider [page 905]
Connecting Using Active/Active (Read Enabled) (SAP HANA Platform)
SAP Note 2053504

12.1.2.9.3 Invisible Takeover and Restart

During an invisible takeover or a restart, the session's state needs to be recovered and restored to the new primary system.

During a standard takeover you switch your active system from the current primary system to the secondary system. After a standard takeover, the primary system loses all connections to the client. Moreover, the secondary system is not aware of the previous connections, which existed between the client and the primary system. This is different in an invisible takeover.

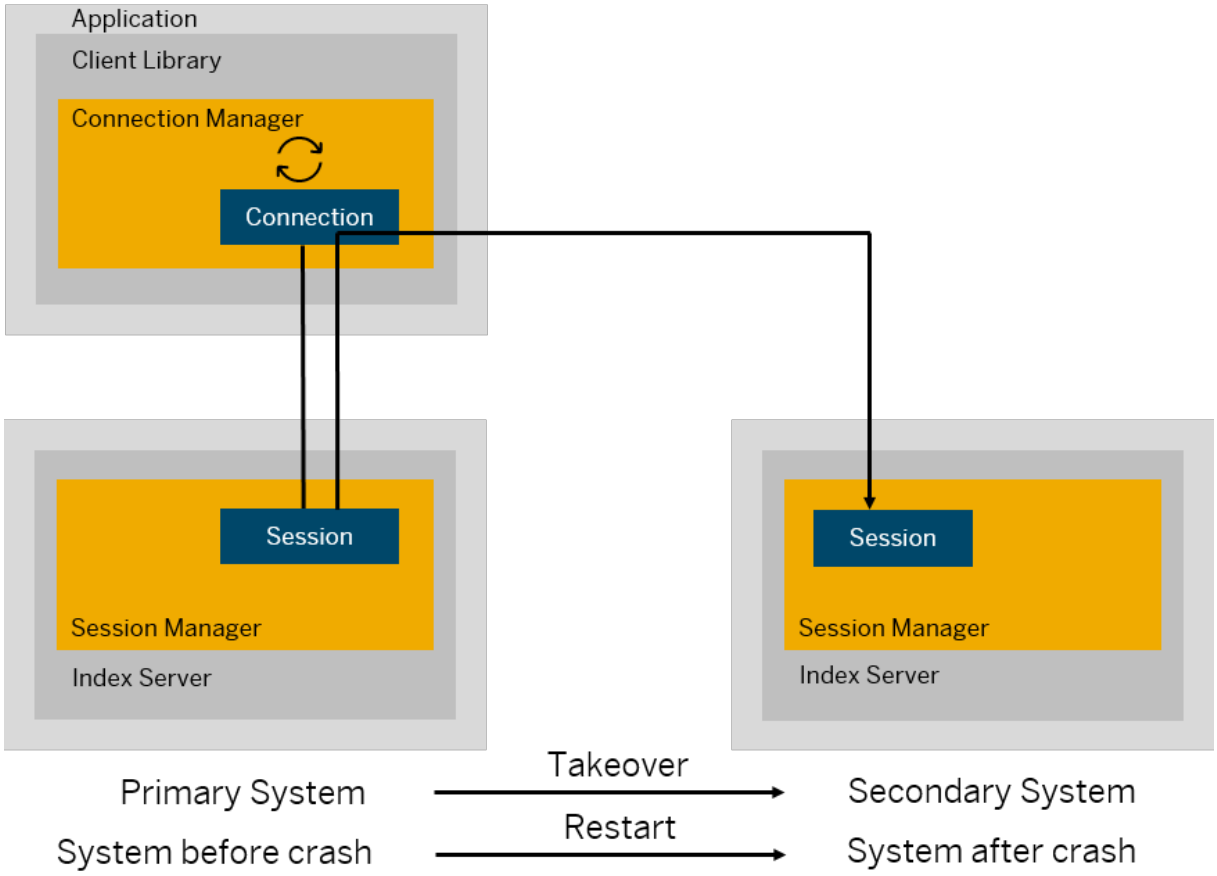
You can perform an invisible takeover to achieve an automatic recovery of your sessions after takeover to your new primary system. For dedicated client applications this takeover will be invisible. Differently from a standard takeover, an invisible takeover ensures that the client reconnects to the primary system and the sessions are restored to the secondary system.

This seamless recovery is possible also when restarting the system (for example, after a system crashes).

The session's state needs to be recovered and restored to the new primary system in an invisible takeover scenario or to the new system in a restart scenario. The cross-layer between the session and the client library makes the seamless recovery possible. This cross-layer feature called *transparent session recovery* recovers the current session's state and the physical connection.

> ### Note
>
> The transparent session recovery is supported by JDBC and SQLDBC for SAP HANA 2.0.

Use the graphic below to better understand this process in an invisible takeover or a restart scenario after the active system crashed:



Configuration

The `enable_session_recovery` parameter in the session section of the ini file controls the session recovery. The default value is *true* recovering all session variables and restoring the client connections from the primary system to the secondary system. This parameter is configurable online, but the changes can be applied only to the connections established after making the changes.

Limitations

- Sessions which have created or updated a global temporary table with any DDL or DML commands won't be recovered. However, sessions which have created a local temporary table will be recovered without the table recovery.
- Sessions which have application locks acquired by ACQUIRE_APPLICATION_LOCK won't be recovered.
- Ongoing write transaction will be rolled back with an error and the session can be recovered when an application restarts the failed transaction with no explicit reconnect trial from the application.
- Almost all session variables from the current session context are recovered.

- If the client has sent a request to the server but the server has not successfully responded back, the session is not recovered. However, sessions are still recovered when a SQL command is not sent from the client to the server.

12.1.2.9.4 Takeover with Handshake

The takeover with handshake can be used for a planned takeover (in a system maintenance scenario) to perform a safe (and 'graceful') takeover.

During a planned takeover, it is important to ensure that no data gets lost (all primary updates must be available on the secondary system) and that the former primary system is isolated to avoid a situation with multiple active primary systems ('split-brain'). The takeover with handshake is ideal for this scenario since the primary is suspended but continues to run. All new writing transactions on the primary system are suspended and the takeover is only executed when all redo log is available on the secondary system. It is not necessary to check the replication status or to stop the old primary before the takeover.

The following limitations apply:

- This option is only supported on the second tier.
- Takeover with handshake is not supported with Dynamic Tiering services.

> ⓘ Note
>
> The takeover with handshake is only performed if the required conditions are met, that is: log is transferred and synchronized with the secondary, and the primary is suspended with replication services still active. If this is not the case the takeover is aborted and reported as an error. In that case, there is no impact on the system and the replication remains as it was.
>
> Two options are available for takeover, suspendPrimary and stopPrimary.
>
> - `SuspendPrimary`ensures that no data are lost during the takeover, and that no write transactions can be carried out on the primary system after the takeover. However, as the primary system remains online after the takeover, it is possible that a client application may still be connected to the primary system and may still run read statements. To prevent this, you can isolate the primary system, for example, using a firewall or by disconnecting physical network ports, or you can instead use the --stopPrimary command.
> - `StopPrimary` additionally automatically stops the primary when the takeover is completed.

Takeover with SuspendPrimary

You can trigger a takeover with handshake using the --suspendPrimary command on the secondary system:

```
hdbnsutil -sr_takeover --suspendPrimary ––maxWriteTransactionWaitTime=500
```

With suspendPrimary, the optional parameter maxWriteTransactionWaitTime can be used to specify a time period in seconds to determine how long the primary should wait for any running write transactions to complete before suspending the primary (by default there is no wait time).

After the takeover, the suspended primary is unblocked when you register it as the new secondary as shown in the following example command line:

```
hdbnsutil -sr_register --remoteHost=myHost --remoteInstance=50
```

```
--replicationMode=sync --operationMode=logreplay --name=SiteA --online
```

Alternatively, the suspended primary can be unblocked and continue to be used in its existing role using the commands shown here. Note that in this case there is no safeguard against having two active primaries.

Command	Action
hdbnsutil -sr_unregister --name=<SiteB>	Unregister the old secondary.
hdbnsutil -sr_resumeSuspendedPrimary	Resume the suspended primary (execute this on the primary).

See also SAP Note 3193844 - How to unblock from suspendPrimary mode.

Takeover with StopPrimary

The stopPrimary command carries out the following steps in sequence:

1. Suspend the primary system
2. Synchronize the secondary system
3. Stop the primary system
4. Trigger takeover by the secondary system

Use the --stopPrimary command on the secondary system:

```
hdbnsutil -sr_takeover --stopPrimary
```

> **Note**
>
> `StopPrimary` will stop the primary immediately after the secondary system has been synchronized.

Related Information

Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 885]
SAP HANA System Replication Command Line Reference [page 811]

12.1.2.9.5 Automatic Registration After Takeover

In multi-target replication scenarios you can automate the process of registration after a takeover.

An option is available to automatically re-register the secondaries in the landscape which were previously registered before the takeover. This is particularly useful in multi-target replication scenarios and helps to avoid the normal procedure of having to shut down, register and restart each secondary.

To use this feature, set the parameter `register_secondaries_on_takeover` to TRUE (in the `[system_replication]` section of the global.ini file). The parameter must be set on the new primary. After connecting to the source all services are then automatically restarted and synchronized with the new primary.

This case is illustrated in the topic *Disaster Recovery Scenarios for Multitarget System Replication* where secondary sites in a different data center are registered to a new primary.

You can check the status of each server after a takeover by querying the view M_SERVICE_REPLICATION which has details of the status of all replication sites in the landscape.

Related Information

Disaster Recovery Scenarios for Multitarget System Replication [page 848]

12.1.2.10 Secondary Time Travel

You can start the secondary system or the log replay on the secondary system at a previous point in time.

Secondary time travel is a temporary takeover which is effective at a previous point in time. Once secondary time travel has been configured, snapshots and additional log data are retained on the secondary system for a defined period of time. These are then used to start the system at a selected point in the past by opening the appropriate snapshot and replaying the required log entries to reach the requested point. For this reason secondary time travel only works with the logreplay operation modes (`logreplay` or `logreplay_readaccess`). Secondary time travel is started using the command: `hdbnsutil -sr_timetravel` with additional options to specify the start time.

> ⓘ Note
>
> You can safely use the SYNC replication mode with the full sync option; secondary time travel maintains a connection to the primary system and does not risk leading to a freeze of the primary (see Replication Modes).

Secondary time travel can be used in two ways:

1. To quickly access again data, which was deleted in the original system. For more information, see *Execute Secondary Time Travel*.
2. To intentionally keep the secondary system's log replay delayed. This can be used with Active/Active (Read Enabled) to read older data from the secondary system while the secondary keeps replicating. For more information, see *Execute Secondary Time Travel While Replication Continues*.

Using option one, the secondary system does a takeover up to the specified point in time so that data can be read. It is then necessary to resync with the primary so that the secondary is again available for a complete takeover. You can do this using the `-sr_register` command line option as described in *Configure SAP HANA System Replication with hdbnsutil*.

If time travel is configured in a muliti-target replication scenario it would be possible, for example, to maintain a tier 3 disaster recovery host which is reserved for accessing data in time travel mode.

Option two uses an additional command line parameter: `startMode=replicate`. In this case, the secondary system is restarted but there is no takeover and replication of log data continues. The system restarts at the specified point in time and log replay is stopped at this point. Commands are then available to manually advance the log replay or to resume automatic log replay. This option offers more flexibility; because there is

no takeover you can restart the system repeatedly if necessary at any available snapshot. The earliest point in time is defined by the setting of the `timetravel_max_retention_time` parameter described below.

Using Active / Active in this scenario (that is, with operation mode `logreplay_readaccess` - a license is required) it is possible to then read data at the latest point of log replay. Operation mode `logreplay` does not provide read access to the delayed replay.

Note that using this option in a muliti-target replication scenario it is only possible to read data at tier 2.
Also, if data will be read using hint based statement routing, a specific server must be named; a configuration parameter is available for this: `hint_based_routing_site_name`.

Preparation: Configuration Parameters

Configuration parameters for secondary time travel can be found in the system_replication section of the global.ini file, these must be configured on the secondary system. See *Configuration Parameters* for details.

To prepare the system for time travel, the `timetravel_max_retention_time` parameter must be set on the secondary system. This parameter defines the time period to which the secondary system can be brought back in the past. After setting this parameter, the secondary starts retaining log data and snapshots.

Optionally, the `timetravel_snapshot_creation_interval` parameter can be modified to adjust the frequency of snapshot creation. The default value of this parameter is 1440 minutes (1 day).

These parameters should be maintained in relation to each other, for example, the snapshot interval time will be less than or equal to the max retention time. The parameter with the greater value determines the length of time for which logs and snapshots must be retained on the system.

Monitoring

Monitoring of time travel is possible using the system view M_SYSTEM_REPLICATION_TIMETRAVEL which includes details such as start times and redo log positions. Also M_SYSTEM_REPLICATION_TAKEOVER_HISTORY shows secondary time travel events. A time travel takeover is shown with the value 'TIMETRAVEL' in the TAKEOVER_TYPE column and any comments entered with the time travel commands are visible in the COMMENTS column.

Security: Root Keys

Because of security requirements the active root keys (visible in view ENCRYPTION_ROOT_KEYS) in the restored secondary system may have timestamps which are more recent than the specified time travel date. The latest root keys are required because old root keys are only used for decryption.

Related Information

Execute Secondary Time Travel [page 828]
Configuration Parameters [page 830]
Execute Secondary Time Travel While Replication Continues [page 829]
Monitoring Secondary Time Travel [page 831]
Replication Modes for SAP HANA System Replication [page 771]
Configure SAP HANA System Replication with hdbnsutil [page 793]

12.1.2.10.1 Execute Secondary Time Travel

You can start the secondary system in online mode at a previous point in time to access again data, which was deleted in the original system.

Prerequisites

- Set the requiered parameters to define the point in time to which the secondary system can be brought back in the past. For a full list of available parameters for secondary time travel, see *Configuration Parameters*.

  > **Note**
  >
  > Set the parameters carefully to avoid log full or disk full situations. For time travel to work, log and snapshots are kept online in the data area. Because of this, log and data will grow on the secondary system when time travel is turned on. The system workload determines how much data is needed.

- Use the `logreplay` or the `logreplay_readaccess` operation mode in your system replication setup.

Procedure

1. Stop the secondary system.
2. Execute `hdbnsutil -sr_timetravel --startTime=<startTime> [--callTakeoverHooks=on|off] [--comment="Your Comment"]`

   For `startTime` use the following format specified in UTC: dd.mm.yyyy-hh.mm.ss

   You can specify if takeover hooks should be called. If the `timetravel_call_takeover_hooks` parameter is not explicitly specified, takeover hooks won't be called. For more information on takeover hooks, see *Implementing a HA/DR Provider*.

   Use the `--comment` to add a reason for the time travel. This comment is displayed in the M_SYSTEM_REPLICATION_TAKEOVER_HISTORY monitoring view in the COMMENTS column.
3. Start the secondary system.

   The secondary system will enter in online mode at the specified point in time during restart. After restart, the other services read the requested point in time and open their persistence using this information. If the requested point in time cannot be reached, then time travel will be aborted. A check ensures that there are time travel snapshots older than the start time for each service.

Related Information

Configuration Parameters [page 830]
M_SYSTEM_REPLICATION_TAKEOVER_HISTORY System View

Implementing a HA/DR Provider [page 905]

12.1.2.10.2 Execute Secondary Time Travel While Replication Continues

You can start the log replay at a previous point in time to read older data from the secondary system, while the secondary keeps replicating.

Prerequisites

- Set the required parameters to define the point in time to which the secondary system can be brought back in the past. For a full list of available parameters for secondary time travel, see *Configuration Parameters*.

  > **Note**
  >
  > Set the parameters carefully to avoid log full or disk full situations. For time travel to work, log and snapshots are kept online in the data area. Because of this, log and data will grow on the secondary system when time travel is turned on. The system workload determines how much data is needed.

- The `logreplay` or `logreplay_readaccess` operation modes are supported but in order to read data from the secondary a license for Active / Active (Read Enabled) is required with operation mode logreplay_readaccess. Operation mode logreplay does not provide read access.

Procedure

1. Stop the secondary system.
2. Execute `hdbnsutil -sr_timetravel --startTime=<startTime> --startMode=replicate`

   For `startTime` use the following format specified in UTC: dd.mm.yyyy-hh.mm.ss
3. Start the secondary system.

   After the system has started, the persistence has been opened on the specified point in time, it is replicating log, and log replay is not running.
4. Optional: Trigger the log replay manually with `hdbnsutil -sr_recoveruntil {--endTime=<timestamp>|max} [--nowait]`

   For `<timestamp>` use the following format specified in UTC: dd.mm.yyyy-hh.mm.ss

   Use `max` to trigger the log replay up to the newest possible point in time. In this case, the target timestamp is automatically determined by checking the valid time travel range for each service.

   Use `--nowait` to specify if the command should be executed asynchronously.
5. Optional: Stop the manual replay mode by setting the `timetravel_logreplay_mode` parameter back to auto or using `hdbnsutil -sr_replaymode --mode={auto|manual}`

Related Information

Configuration Parameters [page 830]

12.1.2.10.3 Configuration Parameters

Use the following parameters to prepare your system for secondary time travel.

The parameters are defined in the system_replication section of the INI file. All parameters are set on the secondary system.

Parameter	**timetravel_max_retention_time**
Type	integer
Unit	minutes
Default	0
Description	If set to 0, secondary time travel is turned off. If this parameter is set to a value different from 0, the secondary system can be brought online up to the defined point in the past.

Parameter	**timetravel_snapshot_creation_interval**
Type	integer
Unit	minutes
Default	1440
Description	Defines how frequently snapshots are created for secondary time travel. Time travel snapshots are kept until they get older than the defined `timetravel_max_retention_time` parameter. If time travel needs to be done on an older point in time, the snapshot that best fits the requested point in time will be opened and the remaining changes are applied via log replay. A new snapshot is created when the point in time defined in this parameter has passed since the last snapshot creation. Snapshots older than the point in time defined in `time_travel_max_retention_time` are dropped.

Parameter	**timetravel_call_takeover_hooks**
Type	bool
Values	true, false
Default	false

Parameter	timetravel_call_takeover_hooks
Description	Indicates if takeover hooks should be called during secondary time travel.

Parameter	timetravel_logreplay_mode
Type	enum
Values	auto, manual
Default	auto
Description	Defines how the log replay is executed on the secondary system. The following settings are allowed: • Auto The log replay is done automatically up to the newest possible log position. • Manual You must manually trigger the log replay up to the requested timestamp using the `-sr_recoveruntil` hdbnsutil command.

12.1.2.10.4 Monitoring Secondary Time Travel

You can monitor the retaining log and the created snapshots.

To monitor secondary time travel, the secondary system must be online. The current time travel range cannot be determined, when the secondary is offline.

You can determine the valid range for which time travel can be executed in two ways:

- Using `hdbnsutil -sr_timetravel --printRange`
  This command provides a range for each service in which time travel can be executed:

Value	Description
START_TIME	Contains the oldest possible point in time for which timetravel can be executed. As time travel is done for all services, the intersection of all ranges have to be checked to make sure, all services can reach the specified timestamp.
END_TIME	Contains the last possible point in time for which timetravel can be executed. For worker services, this timestamp can be some time back without being outdated, if there was no more activity on this worker for some time. To ignore the worker volumes, only the lines for transaction coordinating host can be considered. Those are marked as MASTER in the COORDINATOR_TYPE column.

- Using SQL on the primary system via the _SYS_DATABASES_SR_SITE_<sitename>.M_SYSTEM_REPLICATION_TIMETRAVEL secondary proxy view.

The start time or the log position of the system can be monitored using M_SYSTEM_REPLICATION_TAKEOVER_HISTORY.

12.1.2.11 Performing a Failback

After a takeover has been carried out the roles between primary and secondary can be switched over.

In the case of a failover, the former primary has to be registered as the secondary with the now active primary system. The roles are switched compared to the original setup.

You can perform a failover using the following tools:

- SAP HANA cockpit
  For more information, see the topic *Perform a Failback* in the SAP HANA cockpit guide.
- SAP HANA studio
  For more information, see the topic *Perform a Failback with the SAP HANA Studio* in the SAP HANA studio guide.
- hdbnsutil
  For more information, see *Perform a Failback with hdbnsutil*.

Related Information

Perform a Failback (SAP HANA Cockpit)
Perform a Failback (SAP HANA Studio)
Perform a Failback with hdbnsutil [page 832]

12.1.2.11.1 Perform a Failback with hdbnsutil

You can perform a failback using the hdbnsutil command line tool.

Prerequisites

- You need the operating system user to perform a failback. For more information, see *Operating System User <sid>adm*.
- The former primary system is not running.
- The current primary system is running.

Context

This is the same procedure as is used for setting up a normal secondary described in steps 2 and 3 of *Configure SAP HANA System Replication with hdbnsutil*. To fail back, attach your original primary system as new secondary system to the current primary system. These are identified in the following parameters of the command: --remoteHost=<new primary hostname>, --name=<new secondarySiteName>. Refer also to the hdbnsutil example *Example: Configure SAP HANA System Replication* where the primary and secondary are identified as dcsite1 and dcsite2 respectively.

Procedure

When the former primary is available again and offline, it can be registered as the new secondary.

```
hdbnsutil -sr_register --remoteHost=<new primary hostname>
--remoteInstance=<instance number>
--replicationMode=<sync/syncmem/async>
--operationMode=<delta_datashipping|logreplay|logreplay_readaccess>
--name=<new secondarySiteName>
```

Related Information

Operating System User <sid>adm [page 552]
Configure SAP HANA System Replication with hdbnsutil [page 793]
Example: Configure SAP HANA System Replication [page 796]

12.1.2.12 Disabling SAP HANA System Replication

Remove a system replication configuration when you want to run the two systems separately or if you don't need this high availability and disaster recovery mechanism anymore.

To remove a system replication configuration, unregister the secondary and disable the primary system.

> ⚠ Caution
>
> Keep in mind that after unregistering, the unregistered system is still active and applications may be connected to it. You must therefore ensure that applications are connected to the correct server. Potential problem situations, for example, are where replication with Smart Data Integration is active or where shared log backup locations are used. Refer to the following SAP notes for more information:
>
> - 1945676 - Correct usage of hdbnsutil -sr_unregister. This note describes the use cases of the –`sr_unregister` command
> - 2904125 - Data loss can occur in split-brain state when/if system replication is disabled while SDI is active

You can disable system replication using the following tools:

- SAP HANA cockpit
  For more information, see *Disable SAP HANA System Replication with the SAP HANA Cockpit* in the SAP HANA cockpit guide.
- SAP HANA studio
  For more information, see *Disable SAP HANA System Replication with the SAP HANA Studio* in the SAP HANA studio guide.
- hdbnsutil
  For more information, see *Disable SAP HANA System Replication with hdbnsutil*.

> **Note**
>
> There are only two scenarios where it is necessary to unregister system replication:
>
> - When the secondary system is available, but should be de-coupled permanently
>   You will be able to use the secondary system as a standard SAP HANA installation afterwards.
> - When the secondary system is not available anymore and the primary system needs to be cleaned up in order to be able to register a new system
>   This can occur when the secondary system was uninstalled or when it cannot be recovered after a disaster.

Related Information

Disable SAP HANA System Replication (SAP HANA Cockpit)
Disable SAP HANA System Replication (SAP HANA Studio)
Disable SAP HANA System Replication with hdbnsutil [page 834]
SAP Note 1945676
SAP Note 2904125

12.1.2.12.1 Disable SAP HANA System Replication with hdbnsutil

You can disable SAP HANA system replication with `hdbnsutil`.

Prerequisites

- You are logged on to both systems as the operating system user (user <sid>adm) or are able to enter these credentials when prompted.
- The secondary system must be offline.

Procedure

1. Stop the secondary system:

   ```
   sapcontrol -nr <instance_number> -function StopSystem HDB
   ```

2. On secondary system unregister the secondary system:

   ```
   hdbnsutil -sr_unregister
   ```

3. Start the secondary system.

   The unregister command is only effective when the system restarts.

   > **Note**
   >
   > In cases where system replication is out of sync and you just need to re-register the initial secondary system, unregister is not required - simply use the command `hdbnsutil -sr_register`.
   >
   > For details of the two cases where `hdbnsutil -sr_unregister` is required see *SAP Note 1945676 - Correct usage of hdbnsutil -sr_unregister*.

4. Disable system replication on the primary system as follows:

   ```
   hdbnsutil -sr_disable
   ```

Related Information

SAP HANA System Replication Command Line Reference [page 811]
SAP Note 1945676

12.1.2.13 SAP HANA Multitier System Replication

To offer higher levels of availability, you can link multiple systems in a SAP HANA multitier system replication landscape.

You can configure system replication to support geo-clustering, that is multitier system replication between a primary data center and other geographically remote data centers to form a single highly available system.

Related Information

Configuring SAP HANA Multitier System Replication [page 836]
Performing a Takeover and a Failback in SAP HANA Multitier System Replication [page 843]

12.1.2.13.1 Configuring SAP HANA Multitier System Replication

With multitier system replication, a tier 2 system replication setup can be used as the source for replication in a chained setup of primary system, tier 2 secondary system, and tier 3 secondary system.

After configuring a basic system replication scenario, you must add a third system to provide another level of redundancy. In a multitier setup, the primary system is always on tier 1, a tier 2 secondary has a primary system as its replication source, and a tier 3 secondary has the tier 2 secondary as its replication source.

The operation mode must be the same for all systems. However, if `logreplay_readaccess` is used between tier 1 and tier 2, (see Active/Active Read Enabled for details) as an exception from this rule only the `logreplay` operation mode can be used between tier 2 and tier 3. Furthermore, it is not possible to combine the `logreplay` and `delta_datashipping` operation modes.

Multitier system replication supports various replication mode combinations. For more information, see *Supported Replication Modes Between Systems*.

You can configure multitier system replication using the following tools:

- SAP HANA cockpit
  For more information, see *Example: Configure SAP HANA Multitier System Replication with the SAP HANA Cockpit*.
- SAP HANA studio
  For more information, see *Example: Configure SAP HANA Multitier System Replication with the SAP HANA Studio*.
- hdbnsutil
  For more information, see *Example: Configure SAP HANA Multitier System Replication with hdbnsutil*.

Related Information

Supported Replication Modes Between Systems [page 837]
Example: Configure SAP HANA Multitier System Replication with SAP HANA Cockpit
Example: Configure SAP HANA Multitier System Replication with hdbnsutil [page 841]
Example: Configure SAP HANA Multitier System Replication with SAP HANA Studio [page 839]
Operation Modes for SAP HANA System Replication [page 773]
Active/Active (Read Enabled) [page 852]

12.1.2.13.1.1 Supported Replication Modes Between Systems

In a multitier system replication scenario, the following replication mode combinations are supported.

Replication Mode Combinations

See also Replication Modes for SAP HANA System Replication.

Tier1 to Tier 2	Tier 2 to Tier 3	Description	Use Case
SYNC	SYNC	In this setup, tier 1, tier 2, and tier 3 are coupled with SYNC replication mode. Tier 2 sends the acknowledgment to tier 1 after the log buffer has been received and written to disk, and after the log buffer has also been received and written by tier 3. When primary has received the acknowledge, the buffer has been persisted by all the tiers.	Tier 1 and tier 2 are located in a local data center for fast takeover. Tier 3 is used for disaster recovery in a second close data center.
SYNC	SYNCMEM	Tier 2 sends the acknowledge to tier 1 after the log buffer has been received, written to disk and it has been also received by tier 3. When the primary receives acknowledgment, it is not clear that also tier 3 has persisted the buffer to disk, but disk IO on tier 3 has been triggered.	Tier 1 and tier 2 are located in a local data center for fast takeover. Tier 3 is used for disaster recovery in a second close data center.
SYNC	ASYNC	Tier 1 and tier 2 are closely coupled with replication mode SYNC, while tier 3 is decoupled by using ASYNC. Tier 2 acknowledges the arrival of the redo log buffers in-memory and on disk to tier 1, while it only hands over the redo log buffer to the network without awaiting an acknowledgment from tier 3. If the connection to tier 3 is too slow and the ASYNC replication buffer (an intermediate memory buffer) is running full, ASYNC replication to tier 3 can have an impact on the primary.	Tier 1 and tier 2 are located in a local data center for fast takeover. Tier 3 is used for disaster recovery in a far distant data center.
SYNCMEM	SYNC	In this synchronous setup tier 1 and tier 2 are closely coupled with replication mode SYNCMEM, while tier 3 is closely coupled with SYNC. Tier 2 sends the acknowledgment to tier 1 after the log buffer has been received in memory. IO is triggered asynchronously. The asynchronous IO	Tier 1 and tier 2 are located in a local data center for fast takeover. Tier 3 is used for disaster recovery in a second close data center.

Tier1 to Tier 2	Tier 2 to Tier 3	Description	Use Case
		also triggers the send operation to tier 3. The log write on tier 2 is confirmed, when also tier 3 has written the log buffer. When the primary receives the acknowledge, it is unclear, if tier 3 has already received and persisted the log buffer.	
SYNCMEM	SYNCMEM	In this setup tier 1, tier 2, and tier 3 are coupled with replication mode SYNCMEM. Tier 2 sends the acknowledgment to tier 1 after the log buffer has been received in memory. IO is triggered asynchronously. The asynchronous IO also triggers the send operation to tier 3. The log write on tier 2 is confirmed, when tier 3 has received the log buffer in memory. When the primary receives the acknowledge, it is unclear, if tier 3 has already received and persisted the log buffer.	Tier 1 and tier 2 are located in a local data center for fast takeover. Tier 3 is used for disaster recovery in a second close data center.
SYNCMEM	ASYNC	Tier 1 and tier 2 are closely coupled with replication mode SYNCMEM, while tier-3 is decoupled with ASYNC replication. Tier 2 acknowledges the arrival of the redo log buffers in-memory to tier 1, while it only hands over the redo log buffer to the network without awaiting an acknowledgment from tier 3. If the connection to tier 3 is too slow and the ASYNC replication buffer (an intermediate memory buffer) is running full, ASYNC replication can have an impact on the primary.	Primary and tier 2 are used in a local data center for fast takeover. Tier 3 is used for disaster recovery in a far distant data center.
ASYNC	SYNC	ASYNC mode is used for maximum performance: there is no wait for acknowledgment between tiers 1 and 2. Information about the replication status on tier 1 and tier 2 is available in the ASYNC replication buffer (an intermediate memory buffer). This buffer running full could have a minimal impact on the performance of the primary. In this case tiers 2 and 3 are independent of tier 1 and a replication backlog for tier 2 and tier 3 is possible.	Tiers 1 and tier 2 may be located some distance apart. Tier 3 is used for high availability of tier-2 in a second close data center. This use case is illustrated with a commentary in the topic 'SAP HANA Multitarget System Replication'.

Tier1 to Tier 2	Tier 2 to Tier 3	Description	Use Case
		The send operation to tier 3 is triggered asynchronously but the log write on tier 2 is only confirmed when tier 3 has also written the log buffer.	
ASYNC	SYNCMEM	ASYNC mode is used for maximum performance between tiers 1 and 2. SYNCMEM also gives good performance, the log write on tier 2 is confirmed when tier 3 has received the log buffer in memory.	Tier 1 and tier 2 may be located some distance apart. Tier 3 may be used for high availability of tier-2 in a second data center.
ASYNC	ASYNC	With these asynchronous replication modes there is no wait for acknowledgments between tiers (no acknowledge propagation). A replication backlog for tier 2 and tier 3 is possible. Information about the replication status on tier 1 and tier 2 is available in the ASYNC replication buffer (an intermediate memory buffer). This buffer running full could cause a minimal impact on the performance of the primary.	Tier 1 performance is most important as well as a disaster recovery capability. For best performance of tier 1 decouple tier 2 and tier 3. Data loss on tier 2 and tier 3 is possible to some extent, but performance is more critical.

Related Information

Replication Modes for SAP HANA System Replication [page 771]
SAP HANA Multitarget System Replication [page 846]

12.1.2.13.1.2 Example: Configure SAP HANA Multitier System Replication with SAP HANA Studio

You can configure SAP HANA multitier system replication using the SAP HANA studio.

Prerequisites

- You have considered all the general prerequisities needed to set up system replication. For more information, see *General Prerequisites for Configuring SAP HANA System Replication*.
- You have added the systems in the SAP HANA studio.

- You have verified that the `log_mode` parameter in the `persistence` section of the `global.ini` file is set to **`normal`** for the systems.
  You can do this in the Administration editor (*Configuration* tab) of the SAP HANA studio.
- You have stopped the tier 3 secondary system.

Context

The following example describes how to add a tier 3 secondary with a synchronously running tier 2 system replication.

Procedure

1. Enable system replication on the tier 2 secondary, which has to be online, as follows:
   a. In the *Systems* view right click the tier 2 secondary system, choose *Configuration and Monitoring* > *Configure System Replication*
   The *Configure System Replication* dialog opens. The *Enable System Replication* option is selected by default. The site name is already known from the topology metadata.
   b. Choose *Next*.
   c. Review the configured information and choose *Finish*.
2. Register the tier 3 secondary system as follows:
   a. You have installed and configured three identical, independently operationalStop the tier 3 secondary system if it is still running. Right-click the tier 3 secondary system and choose *Configuration and Monitoring* > *Stop System*
   b. In the *Systems* view, right-click the tier 3 secondary system and choose *Configuration and Monitoring* > *Configure System Replication*.
   The *Configure System Replication* dialog opens.
   c. Choose *Register Secondary System* and then *Next*.
   d. Enter the required system information and the logical name used to represent the tier 3 secondary system.

      > **Note**
      > If you are operating a distributed system on multiple hosts, you enter the name of the host on which the master name server is running.

   e. Specify the replication mode and enter the tier 2 secondary system's host name.

      For more information, see *Supported Replication Modes Between Systems*.
   f. Review the configured information and choose *Finish*.

Results

The secondary system is automatically started and the replication process to the tier 3 secondary then starts automatically.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 765]
Supported Replication Modes Between Systems [page 837]

12.1.2.13.1.3 Example: Configure SAP HANA Multitier System Replication with hdbnsutil

Learn how to configure SAP HANA multitier system replication with the hdbnsutil command line.

Prerequisites

- You have considered all the general prerequisities needed to configure system replication. For more information, see *General Prerequisites for Configuring SAP HANA System Replication*.
- You have installed and configured three identical, independently operational SAP HANA systems – a primary system, a tier 2 secondary system, and a tier 3 secondary system.

Context

The following steps show how to configure such a system. In this scenario there are three SAP HANA systems: A, B, and C, named SiteA, SiteB, and SiteC. Furthermore, in this scenario multitier system replication supports a tier 2 secondary with sync replication mode and a tier 3 secondary with async replication mode. The operation mode is `logreplay`.

Procedure

1. [A] Start the SAP HANA database.
2. [A] Create a data backup or storage snapshot. In multiple-container systems, the system database and all tenant databases must be backed up.

3. [A] Enable system replication and give the system a logical name. As <sid>adm:

```
cd /usr/sap/<sid>/HDB<instancenr>/exe
```

```
./hdbnsutil -sr_enable --name=SiteA
```

4. Stop the tier 2 secondary.

   As <sid>adm run the SAPControl program to shut down the system:

```
/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> –function StopSystem
HDB
```

5. [B] On the stopped tier 2 secondary, register site B with Site A as <sid>adm:

```
hdbnsutil -sr_register --replicationMode=sync --operationMode=logreplay --
name=SiteB
--remoteInstance=<instId> --remoteHost=<hostname_of_A>
```

6. [B] Start the tier 2 secondary system.

   As <sid>adm run the SAPControl program to start the system:

```
/usr/sap/hostctrl/exe/sapcontrol sapcontrol -nr <system number> -function
StartSystem HDB
```

7. [B] Enable this site as the source for a tier 3 secondary system:

   As <sid>adm on the tier 2 secondary run `hdbnsutil -sr_enable`

8. [C] Stop the tier 3 secondary system.

   As <sid>adm run the SAPControl program to shut down the system:

```
/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> –function StopSystem
HDB
```

9. [C] On the stopped system, register siteC as a tier 3 secondary system as <sid>adm:

```
hdbnsutil -sr_register --replicationMode=async --operationMode=logreplay --
name=SiteC
--remoteInstance=<instId> --remoteHost=<hostname_of_B>
```

10. [C] Start the SAP HANA database on the tier 3 secondary.

    As <sid>adm run the SAPControl program to start the system:

```
/usr/sap/hostctrl/exe/sapcontrol sapcontrol -nr <system number> -function
StartSystem HDB
```

11. Check the replication status with systemReplicationStatus.py on command line or in the M_SERVICE_REPLICATION system view.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 765]
Supported Replication Modes Between Systems [page 837]
Operation Modes for SAP HANA System Replication [page 773]

12.1.2.13.2 Performing a Takeover and a Failback in SAP HANA Multitier System Replication

You can perform a takeover and a failback also in a SAP HANA multitier system replication.

If the primary system fails, you can perform a takeover to the tier 2 secondary system.

Once your failed system is operational again, you can attach it as a tier 3 secondary system or you can restore the original multitier system replication configuration. To learn how to perform these steps with the hdbnsutil command line, see *Attach the Original Primary System as a New Tier 3 Secondary System* and *Restore the Original SAP HANA Multitier System Replication Configuration*.

Related Information

Example: Attach the Original Primary System as a New Tier 3 Secondary System [page 843]
Example: Restore the Original SAP HANA Multitier System Replication Configuration [page 844]

12.1.2.13.2.1 Example: Attach the Original Primary System as a New Tier 3 Secondary System

Once your failed system is operational again, you can attach it as a tier 3 secondary system.

Context

The steps below show how to set up multitier system replication again after a takeover. In these scenarios there are three SAP HANA systems A, B and C, named SiteA, SiteB, and SiteC. Furthermore, in this scenario multitier system replication supports a tier 2 secondary with sync replication mode and a tier 3 secondary with async replication mode.

Procedure

SiteA failed, SiteB has taken over and now you attach SiteA as the tier 3 secondary.

1. [C] Change the replication mode of the new tier 2 secondary:

   ```
   cd /usr/sap/<sid>/HDB<instance_number>/exe
   ./hdbnsutil -sr_changemode --replicationMode=sync
   ```

   Multitier system replication supports various replication mode combinations. For more information, see *Supported Replication Modes between Sites*.

2. [C] Enable SiteC as the replication source:

```
hdbnsutil -sr_enable
```

3. [A] Make sure that the SAP HANA database is stopped. This should be the case as a takeover was already carried out otherwise you can stop it with the following command:

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StopSystem
HDB
```

4. [A] Register SiteA as a new tier 3 secondary.

```
hdbnsutil -sr_register --replicationMode=async --name=SiteA --
remoteInstance=<instId> --remoteHost=<hostname_of_C>
```

5. [A] Start the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StartSystem
HDB
```

6. [B] Check in M_SERVICE_REPLICATION that sync system replication is ACTIVE from SiteB to SiteC and that async replication is ACTIVE from SiteC to SiteA.

Related Information

Supported Replication Modes Between Systems [page 837]

12.1.2.13.2.2 Example: Restore the Original SAP HANA Multitier System Replication Configuration

Once your failed system is operational again, you can restore the original SAP HANA multitier system replication configuration.

Context

The steps below show how to set up multitier system replication again after a takeover. In these scenarios there are three SAP HANA systems A, B and C, named SiteA, SiteB, and SiteC. Furthermore, in this scenario multitier system replication supports a tier 2 secondary with sync replication mode and a tier 3 secondary with async replication mode.

Procedure

You want to restore the original multitier setup:

1. [C] Stop the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StopSystem
HDB
```

2. [A] Stop the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StopSystem
HDB
```

3. [A] Register as secondary:

```
hdbnsutil -sr_register --replicationMode=sync --name=SiteA --
remoteInstance=<instId> --remoteHost=<hostname_of_B>
```

4. [A] Start the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StartSystem
HDB
```

5. [B] Check in M_SERVICE_REPLICATION that sync system replication is ACTIVE from SiteB to SiteA.
6. [A] SiteA takes over as the primary system:

```
hdbnsutil -sr_takeover
```

7. [B] Stop the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StopSystem
HDB
```

8. [A] Enable system replication:

```
hdbnsutil -sr_enable --name=SiteA
```

9. [B] Register SiteB as the tier 2 secondary of SiteA.

```
hdbnsutil -sr_register --replicationMode=sync --name=SiteB --
remoteInstance=<instId> --remoteHost=<hostname_of_A>
```

10. [B] Start the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StartSystem
HDB
```

11. [B] Enable SiteB as a replication source system:

```
hdbnsutil -sr_enable
```

12. [C] Register SiteC as a tier 3 secondary in the multitier system replication scenario:

```
hdbnsutil -sr_register --replicationMode=async --name=SiteC --
remoteInstance=<instId> --remoteHost=<hostname_of_B>
```

13. [C] Start the SAP HANA database

```
/usr/sap/hostctrl/exe/sapcontrol -no <instance_number> –function StartSystem
HDB
```

14. [B] Check in M_SERVICE_REPLICATION that sync replication is ACTIVE from SiteA to SiteB and that async replication is ACTIVE from SiteB to SiteC.

12.1.2.14 SAP HANA Multitarget System Replication

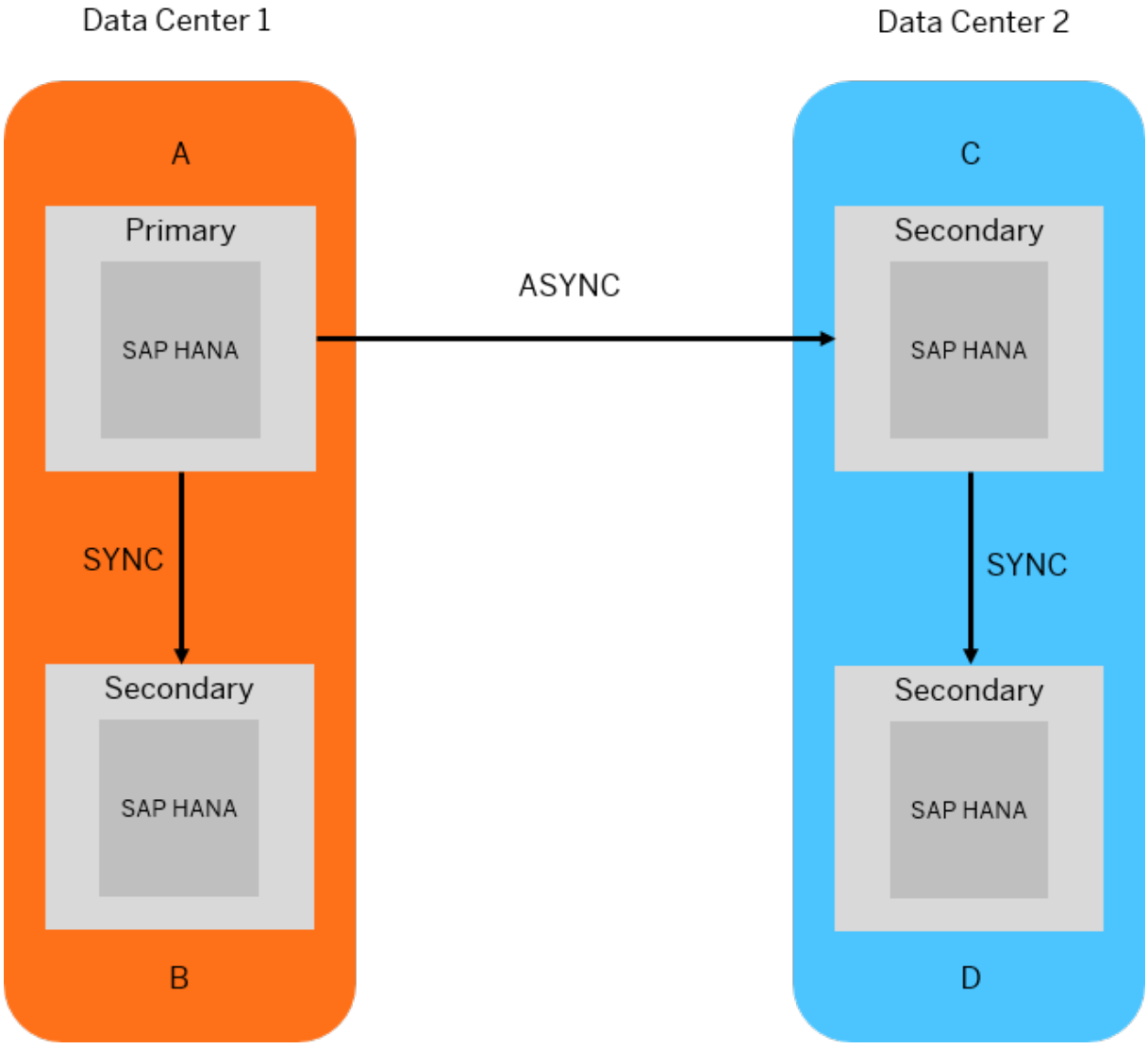
In a multitarget system replication, primary and secondary systems can replicate data changes to more than one system.

Multitarget system replication brings advantages for several use cases:

- Update scenarios
- Rearrangements of system replication multitier chains
- Reaching higher availability (before stopping existing structures, new structures can be built and established)

The graphic shows a possible setup for multitarget system replication.

Primary system A in data center 1 replicates data changes to secondary system B in the same data center. Primary system A also replicates data changes to secondary system C in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center with system C.



To understand how to handle different disaster recovery scenarios, see *Disaster Recovery Scenarios for Multitarget System Replication*. In a multitarget system replication, the secondary systems can be configured to automatically re-register to a new source system when the original source system unavailable.

For multitarget system replication either the `logreplay` or `logreplay_readaccess` mode is required.

In a multitarget system replication setup, you can configure multiple secondaries as Active/Active (read enabled). Only one of these secondaries can be accessed via hint-based statement routing; the others must be accessed via direct connection. For more information on Active/Active (read enabled), see *Active/Active (Read Enabled)*.

Related Information

Example: Configure a SAP HANA Multitarget System Replication [page 847]
Operation Modes for SAP HANA System Replication [page 773]
Disaster Recovery Scenarios for Multitarget System Replication [page 848]
Full Sync Option for SAP HANA System Replication [page 800]
Log Retention [page 781]
Log Retention and Multitarget System Replication [page 784]
Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites [page 895]
Use Multitarget System Replication for Near Zero Downtime Upgrades [page 888]
Active/Active (Read Enabled) [page 852]

12.1.2.14.1 Example: Configure a SAP HANA Multitarget System Replication

You can configure a multitarget system replication in which a primary system replicates data changes to more than one secondary system.

Context

In this example, primary system A in data center 1 replicates data changes to secondary system B in the same data center. Primary system A also replicates data changes to secondary system C in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center with system C.

Procedure

1. On primary system A in data center 1:
   a. Create backups.

b. Enable system replication.

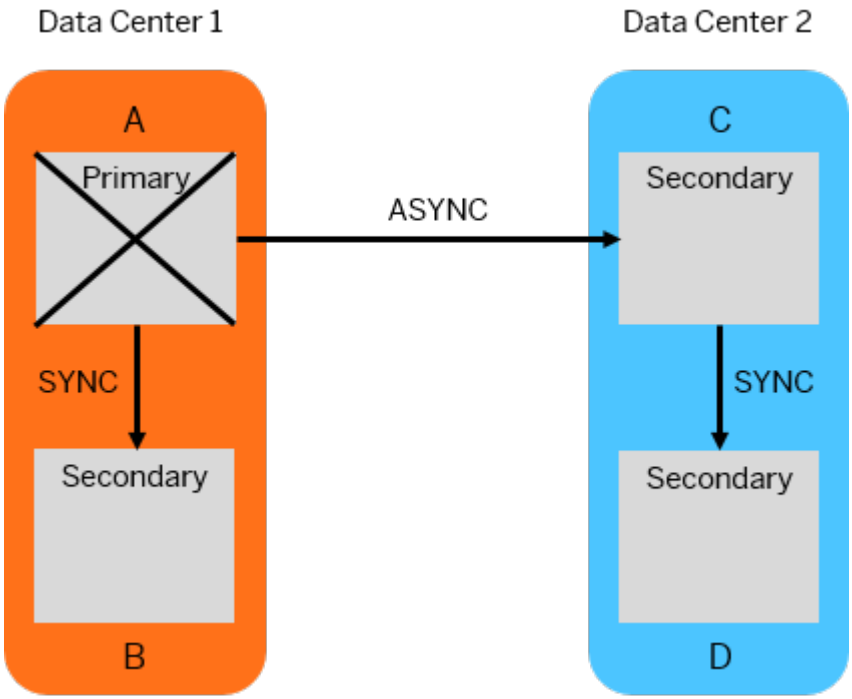
2. On the local secondary system B in data center 1:
   a. Stop the system.
   b. Register it to A.
   c. Start the system.
3. On the remote secondary system C in data center 2:
   a. Stop the system.
   b. Register it to A.
   c. Start the system.
4. On the remote secondary system D in data center 2:
   a. Stop the system.
   b. Register it to C.
   c. Start the system.

12.1.2.14.2 Disaster Recovery Scenarios for Multitarget System Replication

Several solutions are available when the systems involved in a multitarget system replication configuration fail.

We are using the setup described in *Multitarget System Replication* to exemplify the procedure. In this setup, primary system A replicates data changes to secondary system B located in the same data center. Primary system A also replicates data changes to the secondary system C located in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center with system C.

Failure on Primary System A



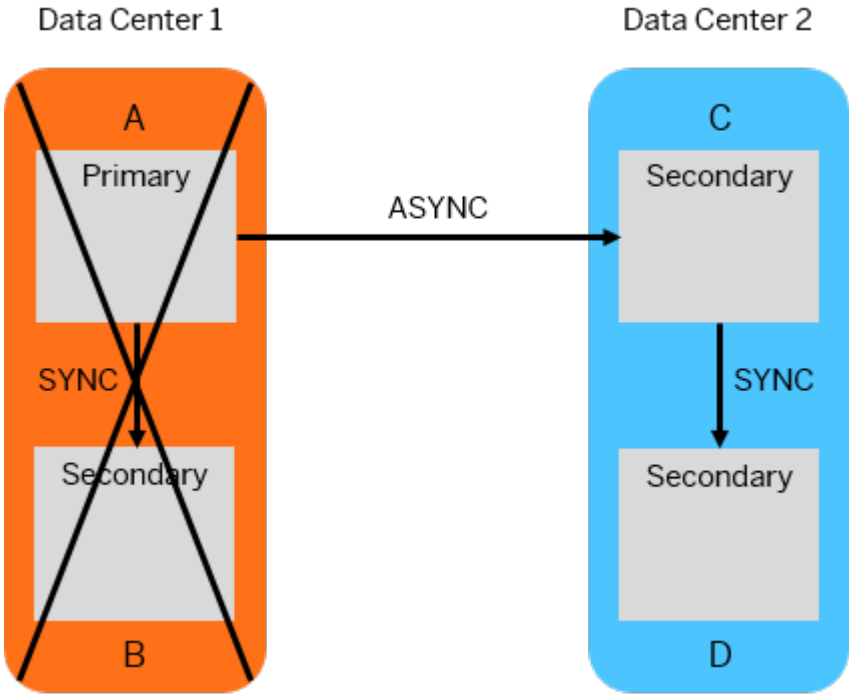
When primary system A fails, proceed as follows:

1. Take over on secondary system B in data center 1.
2. Register secondary system C in data center 2 to the new primary system B in data center 1. Then, register secondary system D in data center 2 to secondary system C.
3. After the failure on the previous primary system A is solved, register it to the new primary system B in data center 1.

Alternatively, you can set the `global.ini/[system_replication]/register_secondaries_on_takeover` parameter to **`true`** and take over on secondary system B in data center 1. As a result, secondary system C in data center 2 will register automatically to the new primary system B in data center, while secondary system D in data center 2 will register automatically to secondary system C. Using this option the replication mode already in use is retained when the secondaries are re-registered.

After the failure on the previous primary system A is solved, register it to the new primary system B in data center 1. See also the additional options described in *Automatic Registration After Takeover*.

Failure of Data Center 1



When all the systems in data center 1 fail, proceed as follows:

1. Take over on secondary system C in data center 2.
2. After the failure on the previous primary system is solved, register system A to the new primary system C in data center 2.
3. Register secondary system B as tier 3 to system A in data center 1.

For more information about takeover and failback, see *Performing a Takeover* and *Performing a Failback*.

Related Information

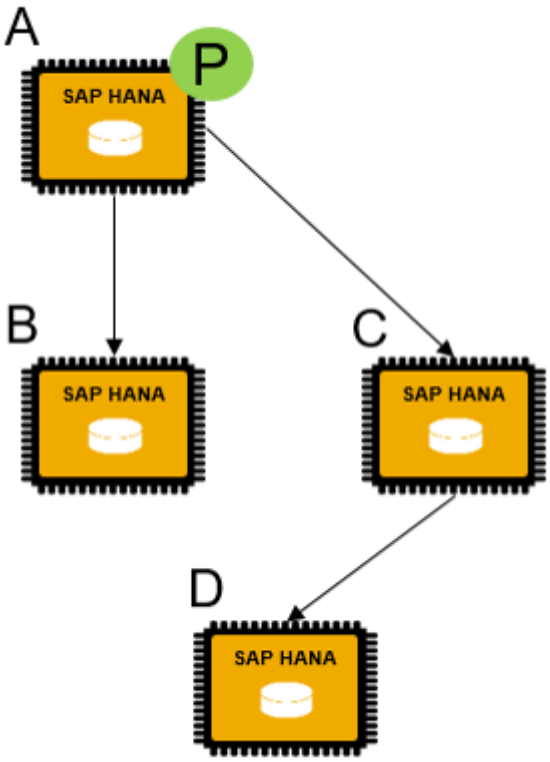
Performing a Takeover [page 815]
Performing a Failback [page 832]
Full Sync Option for SAP HANA System Replication [page 800]
Log Retention [page 781]
Automatic Registration After Takeover [page 825]

12.1.2.14.3 Automated Search for Alternative Source Sites

You can maintain a list of alternative source sites to which a secondary can be registered in case the preferred site is not available for some reason.

To use the option of considering alternative system replication sites to register to, set up a list of site names in the parameter `alternative_sources` (in the `[system_replication]` section of the global.ini file). In the

multi-target scenario illustrated here where site A is the main primary, this could be used, for example, for site D so that if a connection to site C is not possible then it will register to site B:



Set the value of the configuration parameter on the lower tier (in this case site D) giving both alternative source site names in a comma-separated list. This configuration parameter also supports the option of specifying the replication mode for each site by appending it with a colon:

```
alternative_sources=SiteC:ASYNC,SiteB:SYNCMEM
```

Note that this feature simply lists preferred sites, it does not trigger a takeover.

Two further parameters in the `[system_replication]` section of the global.ini file are used to determine firstly, the number of times the system will try to connect before switching to the alternative list (default value 20), and also the time interval between each attempt to connect (default value 30 seconds):

```
retries_before_register_to_alternative_source
reconnect_time_interval
```

12.1.2.15 Active/Active (Read Enabled)

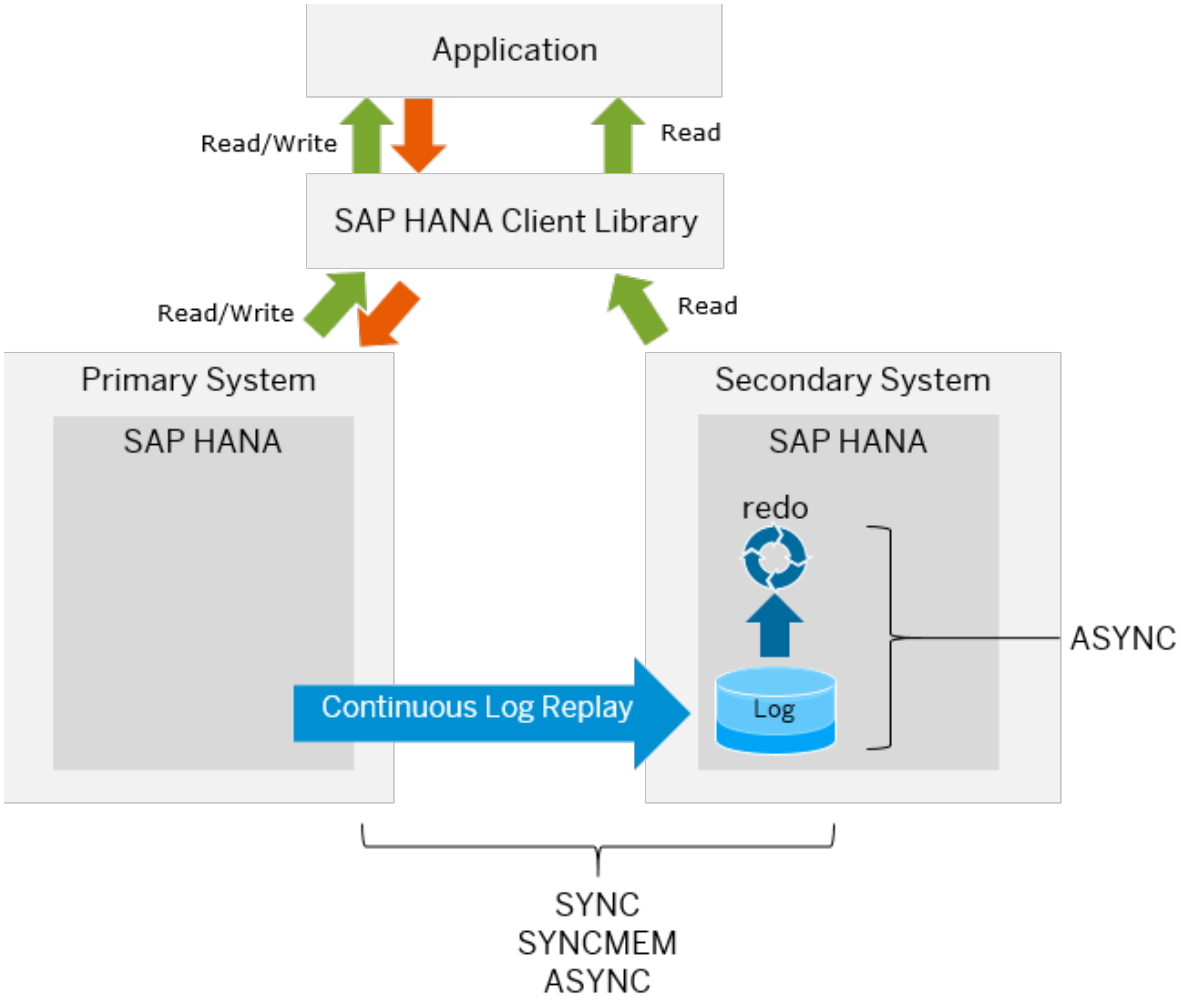
Active/Active (read enabled) enables SAP HANA system replication to support read access on the secondary system.

Active/Active (Read Enabled)

Active/Active (read enabled) reduces the load on the primary system but does not double the capacity; it simply extends read capabilities. In a replication system that is configured for Active/Active (read enabled), the SQL ports on the secondary system are open for read access. This makes it possible to use the secondary system for read-intense tasks and to have a better balance of workloads improving the overall performance of the SAP HANA database.

Active/Active (read enabled) is activated with the System Replication operation mode `logreplay_readaccess`, and uses the continuous logreplay mode. This operation mode provides fast takeovers, reduced need for bandwidth in continuous operation, and support for all replication modes: SYNC (with or without the full sync option), SYNCMEM, and ASYNC. For a detailed illustration of the general system replication processes, see *System Replication*.

The following graphic shows an Active/Active (read enabled) configuration: the primary system is fully active and supports reading and writing and the secondary system is enabled for read queries.



Operation Mode

To enable read access on the secondary system use the operation mode `logreplay_readaccess` when configuring system replication. Two options are supported to connect to the secondary as described in the topic 'Connection Types'.

To use this operation mode, the primary and the secondary systems must have the same SAP HANA version. For this reason, read-only access to the secondary system is not possible during a rolling upgrade until both versions are the same again.

In a multitier setup, read access on the secondary is only supported for tier 2. In a 3-tier system using Active/Active (read enabled), the `logreplay_readaccess` mode is required between the primary and the active secondary systems, while the logreplay mode is required between the other (tier 2 and tier 3) secondary systems.

In a multitarget setup, multiple secondary systems with read access (`logreplay_readaccess`) are supported; users can connect to any of the read-enabled secondaries at tier 2 to read data. In this case, if data will be read using hint based statement routing (using the RESULT_LAG hint), the server to which requests will be routed must be named in the configuration parameter `hint_based_routing_site_name` in

the system replication section of the global.ini file. Hint based routing is only possible to the secondary named in the parameter. To execute statements on any other read enabled secondary site users must connect directly to the site. Refer also to KBA 3122356 *How to designate the secondary site for hint based routing*.

> **Note**
>
> With the operation mode `logreplay_readaccess`, the secondary system allows read-only access on column tables via SQL providing a delayed view on the data compared to the primary system. There is no minimum delay guarantee. Read access on system and monitoring views is supported as well.

For more information about the operation mode `logreplay_readaccess`, see *Operation Modes for SAP HANA System Replication*. An SAP HANA Academy video on Active / Active (read enabled) is available on YouTube.

License Management

The SAP HANA Active/Active read-enabled option requires additional licensing (see also the *Feature Scope Description for SAP HANA*); the license for Active/Active (read enabled) also covers the multi-target replication scenario. Active/Active is enabled by setting the system replication operation mode to `logreplay_readaccess`, the secondary system is then operated automatically with the license key of the primary system. Changes to the license key are done on the primary system and replicated to the secondary system (see also *Managing SAP HANA Licenses*).

Related Information

Feature Scope Description for SAP HANA
Connection Types [page 856]
System Replication [page 760]
Configuring SAP HANA System Replication [page 792]
Operation Modes for SAP HANA System Replication [page 773]
Client Support for Active/Active (Read Enabled) - Client Interface Programming Reference
HINT Details (Hints for Active/Active (Read Enabled)) in the SAP HANA SQL and System Views Reference
Managing SAP HANA Licenses [page 137]
SAP Note 3122356
YouTube Channel SAP HANA Academy - System Replication: Active/Active Read Enabled

12.1.2.15.1 Generic Conditions for Active/Active (Read Enabled)

When using the secondary system for read access, several aspects need to be considered.

Points to Consider

- The processors in the primary and secondary systems must be both either Intel-based or IBM Power-based with the same byte ordering. A platform mixture is not supported.
- The secondary system allows read access if the primary system runs the same SAP HANA version (that is, the same build number, for example: 2.00.048.01.1593581573). A different version leads to prohibiting the read access until the same software version is used.
- The redo log replay runs as an asynchronous process on the secondary system. The secondary system provides statement level snapshot isolation with potentially delayed view on the data and no minimum delay guarantee.
- The secondary system gets its own virtual IP addresses or host names representing the secondary function.
- DML executions for table types that do not lead to redo log writing are possible on Active/Active (read enabled) secondary systems. This applies, for example, to global temporary tables, local temporary tables or row store no-logging retention tables. Explain Plan is also available on the secondary system. For more information, see *Data Manipulation Statements*.
- The query execution in the secondary system is rejected if it needs background migrations requiring redo log writes (for example, L2-Delta migration).

  > → Recommendation
  >
  > Perform the migration in the primary system (for example, load table) and wait until it is replicated and replayed in the secondary system.

- Internal processes for operations like Column Store delta merges take place on the secondary system.

Limitations

- Active/Active (read enabled) is supported in a multitier SAP HANA replication system. However, read access is limited to tier 2. The `logreplay` operation mode is required between tier 2 and the further tier level and no read access connections can be opened to the further tier level.
- If Active/Active (read enabled) is used with Dynamic Tiering services, there is no read access to Dynamic Tiering data on the secondary system.
- The export of tables is possible with CSV as target. However, binary exports on the secondary system are not supported.
- The use of workload classes is not supported on the secondary system.

Support for Multiple SAP HANA Databases

It is possible to use the read-enabled secondary system for other SAP HANA systems such as development or QA environments. In this case the following sizing conditions apply:

- The secondary hardware must offer the same CPU and memory capacities as those offered by the primary system **plus** the resources for the additional system.
- After a takeover, the system must be capable of handling both the primary's writing load and the secondary's reporting load.

For more information about this scenario, see also SAP Note 1681092 *Multiple SAP HANA DBMSs (SIDs) on one SAP HANA system*.

Related Information

Data Manipulation Statements
SAP Note 1681092
SAP Note 2447994

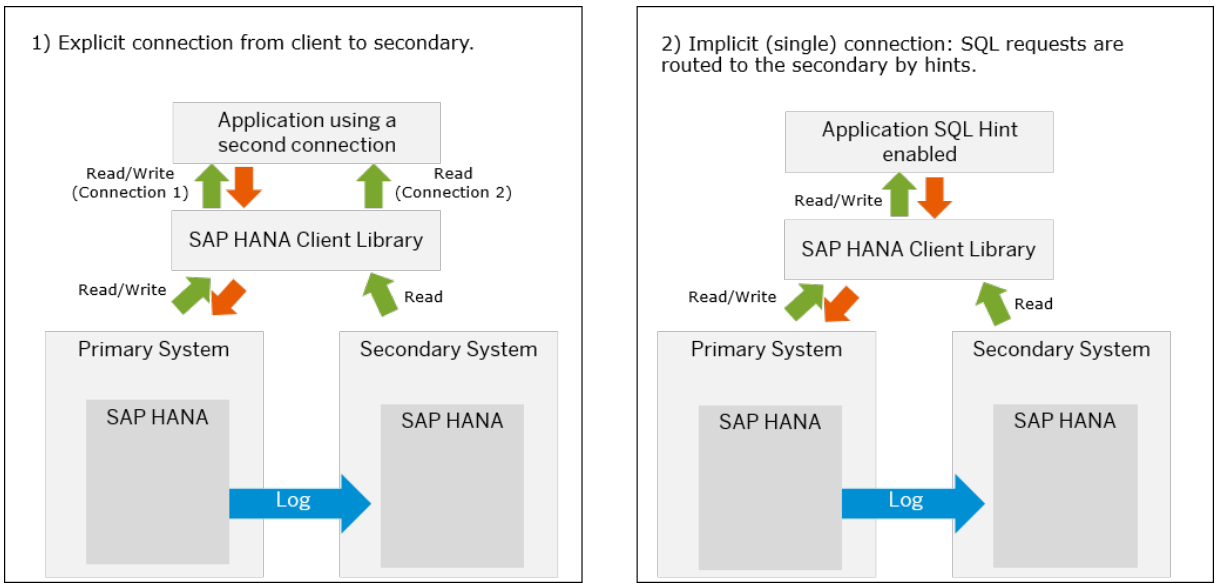
12.1.2.15.2 Connection Types

You can connect to an Active/Active (read enabled) secondary system either directly, or indirectly through the primary.

There are two ways to access the read-enabled secondary system:

1. Opening an explicit connection to the secondary system. In this case the application queries the read enabled secondary directly.
2. Executing an SQL statement on the primary system with a hint, the hint is evaluated and reroutes the query to the secondary system.

To provide access to the read enabled secondary it must have its own virtual IP address. The following illustration shows a comparison of these two methods:



When setting up Active/Active (read enabled) you should verify that the SQL ports on the secondary system are open and ready to take SQL commands. You can see the status of the port in the monitoring view M_SYSTEM_REPLICATION in column SECONDARY_READ_ACCESS_STATUS, the status should be 'Active'. In SAP HANA cockpit, the status of the port on the secondary is visible in the System Replication app on the primary; the status should show: 'Secondary Read Access: Enabled'.

For more details of the client connection refer to the *SAP HANA Client Interface Programming Reference for SAP HANA Platform* (link given below).

Hint-based Statement Routing

In this case the application connects to the primary system and the query execution unfolds as follows:

1. The SAP HANA client sends the statement-prepare with hint to the primary system.
2. The primary system decides where to execute the statement and returns the result to the SAP HANA client.
3. The SAP HANA client opens an additional connection to the secondary system according to the host information returned by the primary system. The client sends the statement execution call to the secondary system. The session property changes are handed over to the secondary system via the SAP HANA client. If the secondary system cannot execute the statement, it returns an error and the SAP HANA client sends the statement to the primary system.

For more details and an example see the following topic *Hint-Based Routing for Active/Active (Read Enabled)*. All available hints for SAP HANA are described in the section *HINT Details* in the *SAP HANA SQL and System Views Reference Guide*.

To cancel long-running sessions on a read enabled secondary system, use: `ALTER SYSTEM CANCEL SESSION`.

Limitations:

> **Note**
>
> Hint-based statement routing is supported by SAP HANA ODBC, SQLDBC, ADO.Net, JDBC drivers for SAP HANA 2.0 and the SAP HANA Node.js.

- The SAP HANA client has an open working connection to the primary system.
- Hint-based statement routing is not supported for CALL procedures.
- Temporary tables are not supported.
- Hint-based statement routing is not supported for write transactions.

Related Information

HINT Details (SAP HANA SQL and System Views Reference)
M_SYSTEM_REPLICATION System View (SAP HANA SQL and System Views Reference)
Hint-Based Statement Routing for Active/Active (Read Enabled) [page 858]
Client Support for Active/Active (Read Enabled) - Client Interface Programming Reference

12.1.2.15.3 Hint-Based Statement Routing for Active/Active (Read Enabled)

Connections to a primary system can use hint-based statement routing statement execution to a secondary system on a per-statement basis. This reduces the load on the primary system and increases overall performance.

To indicate that a statement should be hint-based routed to the secondary system, add the hint text `WITH HINT(RESULT_LAG('hana_sr'))` to the end of the SQL SELECT statement. For example:

```
SELECT C1, C2 FROM T1 WHERE C3 = 'constant value' WITH
HINT(RESULT_LAG('hana_sr'))
```

Queries that are executed directly (not prepared) are not hint-based routed even if they contain a hint. To take advantage of hint-based statement routing, there must be separate prepare and execute operations at the SQLDBC or JDBC level. In some cases, applications or interfaces that use SQLDBC or JDBC (such as SAP HANA Studio, SAP HANA Cockpit, ABAP, or PyDBAPI) can perform a separate prepare and execute without the user's knowledge.

Routing Conditions

In the following cases, the execution transparently falls back to the primary system, even if the statement contains a hint to route to the secondary system:

- The connection's isolation level is set to repeatable read or serializable.
- Connection to the secondary system is not possible (for example, there is a secondary system outage, a networking issue, and so on).
- The connection currently has a write transaction (uncommitted insert, update, or delete) in progress.
- The query references temporary tables.

If the routed connection is dropped while fetching from a hint-routed query result set, an error may be returned to the application.

> ⓘ Note
>
> Hint-based statement routing is only applied to SELECT statements.

Fallback Routing

In the following cases, a statement that has been routed to the secondary system gets re-routed to the primary system:

- The hint contains a maximum delay time parameter and the secondary system is delayed by more than that amount.
- The secondary system is near its maximum memory usage.
- The statement prepared in the primary system does not detect any access to the Dynamic Tiering data, but the statement execution in the secondary system requires Dynamic Tiering data.

Timeout

If the previous hint-based routed statement execution falls back to the primary system due to a connection or communication error, then future hint-based statement routing does not attempt to re-connect to the secondary system for several seconds. This avoids the performance cost of retrying the connection to the secondary system frequently when it is likely to fail.

In this case, the time between reconnection attempts to the secondary system is between five seconds and five minutes from the last reconnection attempt. The time between reconnection attempts automatically increases if reconnection attempts continue to fail.

In a multitier system replication system, hint-based statement routing always routes from the primary to the secondary system.

Related Information

HINT Details
Connection Types [page 856]

12.1.2.15.4 Configuration Parameters

Several parameters are available for configuring Active/Active (read enabled).

Parameter:	operation_mode
Values:	logreplay_readaccess
System:	Secondary
Description:	System Replication uses an initial data shipping to initialize the secondary system. After that, only log shipping is done and the log buffers received by the secondary system are being replayed there. Savepoints are executed individually for each service. Column table merges are executed on the secondary system. Additionally, read access via SQL is provided to the secondary system.

Relevant for Active/Active (read enabled) are also `enable_log_retention` and `logshipping_max_retention_size`. For more information about these parameters, see *SAP HANA System Replication Configuration Parameters*.

Related Information

SAP HANA System Replication Configuration Parameters [page 801]

12.1.2.15.5 Checking the Active/Active (Read Enabled) Configuration

You can check if your system replication is configured as an Active/Active (read enabled) system.

SAP HANA Cockpit

On the *System Replication* tile on the primary system, the `logreplay_readaccess` operation mode indicates that your system is an Active/Active (read enabled) system. Additionally, an enabled secondary read access informs you that the SQL ports are open for reading on the Active/Active (read enabled) secondary system.

On the system overview page of the secondary system, the *Mode: read-only* indicates that your system is an Active/Active (read enabled) system. Additionally, the *Delay* in ms is shown on top indicating how far behind is the consistent view on the data of this secondary system compared to the current data of the primary system.

For examples and more information, see *System Replication Tile* and *Example: Monitoring SAP HANA System Replication with SAP HANA Cockpit*.

SAP HANA Studio

On the primary system, select M_SYSTEM_REPLICATION from the monitoring view. To find out if your system is an Active/Active (read enabled) system, verify the columns *OPERATION_MODE* and *SECONDARY_READ_ACCESS_STATUS*.

Command Line

As `<sid>adm` run one of the following commands and then look for the operation mode `logreplay_readaccess`:

```
python
          $DIR_INSTANCE/exe/python_support/systemReplicationStatus.py --
sapcontrol=1 | grep OPERATION_MODE
               service/ld4144/30207/OPERATION_MODE=logreplay_readaccess
               service/ld4144/30201/OPERATION_MODE=logreplay_readaccess
               service/ld4144/30203/OPERATION_MODE=logreplay_readaccess
```

or

```
hdbnsutil -sr_state | grep "operation mode"
           operation mode: logreplay_readaccess
```

Related Information

System Replication Tile
Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit

12.1.2.15.6 Resource Management

A number of configuration parameters are available to set the resource limits for read accesses on the secondary system.

Use the parameters below to set a memory limit or thread limit for read accesses on the secondary system:

Parameter:	**sr_total_statement_memory_limit**
Type:	int (GB)
Default:	(empty)
Section	memorymanager
Description:	Memory limit in GB: • (empty): 50% of global allocation limit • 0: disable the feature • N: set the value as a limit The total statement memory is limited to 50 % of the global allocation limit, because 50% of the storage is reserved for log replay. Log replay should not fail because of memory limitations.

Parameter:	**sr_total_statement_thread_limit**
Type:	int (count)
Default:	0 (no limit)
Section	execution
Description:	Regulates the number of running job workers over all of running statements on the secondary. This makes it possible to control running threads on the secondary so that shipping and log replay can use the remaining CPU resources. • 0: disable the feature • N: set the maximum number of workers permitted

Parameter:	**sr_enable_tracking**
Type:	bool
Default:	on
Section	resource_tracking

Parameter:	sr_enable_tracking
Description:	Main switch for resource tracking used on the system replication secondary system.

Parameter:	sr_memory_tracking
Type:	bool
Default:	on
Section	resource_tracking
Description:	Enables or disables memory tracking on the secondary system.

12.1.2.15.7 Authentication Methods

There are several authentication methods supported for an Active/Active (read enabled) system replication.

The following authentication methods are supported for the primary system:

- Basic (User Name/Password)
- Kerberos
- SAML
- Session Cookies

The secondary system delegates the authentication phase to the primary system using the existing communication channel from the secondary system to the primary system. Remote authentication tickets or credentials are sent over the data centers.

12.1.2.15.8 Virtual IP Address Handling

In an Active/Active (read enabled) configuration, a second virtual IP address for the read access on the secondary system is needed.

Since in an Active/Active (read enabled) configuration both systems are open for SQL access, a second virtual IP address for read access on the secondary system is needed.

During takeover you can keep the virtual IP address of the secondary system. This virtual IP address will be used for read access until a reconnect occurs. The former virtual IP address of the primary system is also rebound to access the former secondary system, which is the now active system. In this situation, two virtual IP addresses are available for accessing the former secondary system after takeover. For more information, see *Client Connection Recovery After Takeover*.

> **Note**
>
> Make sure that the now active system is capable to handle the workload of the former primary system and the read-access secondary system.

During failback the system replication systems switch their roles and the virtual IP addresses switch their locations too.

Related Information

Client Connection Recovery After Takeover [page 817]
Failback
Connecting Using Active/Active (Read Enabled) (SAP HANA Platform)

12.1.2.15.9 Monitoring Active/Active (Read Enabled)

You can monitor the Active/Active (read enabled) solution using proxy views or the SAP HANA Cockpit.

Proxy views

The embedded statistics server runs in the primary system and collects data from the secondary system providing them in the corresponding proxy schema. For more information, see *Monitoring Secondary Systems*.

SAP HANA Cockpit

The monitoring functionality in the SAP HANA cockpit supports access to the read-enabled system. Additionally, it provides information about the delay of the currently available consistent view. For more information, see *Monitoring SAP HANA System Replication with the SAP HANA Cockpit*.

Related Information

Monitoring Secondary Systems [page 867]
Monitoring SAP HANA System Replication

12.1.2.16 Monitoring SAP HANA System Replication

You can monitor SAP HANA system replication with different tools.

You can monitor system replication using the following tools:

- SAP HANA cockpit
  For more information, see *Monitoring SAP HANA System Replication with the SAP HANA Cockpit* in the SAP HANA cockpit guide.
- SAP HANA studio
  For more information, see *Monitoring SAP HANA System Replication with the SAP HANA Studio* in the SAP HANA studio guide.
- hdbnsutil
  For more information, see *Monitoring SAP HANA System Replication with hdbnsutil*.

Related Information

Monitoring SAP HANA System Replication (SAP HANA Cockpit)
Monitoring SAP HANA System Replication (SAP HANA Studio)
Monitoring SAP HANA System Replication with hdbnsutil [page 873]
SAP HANA System Replication Details [page 875]
Monitoring and Replicating INI File Parameter Changes [page 871]
SAP HANA System Replication Alerts [page 865]
Monitoring Secondary Systems [page 867]

12.1.2.16.1 SAP HANA System Replication Alerts

This is an overview of all alerts issued for the primary system.

Refer to the 'Alerts Reference' topic in the *SAP HANA Troubleshooting and Performance Analysis Guide* for more details and recommended user actions to resolve alerts.

Alert ID 78	**Connection Closed**
Description	Alert 78 is raised when a system replication connection is closed.
Alert ID 79	**Configuration Parameter Mismatch**
Description	Alert 79 is raised when there is a configuration parameter mismatch between the primary and the secondary system.

Alert ID 94	Logreplay Backlog
Description	Alert 94 is raised when the system replication logreplay backlog increases. A delayed log replay on the secondary system causes a longer takeover time. The alert has a different priority based on the size of the redo log that was not yet replayed: • Low: 10 GB < logreplay backlog < 50 GB • Medium: 50 GB <= logreplay backlog < 500 GB • High: logreplay backlog >= 500 GB To identify the reason for the increased system replication logreplay backlog, check the state of the services on the secondary system. To get more information, monitor the secondary system. Possible causes for the increased system replication logreplay backlog can be, for example, a slow or not functioning log replay, or a non-running service on the secondary system.

Alert ID 104	Increased Log Shipping Backlog
Description	Alert 104 is raised when the system replication log shipping is delayed or does not work properly causing data loss on the secondary system, if a takeover is done. The alert has a different priority based on the threshold reached: • Low: 1 GB < log shipping backlog < 10 GB • Medium: 10 GB <= log shipping backlog < 50 GB • High: log shipping backlog >= 50 GB To identify the reason for the increased log shipping backlog, check the status of the secondary system. Possible causes for the increased backlog can be a slow network performance, connection problems, or other internal issues (for example, in SYNC or SYNCMEM replication modes). **Note** The calculated log shipping backlog size is the total size of the data. If you are using data compression for data and log files then the real size kept in the log buffer will be smaller. Compression of the log buffer tail, for example, is enabled by default see Data and Log Compression for details.

Alert ID 106	ASYNC Replication In-Memory Buffer Overflow
Description	Alert 106 is raised when the local in-memory buffer in the ASYNC replication mode is running full indicating possible network issues with the connection to the secondary system. The alert has a different priority based on the threshold reached: • Medium: if buffer runs full once within 24 hours • High: if buffer runs full more than once within 24 hours To identify the reason for the local in-memory buffer running full, check the buffer size, the network, the IO on the secondary system, or look for peak loads. The alert depends on the setting of the `logshipping_async_wait_on_buffer_full` parameter. For more information about this parameter, see *SAP HANA System Replication Configuration Parameters*.

Alert ID 107	Inconsistent fallback snapshot
Description	Alert 107 is raised for the primary and the secondary systems when there are broken fallback snapshots. For more information about fallback snapshots, see *Create a Fallback Snapshot*.

Alert ID 990	System Replication support issue in ESS
Description	Identifies issues while initializing system replication support in the Embedded Statistics Server (ESS).

> **Note**
>
> Before SAP HANA 1.0 SPS 09 there was one alert categorized as "Internal Event" (Alert 21) which covered alerts 78 and 79. Both situations were covered by one event type and could only be distinguished by the information text provided. Since SAP HANA 1.0 SPS 11 old style alerts based on alert 21 are not created anymore as a default.
>
> You can create them by setting the configuration parameter `keep_old_style_alert` to **`true`**. These alerts can be required to keep the existing monitoring infrastructure working. If activated, new alerts and old style alerts are created in parallel. Old alerting can be disabled by setting the `keep_old_style_alert` configuration parameter to **`false`** in global.ini file.
>
> The new alerts require that you have migrated to the embedded statistics server. For more information, see *SAP Note 1917938*.

Related Information

Monitoring Secondary Systems [page 867]
SAP HANA System Replication Configuration Parameters [page 801]
Create a Fallback Snapshot
Data and Log Compression [page 786]
Alerts Reference
SAP Note 1917938

12.1.2.16.2 Monitoring Secondary Systems

Remote SQL access on the primary system makes it possible to monitor and report on the secondary system statistics.

You can monitor the secondary system through proxy schemas and views. Proxy schemas and views are provided on the primary system, they extract information from the corresponding monitoring views on the secondary system. The retrieval of statistics is unaffected by the replication or operation mode and is available for a two system replication setup as well as for multitier landscapes.

Proxy Schemas

A new schema is created on the primary system for each registered secondary system. This schema follows the naming convention _SYS_SR_SITE_<siteName>, where <siteName> is the case-sensitive name given at the registration time of the secondary system. This schema contains a selected subset of monitoring views (for example, M_VOLUME_IO_TOTAL_STATISTICS), which proxies the statistics from the secondary system. These proxy views have the same column definitions as the equivalently named public synonyms already available for the primary system. When a secondary system is unregistered, the corresponding schema will be dropped. See the following topic 'Proxy Schemas' for how to locate proxy schemas using Database Explorer.

> **Note**
>
> If system replication is configured as an Active/Active (read enabled) system with the `logreplay_readaccess` operation mode, then more data is available from the secondary system in the _SYS_SR_SITE_<siteName> proxy schema and more monitoring views of the secondary system can be accessed using virtual tables. The query given below can be used to identify which collectors and alerts are related to Active/Active views.

SYS_DATABASES_SR_SITE <secondary_site_name> is a second proxy schema containing proxy views, which can be queried to get information from the corresponding view in the SYS_DATABASES schema on the secondary system. These proxy views simplify the monitoring of secondary systems from the primary system.

Alerts and Collectors

Based on these views and tables available in the proxy schema, the statistics server is able to generate alerts on the secondary systems of a system replication landscape. Alerts issued by the secondary systems are displayed in the *Alerts* tile of the SAP HANA cockpit.

The embedded statistics service uses event checkers to monitor the system and collectors to gather the required historical data from system views. For example, the collector COLLECTOR_GLOBAL_DISK_USAGE reads data from the system view M_DISK_USAGE.

When this feature is active an additional set of collectors is used for monitoring secondary systems. You can retrieve the full list of collectors by running the following query on the primary:

```
SELECT so.ID, so.NAME, so.TYPE, ss.EXTENSION_TYPE
FROM _SYS_STATISTICS.STATISTICS_OBJECTS so
LEFT JOIN _SYS_STATISTICS.STATISTICS_SCHEDULE ss ON so.ID = ss.ID
WHERE ss.EXTENSION_TYPE='SR' OR ss.EXTENSION_TYPE='AA'
ORDER BY ID;
```

The full list includes over seventy items the following lines show a sample of the output. The EXTENSION_TYPE value identifies Active/Active views (value 'AA'):

ID	NAME	TYPE	EXTENSION_TYPE
78	Alert_Replication_Connection_Closed	Alert	SR
89	Alert_Missing_Volume_File	Alert	SR

ID	NAME	TYPE	EXTENSION_TYPE
117	Alert_Mon_Column_Tables_Record_Count_Incl	Alert	AA
5001	Collector_Global_Rowstore_Tables_Size	Collector	AA
5002	Collector_Host_Service_Component_Memory	Collector	AA
5003	Collector_Host_Service_Memory	Collector	AA
5004	Collector_Host_Service_Statistics	Collector	AA
5005	Collector_Host_Resource_Utilization_Statistics	Collector	AA

The activity of the collection service for secondary systems is also monitored by the statistics service and an Alert is triggered if the collection service fails (this may occur for example when initializing a system which has been started or stopped). In this case the ESS continues to monitor the primary site (and will attempt to resume monitoring of the secondary), but intervention by the administrator may be necessary. Refer to the 'Alerts Reference' topic in the *SAP HANA Troubleshooting and Performance Analysis Guide* for more details of alerts and recommended user actions.

Alert 990	System Replication support issue in ESS
Description	Identifies issues while initializing System Replication support in the Embedded Statistics Server (ESS).
User Action	System replication support has been deactivated in ESS. Check the trace files for issues with System Replication.

The process of monitoring secondary systems can be disabled from all sites if required using the following settings:

On the system database:

```
ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini','SYSTEM')
SET ('statisticsserver','enable_system_replication_support') = 'no' with
reconfigure;
```

On a tenant database:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','SYSTEM')
SET ('statisticsserver','enable_system_replication_support') = 'no' with
reconfigure;
```

You can check whether the statistics service for system replication is enabled or not by running the following queries:

```
SELECT * FROM _SYS_STATISTICS.STATISTICS_PROPERTIES WHERE KEY =
'internal.views.SR';
SELECT * FROM _SYS_STATISTICS.STATISTICS_PROPERTIES WHERE KEY =
'internal.views.AA';
```

When the feature is enabled these select statements return the property details. Properties for Active/Active are only returned when the Active/Active feature is also enabled.

Limitations

- Monitoring view access is only possible if the primary and secondary systems run with exactly the same software version; that is, the same build number, for example: 2.00.048.01.1593581573.
- If you query such a proxy view when the secondary system is not started no results are returned but no SQL error is triggered.
- Querying against multitenant landscapes is limited to single tenant databases or the system database, meaning there are no views unifying all tenants on the system database similar to the SYS_DATABSES schema.

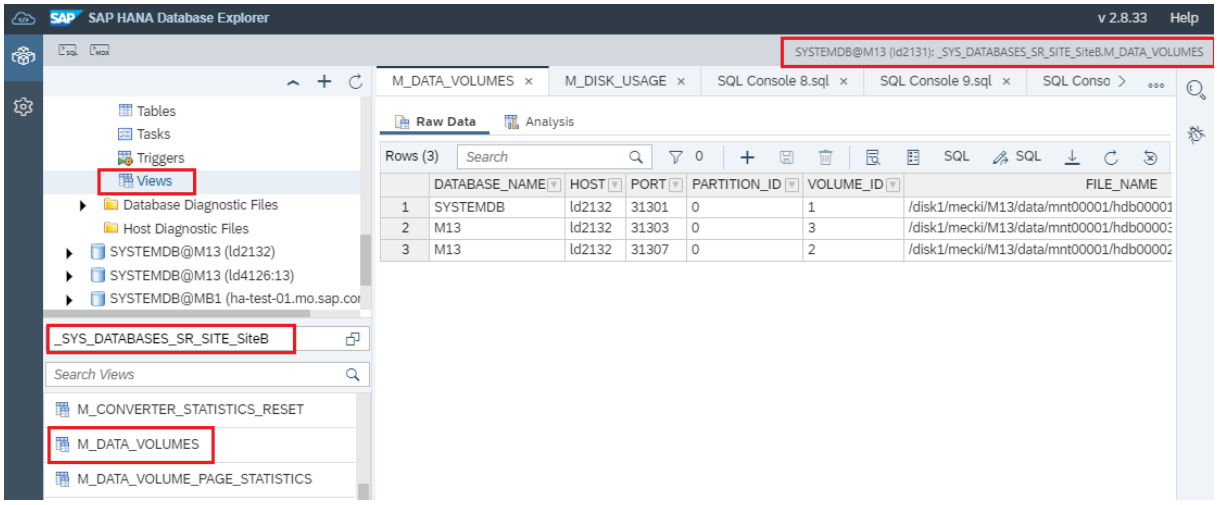
Related Information

Proxy Schemas and Views [page 870]
The Statistics Service [page 143]
Alerts Reference

12.1.2.16.2.1 Proxy Schemas and Views

You can monitor the secondary system through proxy schemas and views.

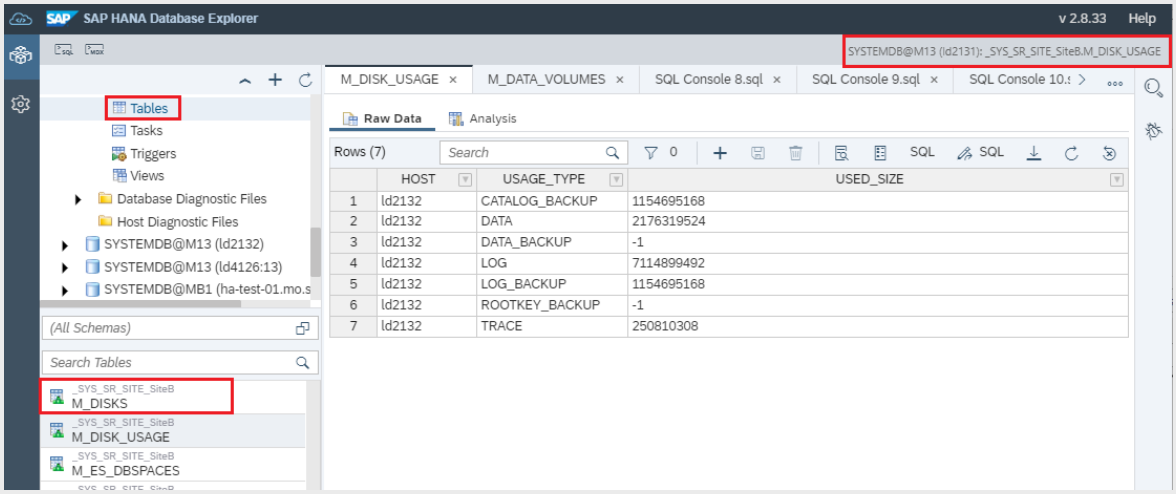
To see the proxy views of the secondary system's monitoring views, you can use the SAP HANA cockpit. On the system overview page of the primary system, choose *Execute SQL* to open the SAP HANA Database Explorer. Then open the primary's catalog and go to the corresponding schema. This is a schema example:



To see the virtual tables available in the proxy schema of the secondary system, open the proxy schema for the primary system in the SAP HANA Database Explorer for the primary system and choose *Tables*. A long list of accessible monitoring views from the secondary will be available.

Example

The virtual tables can look as follows:



Any of these proxy views or virtual tables can be accessed using SQL from the primary system by providing the correct secondary's schema name. For example:

```
select * from "_SYS_SR_SITE_SiteB"."M_HOST_INFORMATION";
```

12.1.2.16.3 Monitoring and Replicating INI File Parameter Changes

INI file parameters should be the same on each site of a system replication landscape and are checked automatically.

The configuration parameter checker reports any differences between primary, secondary, and further secondary systems.

Some parameters may have different settings on the primary and the secondary system on purpose. One example is the `global_allocation_limit` parameter where the secondary is used for other systems. By adding those parameters to the exclusion list below, they are excluded from checking and replication.

Activate ini parameter replication so that changes made on the primary are automatically replicated to the secondary systems. Otherwise, changes should be manually duplicated on the other systems.

The checks of the configuration parameter checker:

- Are done every hour by default.
- Generate alerts.
  The alerts are visible in the SAP HANA cockpit, SAP HANA studio, and the M_EVENTS system view.
  To view the alerts in the SAP HANA cockpit, look for the alerts tile on the system overview page of the SAP HANA cockpit and choose *Show All*. You can see all alerts including the ones created because of parameter mismatch.
- Are optimized for the most recently changed parameters.

Enable and disable the parameter check on the primary site with `[inifile_checker]/enable = true | false`. The parameter checker is on by default.

Enable and disable the parameter replication on the primary site with `[inifile_checker]/replicate = true | false`. The parameter replication is off by default.

The ini file parameter replication follows these rules:

Parameter set on		Activity
Primary System	**Secondary System**	
yes	no	Copy parameter to the secondary system.
no	yes	Delete parameter on the secondary system.
set to value x	set to value y	Copy value x to the secondary system.

The parameter changes on the secondary system are applied differently for each parameter:

- Online changeable parameters become active after the `ALTER SYSTEM` command or by editing the .ini file followed by the automatically triggered `hdbnsutil -reconfig` command.
- Offline changeable parameters become active after a restart. When changing such a parameter, it is necessary to restart the primary and secondary systems before a takeover. For more information about configuration parameters, see *Configuration Parameter Reference*.

To prevent parameters from generating alerts and eventually getting replicated, it is possible to create exclusions. In the following example, different global allocation limits (GAL) on primary and secondary systems can be set without being overwritten by the parameter replication.

Example

If for example you intend to use your secondary system for DEV/QA systems and set the global allocation limit to its minimal value (as described above), you may exclude this `global_allocation_limit` parameter from these checks as follows:

```
[inifile_checker]
enable = true|false
interval = 3600
exclusion_global.ini/SYSTEM = memorymanager/global_allocation_limit
```

The exclusion rules are written in the following syntax (comma separated list) and take effect immediately:

```
exclusion_[inifile name|*][/<LAYER>] = [section with
wildcards|*][/parameter with wildcards|*], ...
<LAYER> := SYSTEM\|HOST\|DATABASE\|\*
```

Related Information

Configuring SAP HANA System Properties (INI Files) [page 126]

SAP HANA Configuration Parameter Reference

12.1.2.16.4 Monitoring SAP HANA System Replication with hdbnsutil

You can monitor SAP HANA system replication using `hdbnsutil`.

Standard System Replication

To view the status of the system replication topology configuration on both systems, execute `hdbnsutil –sr_state` on the primary and the secondary:

```
tedadm@ld2131:/usr/sap/TED/HDB07> hdbnsutil -sr_state
checking for active or inactive nameserver ...
System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~
mode: primary
site id: 1
site name: SITEA
Host Mappings:
~~~~~~~~~~~~~~
ld2131 ->
[SITEA] ld2131
ld2131 ->
[SITEB] ld2132
done.
```

Multitier System Replication

For a multitier system replication the mappings of all three systems are displayed:

```
ut1adm@ld2131:/usr/sap/UT1/HDB01> hdbnsutil -sr_state
checking for active or inactive nameserver ...
System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~
mode: primary
site id: 1
site name: SITEA
Host Mappings:
~~~~~~~~~~~~~~
ld2131 ->[SITEA] ld2131
ld2131 ->[SITEC] ld2133
ld2131 ->[SITEB] ld2132
done.
```

When using the additional option `–-sapcontrol=1`, the key value pair output can be parsed by a script line by line.

Here is the output where the `-sr_state` command was executed on a primary system of a multitier system replication:

```
ut1adm@ld2131:/usr/sap/UT1/HDB01> hdbnsutil -sr_state --sapcontrol=1
checking for active or inactive nameserver ...
SAPCONTROL-OK: <begin>
mode=primary
```

```
site id=1
site name=SITEA
mapping/ld2131=SITEA/ld2131
mapping/ld2131=SITEC/ld2133
mapping/ld2131=SITEB/ld2132
SAPCONTROL-OK: <end>
Done
```

Here is the output where the `-sr_state` command was executed on a tier 2 secondary site of a multitier system replication:

```
ut1adm@ld2132:/usr/sap/UT1/HDB01> hdbnsutil -sr_state --sapcontrol=1
checking for active or inactive nameserver ...
SAPCONTROL-OK: <begin>
mode=sync
site id=2
site name=SITEB
active primary site=1
mapping/ld2132=SITEA/ld2131
mapping/ld2132=SiteC/ld2133
mapping/ld2132=SITEB/ld2132
primary masters=ld2131
SAPCONTROL-OK: <end>
done.
```

Output Reference

Output	Description
Mode	Can have the values `primary`, `sync`, `async`, and `syncmem` to represent the mode relevant on the system where the command is executed. For example, in a multitier system replication on the primary the mode would be `primary`, on the tier 2 secondary it could be either `sync` or `syncmem`, and on the tier 3 secondary it is `async`.
Site ID	A unique identifier of a system which is incremented for each system attached to a SAP HANA system replication. It is removed, when system replication is disabled.
Site Name	The name you give your systems during the enable and register steps of the system replication configuration.
Mapping/<currentHost>	Shows which hosts are involved in this SAP HANA system replication together with their system name. If the SAP HANA database is offline, this host mapping cannot be shown on the secondaries.
Active primary site	Shows the system ID of the currently active system.
Primary masters	Shows the host names of the currently active master candidates of the primary.

> **Note**
>
> When running `hdbnsutil –sr_state` on an offline SAP HANA, no host mapping will be available. For more information, see *SAP Note 2315257*.

Related Information

SAP Note 2315257

12.1.2.16.5 SAP HANA System Replication Details

Detailed information from the M_SERVICE_REPLICATION and the M_SYSTEM_REPLICATION monitoring views about system replication.

General Overview

Column	Description
Site ID 1	Generated ID of the primary site
Secondary Site ID 2	Generated ID of the secondary site
Service	Name of the service
Volume ID	Persistence volume ID
Operation Mode	• LOGREPLAY • LOGREPLAY_READACCESS • DELTA DATA SHIPPING
Replication Mode	Configured replication mode: • SYNC: synchronous replication with acknowledgment when buffer has been written to disk on the secondary system • SYNCMEM: synchronous replication with acknowledgment when buffer arrived in memory on the secondary system • ASYNC: asynchronous replication where the primary doesn't wait for the acknowledgment • UNKNOWN: is set if replication mode can't be determined (for example, if there are communication errors when getting status information from a service).
Replication Status	Current status of replication: • UNKNOWN: secondary didn't connect to primary since last restart of the primary • INITIALIZING: initial data transfer is running. In this state, the secondary is first usable when this is finished • SYNCING: secondary is syncing again (for example, after a temporary connection loss or restart of the secondary) • ACTIVE: initialization or sync with primary is complete and secondary is continuously replicating. If crash occurs, no data loss occurs in SYNC replication mode. • ERROR: replication can't take place because the secondary system isn't accessible (details can be found in Replication Details)

Column	Description
Replication Details	Additional information for Replication Status, for example, the error text if status is ERROR.
Full Sync	Indicates if the service is currently operating in sync replication mode with the full sync option set. If full sync is enabled in a running system, full sync might not be active immediately. This is done to prevent the system from blocking transactions immediately when setting the parameter to true. Instead, in a first step, full sync has to be enabled. In a second step, it's internally activated when the secondary is connected and becomes ACTIVE. • DISABLED: full sync isn't configured at all • ENABLED: full sync is configured, but it isn't yet active, so transactions don't block in this state. To become active, the secondary has to connect and Replication Status has to be ACTIVE. • ACTIVE: full sync mode is configured and active. If a connection of a connected secondary is getting closed, transactions on the primary side block in this state. If full sync is enabled when an active secondary is currently connected, the FULL_SYNC is immediately set to ACTIVE.
Secondary Fully Recoverable	TRUE: No full data backup is needed after takeover on secondary. Backups created on the primary and local log segments enable a full database recovery. FALSE: Log segments needed for a full database recovery are missing. After takeover, a full data backup has to be executed before a full recovery up to the most recent time point can be executed.
Secondary Active	Status of the secondary node (also see ACTIVE_STATUS in M_SERVICES)
Secondary Connect Time	Timestamp the secondary connected to the primary. If there are reconnects from the secondary side, this field contains the last connect time.
Secondary Reconnect Count	Number of reconnects from secondary side for this service.
Secondary Failover Count	Number of failovers for this service on secondary side.
Buffer Full count	Number of times, the asynchronous replication buffer was full since last service restart (only relevant for replication mode async; 0 for replication modes sync/syncmem).

Log Positions

Column	Description
Last Log Position	Last known log position on primary
Last Log Position Time	Timestamp of last known log position

Column	Description
Replayed Log Position	Log end position of the last known replayed log buffer on secondary site
Replayed Log Position Time	Timestamp of the last known replayed log buffer on the secondary site
Last Shipped Log Position Time	Timestamp of last log position being shipped to secondary
Shipped Log Buffer Count	Number of log buffers shipped to secondary
Shipped Log Buffers Total Size (Bytes)	Size of all log buffers shipped to secondary
Shipped Log Buffers Total Time (µs)	Time taken to ship all the log buffers to the secondary. • SYNC/SYNCMEM: total round-trip time to send the log buffers and receive the acknowledgment. • ASYNC: start time when sending the log buffers, and end time when the OS reports that the log buffers were sent (and the log shipping buffer space was freed). This time could be shorter than the SYNC/SYNCMEM duration
Time delay (ms)	Time delay between the last shipped log position time and the replayed log position time on the secondary
Size delay (Bytes)	Size delay between the last shipped log position size and the replayed log position size on the secondary (1 log position = 64 bytes)

Savepoints

Column	Description
Last Savepoint Version	Last savepoint version on primary
Last Savepoint Log Position	Log position of current savepoint
Last Savepoint Start Time	Timestamp of current savepoint
Last Shipped Savepoint Version	Last savepoint version shipped to secondary
Last Shipped Savepoint Log Position	Log position of last shipped savepoint
Last Shipped Savepoint Time	Timestamp of last shipped savepoint

Full Data Replica

Column	Description
Full Data Replica Shipped Count	Number of full data replicas shipped to secondary
Full Data Replica Shipped Total Size (Bytes)	Total size of all full data replica shipped to secondary
Full Data Replica Shipping Total Time (µs)	Duration for shipping all full data replica
Last Full Data Replica Shipped Size (Bytes)	Size of last full data replica shipped to secondary
Start Time of Last Full Data Replica	Start time of last full data replica
End Time of Last Full Data Replica	End time of last full data replica

Delta Data Replica

This information is only displayed if the operation mode is `delta_datashipping`.

Column	Description
Delta Data Replica Shipped Count	Number of delta data replicas shipped to secondary
Delta Data Replica Shipped Total Size (Bytes)	Total size of all delta data replicas shipped to secondary
Delta Data Replica Shipped Total Time (µs)	Duration for shipping of all delta data replicas
Size of Last Delta Data Replica (Bytes)	Size of last delta data replica
Start Time of Last Delta Data Replica	Start time of last data delta replica
End Time of Last Delta Data Replica	End time of last data delta replica

Log Shipping Backlog

Column	Description
Current Replication Backlog Size (Bytes)	Current replication backlog in bytes. The size of all log buffers that have been created on primary site, but not yet sent to the secondary site. Even in replication modes sync/syncmem this column can have a value different from 0. Here it represents the size of log buffers that are in the local send queue (max number of those buffers is the number configured log buffers on primary site).
Max Replication Backlog Size (Bytes)	Max replication backlog in bytes (max value of BACKLOG_SIZE since system start).
Current Replication Backlog Time (µs)	Current replication backlog in microseconds. This time is the difference between time of the last sent log buffer and the current log buffer. Even in replication modes sync/syncmem this column can have a value different from 0, because log buffers are still in the send queue (max number of these buffers is the number of log buffers configured on primary site).
Max Replication Backlog Time (µs)	Max replication backlog in microseconds (max value of BACKLOG_TIME since system startup).

Log Replay

This information is only displayed if the operation mode is `logreplay` and `logreplay_readaccess`.

Column	Description
Replay Backlog Size (Bytes)	Specifies the size of all log buffers that have been shipped to the secondary site but haven't yet been replayed on the secondary site.
Max Replay Backlog Size (Bytes)	Specifies the maximum value of the REPLAY_BACKLOG_SIZE since the system startup.
Replay Backlog Time (µs)	Specifies the time difference between the time of the last shipped log buffer and the last replayed log buffer on the secondary site.

Column	Description
Max Replay Backlog Time (µs)	Specifies the maximum value of REPLAY_BACKLOG_TIME since the system startup.

Related Information

SQL and System View Reference

12.1.2.17 Host Name Resolution for System Replication

The correct mapping of internal host names between primary and secondary systems is required for system replication.

With SAP HANA system replication, each SAP HANA instance communicates on the service level with a corresponding peer in the secondary system to persist the same data and logs as in the primary system. The replication of the transactional load can be configured to work in synchronous or asynchronous mode, depending mainly on the distance between the two sites.

Communication between the primary and the secondary system is based on internal ('virtual') host names. The host names of the other site must always be resolvable, either through configuration in SAP HANA or corresponding entries in the `/etc/hosts` file.

For system replication it is not necessary to edit the `/etc/hosts` file, internal host names must be mapped to IP addresses in the `global.ini` file to create a dedicated network for system replication. This is necessary to ensure that each site can resolve the host name of other replicating sites and that hosts can be switched seamlessly in the event of a takeover. These virtual host names must be set before registering the secondary system because the `-sr_register` command uses this mapping. Mappings are maintained in the `[system_replication_hostname_resolution]` section:

```
global.ini
[system_replication_hostname_resolution]
<ip-address_site>=<internal-host-name_site>
<...>
```

Virtual host names can also be used if the hostnames have a domain suffix, for example: the internal hostnames can be defined as ab820* and ab830*, but the public names have to include the domain, such as ab820*.abc.xyz.com and ab830*.def.xyz.com

> ⚠ Caution
>
> Note that by default no mappings are specified and system replication communication uses the default network route (typically the public network). If you use a public network instead of a separate network, you **must** secure this connection with additional measures such as a firewall or a virtual private network and/or TLS/SSL.

The mappings are used in combination with the `listeninterface` parameter in the `[system_replication_communication]` section which in a replication scenario can be set to

either `.global` or `.internal`. The following table shows two settings of the `listeninterface` parameter and the scope of the corresponding mapped host names. These are both illustrated in the following graphics showing a multi-node replication environment with a separate internal network for replication.

Another `listeninterface` parameter in the `[communication]` section is used for the communication between SAP HANA services (name server, index server, and so on) in a distributed system, but it has no impact on system replication.

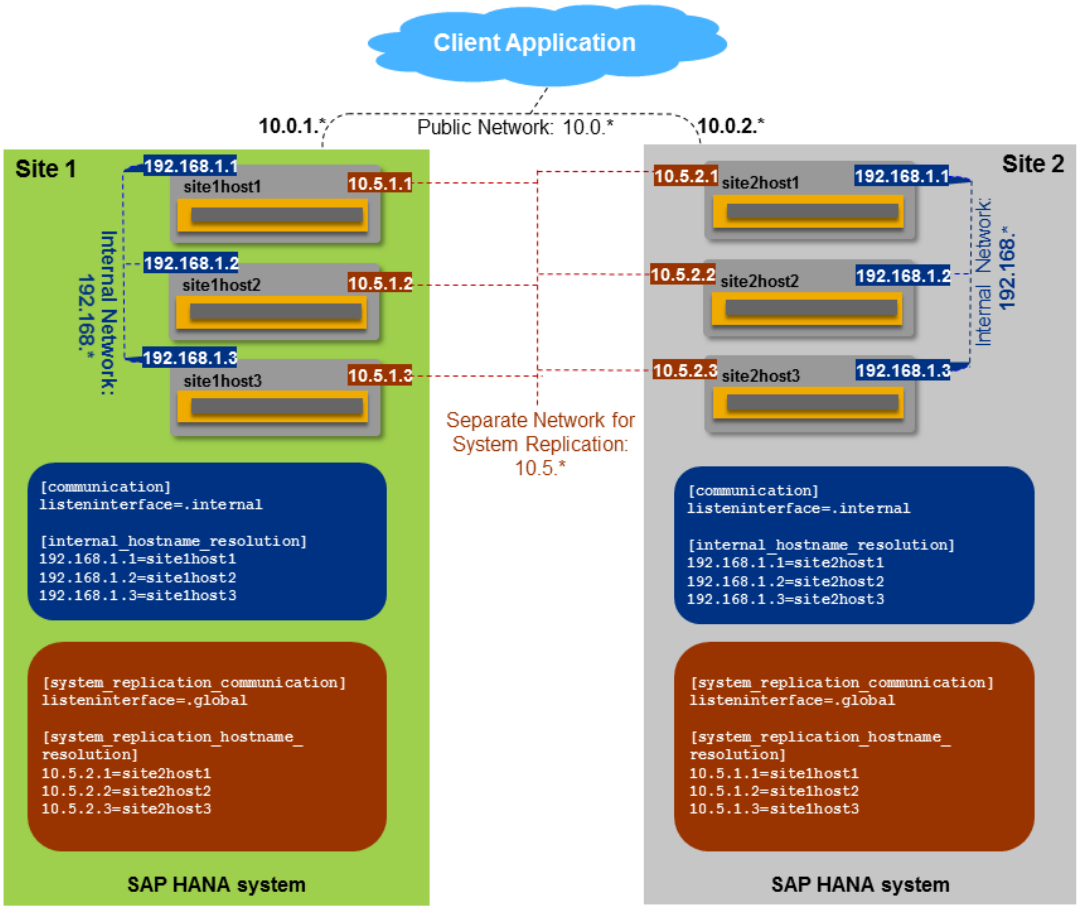
listeninterface	Host Name Resolution Mappings	Additional Information
`.global`	IP addresses and host names of neighboring sites (minimum) or for all hosts of own site as well as for all hosts of neighboring sites. Also applies to multi-tier setups.	This establishes a separate network for system replication communication. → Tip Also for multitier and multitarget setups, this is how you can use a dedicated network for system replication communication.
`.internal`	Entries for all hosts of own site as well as for all hosts of neighboring sites	A separate network is used for system replication communication. The primary hosts listen on the dedicated ports of the separate network only, and incoming requests on the public interfaces are rejected. ⚠ Caution As of SAP HANA 1.0 SPS 11, network communication for system replication with `listeninterface=.internal` is supported for two-tier replication but **not for multitier and multitarget setups**.

Examples

The following examples show the host name resolution configuration for system replication to a secondary site. Three distinct networks can be identified:

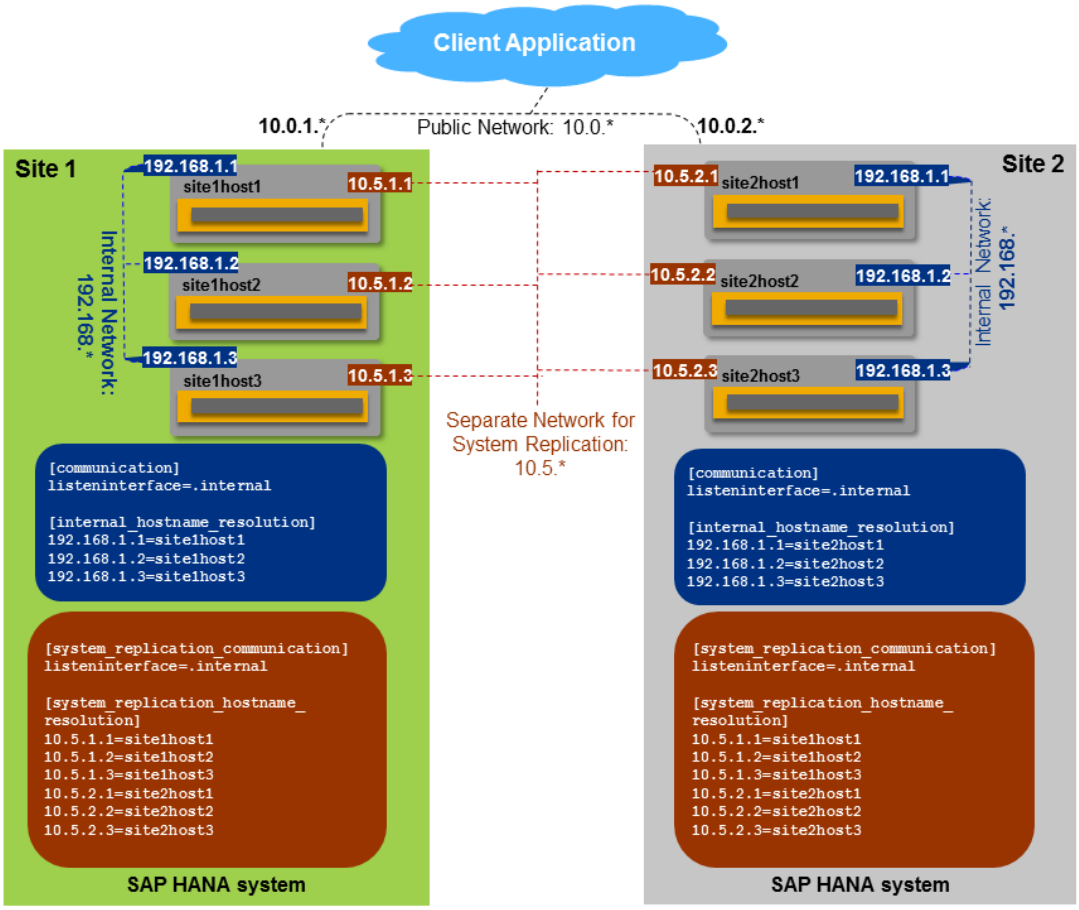
- Public network with addresses in the range of 10.0.1.*
- Network for internal SAP HANA communication between hosts at each site: 192.168.1.*
- Dedicated network for system replication: 10.5.1.*

In the first example, the `[system_replication_communication]listeninterface` parameter has been set to `.global` and only the hosts of the neighboring replicating site are specified.



Global Network

In the following example, the `[system_replication_communication]listeninterface` parameter has been set to `.internal` and all hosts (of both sites) are specified.



Internal Network

Important Security Information

If you use a public network instead of a separate internal network and/or the parameter `listeninterface=.global`, you **must** secure this connection with additional measures such as a firewall or a virtual private network and/or TLS/SSL.

If no separate internal network channel is configured for SAP HANA system replication, the `allowed_sender` parameter can be used to restrict communication between the primary and secondary sites to certain hosts. For this purpose, the following settings are made in the `global.ini` file on the primary site:

```
global.ini
[system_replication_communication]
allowed_sender=<comma-separated list of IP-addresses of secondary or CIDR-
netmasks>
```

An example of this parameter value would be **`10.0.1.0/30`**. The default is no restriction.

Further Information

For more security-related information, see the *SAP HANA Security Guide* and *SAP HANA Security Checklists and Recommendations*.

An SAP Community blog *How to configure HANA network communication channels – Part 2: Internal network* gives further details of these features.

The original white paper describing system replication provides further background information: *Introduction to High Availability for SAP HANA*.

Related Information

System Replication [page 760]
Configure SAP HANA System Replication with hdbnsutil [page 793]
Secure Internal Communication Between Sites in System Replication Scenarios (Security Guide)
Recommendations for Network Configuration (Security Checklists)
White paper "Introduction to High Availability for SAP HANA"
How-To Guide: How To Configure Network Settings for HANA System Replication
SAP Community Blog: How to configure HANA network communication channels – Part 2: Internal network

12.1.2.18 Updating SAP HANA Systems with SAP HANA System Replication

You can update your SAP HANA systems running in a SAP HANA system replication.

If for some reason you have to stop and restart the primary or the secondary, once the systems are available they will automatically try to get in sync again. There are no manual steps necessary.

If the system is running with `logreplay` or `logreplay_readaccess`, see *Resync Optimization* to prevent your system from running full or having to do a full data shipping. This can happen when the time during which the primary could not replicate gets too long. An optimized resync with delta data or log shipping can be achieved avoiding a full data shipping, depending on the the time the upgrade is taking and the settings of the `logshipping_max_retention_size` and `datashipping_snapshot_max_retention_time` parameters.

You must update your SAP HANA systems running in a system replication setup by updating the secondary system first and then updating the primary system. For more information, see *Update an SAP HANA System Running in a System Replication Setup*.

The secondary system can run with a higher software version than the primary system. For more information, see *Use SAP HANA System Replication for Near Zero Downtime Upgrades*.

> **Note**
>
> System Replication with SAP HANA 2.0 requires authentication for data and log shipping channels. The authentication is done using the certificates in the system PKI SSFS store. Thus, there is an additional manual setup step required to exchange certificates in the system PKI SSFS store between the primary and the secondary system when upgrading from SAP HANA 1.0 to SAP HANA 2.0. For more information, see *SAP Note 2369981*.

Hardware can also be exchanged with a minimal downtime using SAP HANA system replication. For more information, see *SAP Note 1984882: Using HANA system replication for Hardware Exchange with Minimum Downtime*.

Related Information

Resync Optimization [page 779]
Configure a User Under the SRTAKEOVER Key [page 884]
Update SAP HANA Systems Running in a System Replication Setup [page 893]
Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 885]
Use Multitarget System Replication for Near Zero Downtime Upgrades [page 888]
SAP Note 2369981
SAP Note 1984882

12.1.2.18.1 Configure a User Under the SRTAKEOVER Key

In preparation for maintenance tasks, you must configure a user in the local userstore under the SRTAKEOVER key.

Context

The SRTAKEOVER user requires the necessary privileges to import the repository content of the new version of the software during the takeover process.

Procedure

1. As `<sid>adm` configure a user in the local userstore under the SRTAKEOVER key. Use a public host name to access the corresponding SQL port of the System DB (`<SystemDBsqlport>`). Execute this command on the primary and secondary systems:

   ```
   hdbuserstore SET SRTAKEOVER <publichostname>:<SystemDBsqlport> <myrepouser>
   <myrepouser_password>
   ```

   > **Note**
   >
   > This configuration step should be performed only in the system database, not in every single tenant.

2. Create a `<myrepouser>` user with the necessary privileges to import the repository content as follows:

   ```
   CREATE USER MY_REPO_IMPORT_USER PASSWORD MyRepoUserPW123;
    GRANT EXECUTE ON SYS.REPOSITORY_REST TO MY_REPO_IMPORT_USER;
   ```

```
 GRANT REPO.READ ON ".REPO_PACKAGE_ROOT" TO MY_REPO_IMPORT_USER;
 GRANT REPO.IMPORT TO MY_REPO_IMPORT_USER;
 GRANT SELECT ON _SYS_REPO.DELIVERY_UNITS TO MY_REPO_IMPORT_USER;
 GRANT REPO.ACTIVATE_IMPORTED_OBJECTS ON ".REPO_PACKAGE_ROOT" TO
MY_REPO_IMPORT_USER;
```

For example, for public host name "mypublichost" and system number "00", "MY_REPO_IMPORT_USER", and "MyRepoUserPW123":

```
hdbuserstore SET SRTAKEOVER mypublichost:30013 MY_REPO_IMPORT_USER
MyRepoUserPW123 NO FORCE_FIRST_PASSWORD_CHANGE
```

The host name has to be the public host name of the host on which the command is executed and the port is the SQL port number of the system database. Include NO FORCE_FIRST_PASSWORD_CHANGE so that the user is not immediately required to change the password.

For more information, see the *Use the User Store (hdbuserstore)* section in the *SAP HANA Client Interface Programming Reference*.

> **Note**
>
> In a scale-out configuration, the command has to be executed on all hosts. If the password for the repository import user is changed, the password saved in the userstore also has to be changed.

Related Information

Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 885]
Use the User Store (hdbuserstore)

12.1.2.18.2 Use SAP HANA System Replication for Near Zero Downtime Upgrades

You can use SAP HANA system replication to upgrade your SAP HANA systems as the secondary system can run with a higher software version than the primary system.

Prerequisites

You configured a user in the local userstore under the SRTAKEOVER key. For more information, see *Configure a User Under the SRTAKEOVER Key*.

System replication is configured and active between two identical SAP HANA systems:

- The primary system is the production system.
- The secondary system will become the production system after the upgrade.

Context

With system replication active, you can first upgrade the secondary system to a new revision and have it take over in the role of primary system. The takeover is carried out in only a few minutes and committed transactions or data are not lost. You can then do an upgrade on the primary system, which is now in the role of secondary.

> ⓘ Note
>
> It is possible to reduce the time required to perform an update. For more information, see *Prepare an Update for Flexible System Downtime* in the *SAP HANA Server Installation and Update Guide*.

The secondary system can be initially installed with the new software version or upgraded to the new software version when the replication has already been configured. After the secondary has been upgraded, all data has to be replicated to the secondary system (already having the new software version). When the secondary system is ACTIVE (all services have synced), a takeover has to be executed on the secondary system. This step makes the secondary system the production system running with the new software version.

If the installed system version on the primary is HANA 2.0 SPS 04 or greater then you are recommended to use the 'takeover with handshake' option to ensure a consistent handover. Using this option the primary continues to run but the writing of transactions on the primary system is suspended. The takeover is only executed when all redo log is available on the secondary system. See step 3 in the following procedure.

If you are upgrading from SAP HANA 1.0 to SAP HANA 2.0 note that system replication with SAP HANA 2.0 requires authentication for data and log shipping channels, this is done using the certificates in the system PKI SSFS store. You must therefore copy the system PKI SSFS key and the data file from the current primary system to the new to-be secondary system. Copy the files before registration when the secondary system is offline; the files can be found here:

```
/usr/sap/<SID>/SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT
/usr/sap/<SID>/SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY
```

For more information, see *SAP Note 2369981: Required configuration steps for authentication with HANA System Replication*.

In an Active/Active (read enabled) system replication setup, the version of the primary and the secondary systems must be identical. For the near zero downtime upgrade to work, the operation mode on the secondary system is automatically set to `logreplay`. Like this, the two systems can get back in sync before the takeover step. To establish again the Active/Active (read enabled) landscape at the end, the `logreplay_readaccess` operation mode must be explicitly specified during the former registration of the primary system as a new secondary system.

For more information about near zero downtime upgrades when using a multitarget system replication setup, see *Use Multitarget System Replication for Near Zero Downtime Upgrades*.

Procedure

1. Upgrade the secondary system's SAP HANA server software and all other components.

From your installation directory execute as root:

```
./hdblcm --action=update
```

2. Verify that system replication is active and that all services are in sync.

   You can check that the column REPLICATION_STATUS in M_SERVICE_REPLICATION has the value ACTIVE for all services.

3. Depending on the version installed on the primary, perform a takeover by doing one of the following:

   - If the installed system version on the primary is HANA 2.0 SPS 04 or greater then you have the option to use the `--suspendPrimary` parameter for a 'takeover with handshake' which ensures that all redo logs are written to disk. In this case, execute the takeover as `<sid>adm` with the following command:

     ```
     hdbnsutil -sr_takeover --suspendPrimary
     ```

   - If the installed system version on the primary is less than HANA 2.0 SPS 04, then:
     - Stop the primary system.
     - Execute the takeover as `<sid>adm` with the following (default) command:

       ```
       hdbnsutil -sr_takeover
       ```

   You can then switch virtual IP addresses to the secondary system, and start using it productively.

4. If XS Advanced is being updated as well, update the XS Advanced applications.

   ```
   ./hdblcm --action=update
   ```

5. If the primary has not been stopped (takeover with handshake option), you can now stop the primary system.

6. Upgrade the original primary from the installation directory as root user using the 'nostart' option. This option is required because otherwise the primary has to be stopped again before it can be registered as the secondary:

   ```
   ./hdblcm --action=update --hdbupd_server_nostart
   ```

   > **Note**
   >
   > For a fast synchronization of the sites – after registering again the original primary system – perform this failback within the time given by the `datashipping_snapshot_max_retention_time` parameter (default 300 minutes), otherwise, a full data shipping will be done. Furthermore, the optimized resync depends on the availability of the last snapshot.
   >
   > For more information about near zero downtime upgrades in multitier system replication, see *SAP Note 2386973*.

7. Register the original primary as secondary as `<sid>adm`.

   ```
   hdbnsutil -sr_register --name=<secondary_alias>
    --remoteHost=<primary_host> --remoteInstance=<primary_systemnr>
    --replicationMode=[sync|syncmem|async] --operationMode=[delta_datashipping|
   logreplay|logreplay_readaccess]
   ```

8. Start the original primary.

Related Information

Configure a User Under the SRTAKEOVER Key [page 884]
Prepare an Update for Flexible System Downtime
Updating the SAP HANA System
Perform a Near-Zero Downtime Update
Use Multitarget System Replication for Near Zero Downtime Upgrades [page 888]
Deploy a Multi-Target Application with Zero-Downtime Maintenance
Takeover with Handshake [page 824]
SAP Note 2369981
SAP Note 1984882
SAP Note 2386973
SAP Note 2494079
SAP Note 2407186
SAP Note 2300936

12.1.2.18.2.1 Use Multitarget System Replication for Near Zero Downtime Upgrades

You can upgrade your SAP HANA systems running in a multitarget system replication setup.

Prerequisites

Multitarget system replication is configured and active between identical SAP HANA systems.

Context

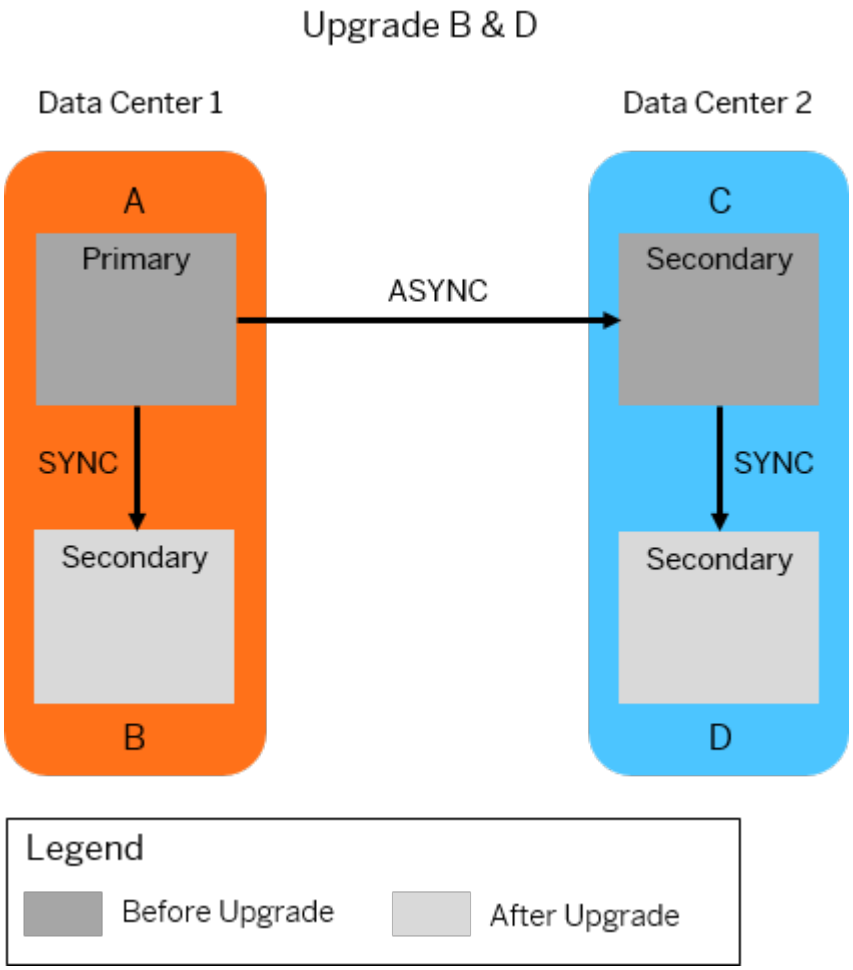
We are using the following setup to exemplify the procedure:

> ### Example
>
> Primary system A replicates data changes to secondary system B located in the same data center. Primary system A also replicates data changes to the secondary system C located in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center with system C.
>
> In this setup:
>
> - The primary system is the production system.
> - The secondary system located in the same data center as the primary system will become the production system after the upgrade. Further secondary systems are located in a remote data center.

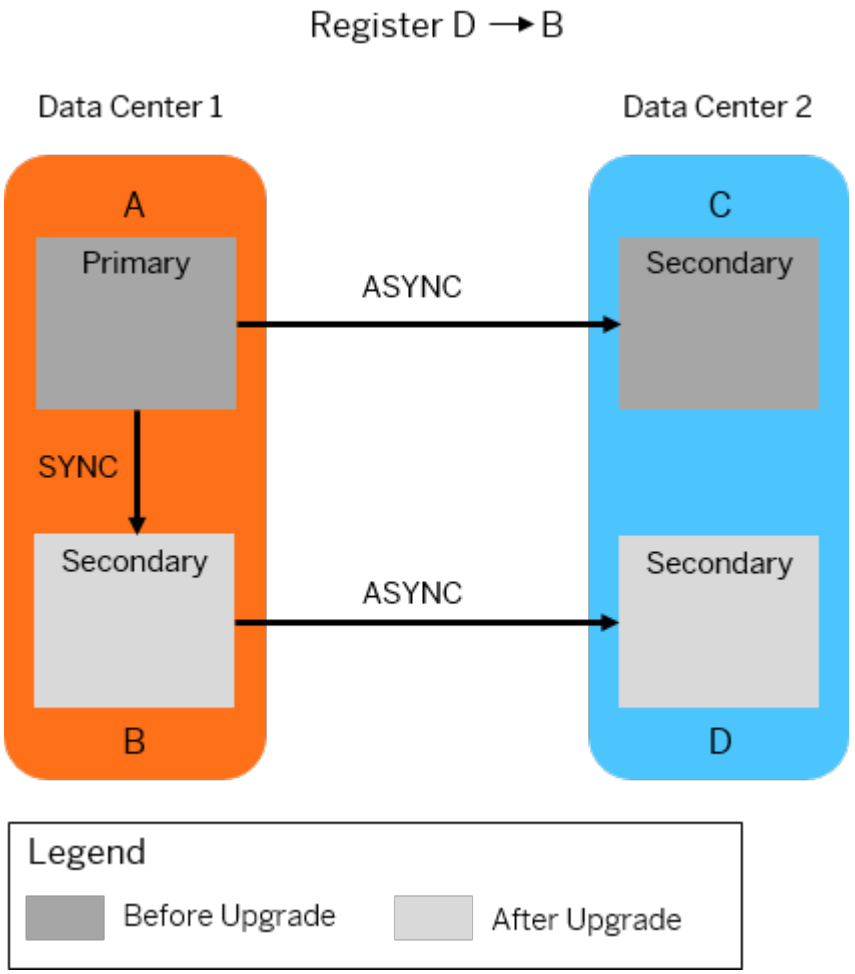
- There is no replication error.

Procedure

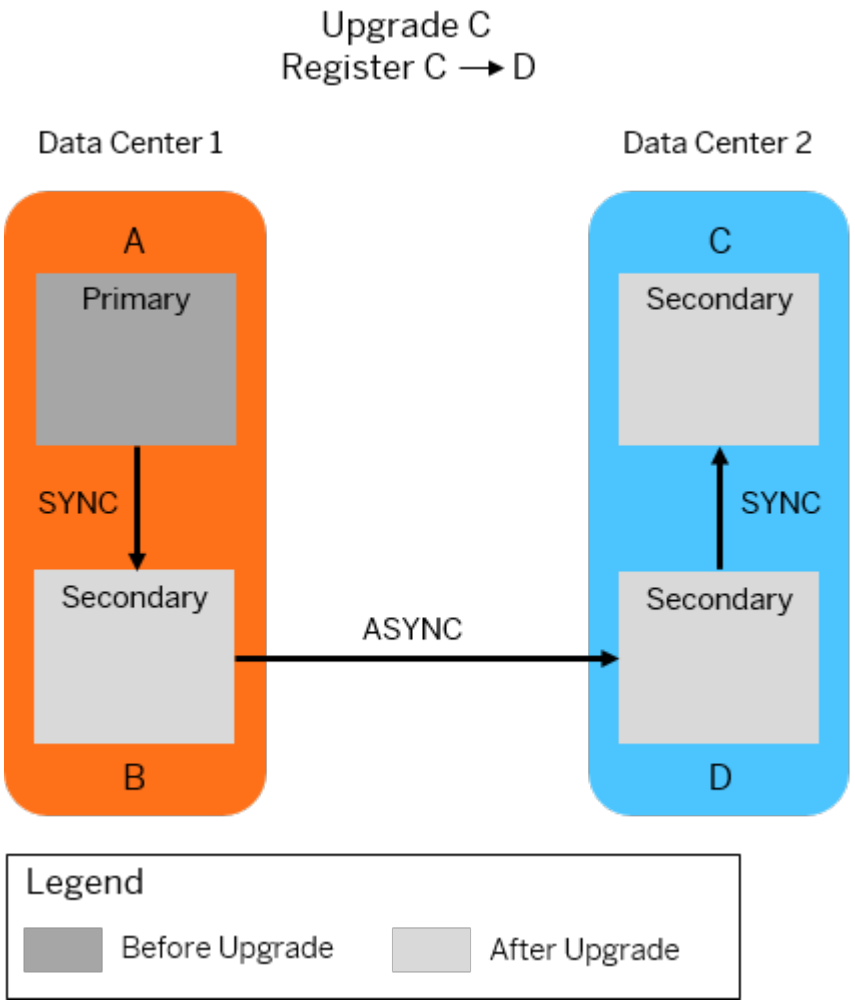
1. Upgrade secondary system B in data center 1 and secondary system D in data center 2.



2. Register secondary system D in data center 2 to secondary system B in data center 1.

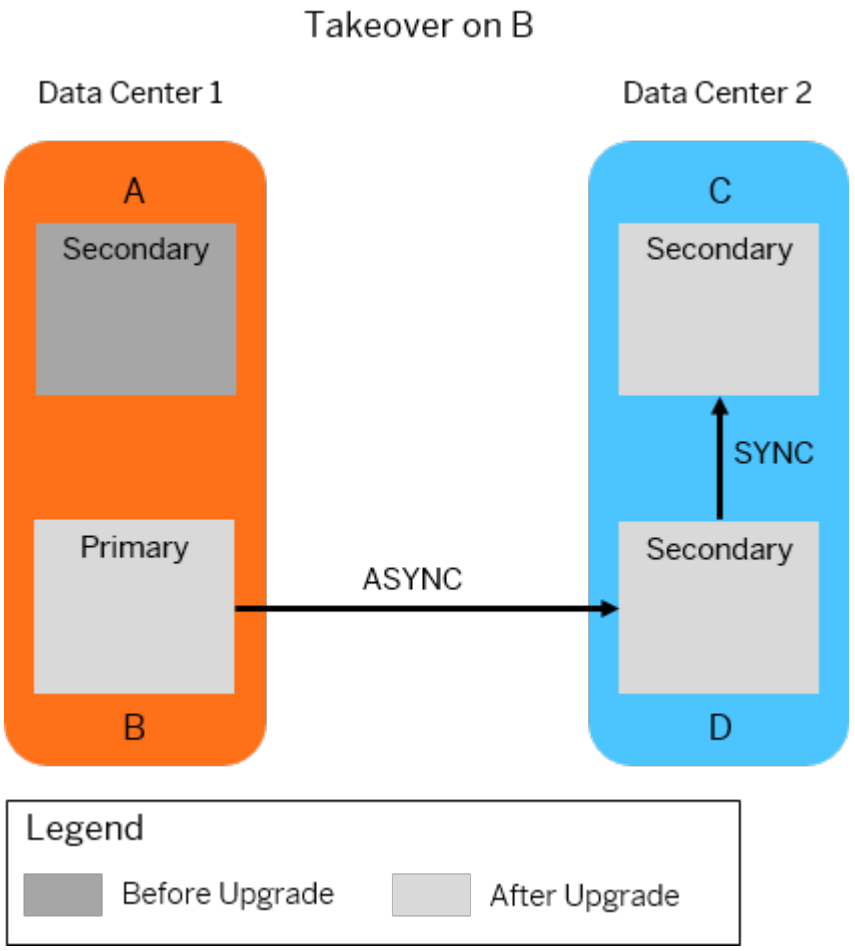


3. Upgrade secondary system C in data center 2. Then, register secondary system C to secondary system D in data center 2.

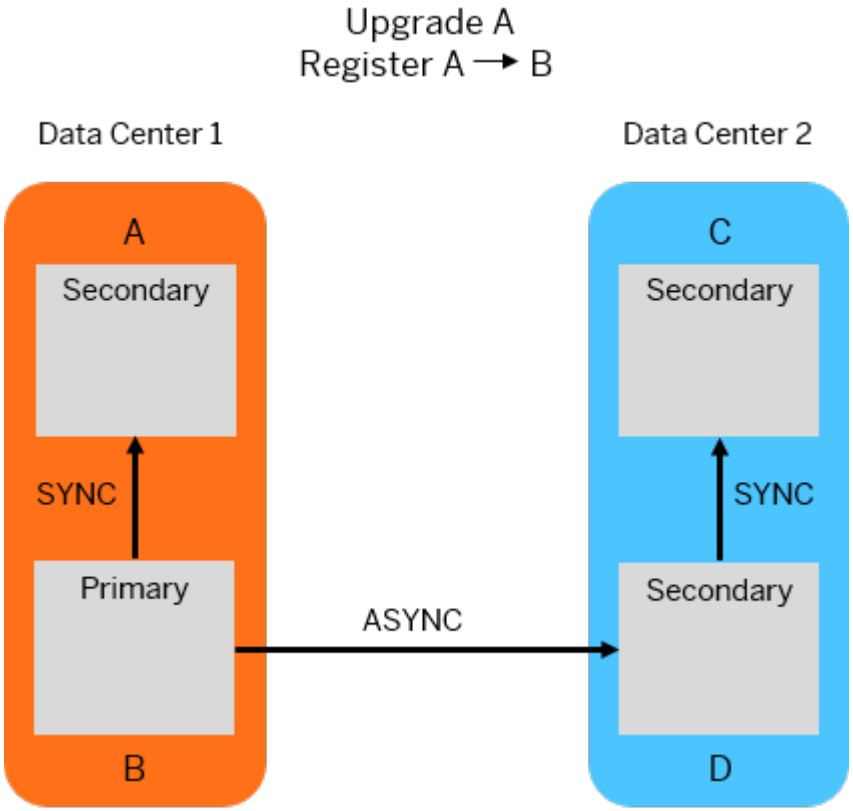


4. Take over on secondary system B in data center 1.

   After takeover, secondary system B will be the new primary system.



5. Upgrade and register the previous primary system A to the new primary system B in data center 1.



Related Information

SAP HANA Multitarget System Replication [page 846]

12.1.2.18.3 Update SAP HANA Systems Running in a System Replication Setup

You can update your SAP HANA system with active system replication by updating the secondary and the primary system one after the other.

Prerequisites

System replication is configured and active between two SAP HANA systems.

Context

You must update your SAP HANA system running in a system replication setup by updating the secondary system first and then updating the primary system.

> → Remember
>
> For system replication setups it is required that the secondary system has the same version as the primary system or a higher version. As such, the secondary system must always be updated before the primary system.

> ⓘ Note
>
> Updating one system after the other results in some downtime. If you want to update your system with reduced downtime, see *Use SAP HANA System Replication for Near Zero Downtime Upgrades*.
>
> It is possible to reduce the time required to perform an update. For more information, see *Prepare an Update for Flexible System Downtime* .

Procedure

1. Upgrade the SAP HANA server software and all installed components on the secondary system.

   From your installation directory execute as root or as `<sid>adm`:

   ```
   ./hdblcm --action=update
   ```

2. With the secondary system online, use the SAP HANA lifecycle management tools to upgrade all the other components to the same revision as the server software.
3. Verify that system replication is active and that all services are in sync.

   You can check that the REPLICATION_STATUS column in M_SERVICE_REPLICATION has the value ACTIVE for all services.
4. Upgrade the SAP HANA server software and all installed components on the primary system.

   From your installation directory, execute as root or as `<sid>adm`:

   ```
   ./hdblcm --action=update
   ```

5. With the primary system online, use the SAP HANA lifecycle management tools to upgrade all other components to the same revision as the server software.
6. Verify that system replication is active and that all services are in sync.

Related Information

Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 885]
Prepare an Update for Flexible System Downtime

SAP Note 2407186
SAP Note 2599514

12.1.2.19 Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites

SSL is enabled by default but you may need to configure TLS/SSL on communication channels between systems using the system public key infrastructure (PKI).

Prerequisites

- You have the credentials of the operating system user, `<sid>`adm.
- You have the system privilege INIFILE ADMIN.

Context

From SAP HANA SPS 07 onwards SSL is enabled by default to secure the following communication channels using TLS/SSL:

- Metadata channel used to transmit metadata (for example, topology information) between the sites
- Data channel used to transmit data between the sites.

Secure communication is activated or deactivated by setting the `enable_ssl` parameter in the `global.ini` file of the system database (setting the parameter for a tenant database is not supported). The parameter can be configured online without interrupting replication, however, for systems using Dynamic Tiering (see following note) a restart is required to make the change effective. You must apply SSL simultaneously for all systems in the replication landscape and the procedure described here assumes that no systems in the landscape are running an older version.

> **Note**
>
> On SAP HANA systems with Dynamic Tiering the `enable_ssl` parameter also enables the system PKI for internal system replication communication. No additional steps are required, however, in the case of Dynamic Tiering a restart of the database is required after changing the value of the parameter. Before you configure communication for Dynamic Tiering see SAP Note 2447994 - *SAP HANA Dynamic Tiering Support for SAP HANA System Replication*.
>
> Be aware that you need additional licenses for SAP HANA options and capabilities. For more information, see *Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities*.

Procedure

In the primary and secondary system, enable TLS/SSL for the data channel.

Set the property `[system_replication_communication] enable_ssl` in the `global.ini` of the system database; the following values are possible, these can also be used with a specific site name as shown in the example below:

Value	Description
off	TLS/SSL is disabled for replication source and target systems
on (default)	TLS/SSL is enabled for replication source and target systems

For a simple system replication scenario involving two systems set the property to **on** in both systems.

For multitier and multitarget system replication scenarios involving three systems, you can apply **on** in all 3 systems to secure all system replication connections. Alternatively, you can use the `site_name` value as index to secure either only the communication to the tier 3 secondary system or only the communication to the primary system.

> **Example**
>
> To exclude the communication between the primary and the secondary, and to secure the communication between all other systems, set the parameter as follows:

```
siteA        ------>          siteB         ------>       siteC
enable_ssl=on                enable_ssl=on                  enable_ssl=on
enable_ssl[siteB]=off        enable_ssl[siteA]=off
```

> **Note**
>
> To avoid communication failure between systems, TLS/SSL must be enabled on all systems at the same time. TLS/SSL will be used as long as the replicating systems are registered and enabled.

Related Information

SAP Note 2447994

12.1.3 Setting Up Host Auto-Failover

Host auto-failover is a local fault-recovery solution that can be used as a supplemental or alternative measure to system replication. One (or more) standby hosts are added to an SAP HANA system, and configured to work in standby mode.

The databases on the standby hosts do not contain any data and do not accept requests or queries as long as they are in standby mode.

When an active worker host fails, a standby host automatically takes its place. Since the standby host may take over operation from any of the primary hosts, it needs access to all the database volumes. This can be accomplished by a shared networked storage server, by using a distributed file system, or with vendor-specific solutions that can dynamically mount networked storage upon failover.

For more information about how to add an additional standby host, see *Adding Hosts to an SAP HANA System*.

You can monitor the status of all active and standby hosts in the SAP HANA cockpit and the SAP HANA studio.

A HA/DR provider script is available to provide hooks that can be called in response to events during host auto-failover. For more information, see *Implementing a HA/DR Provider*.

Related Information

Adding Hosts to an SAP HANA System [page 1089]
Configuring Clients for Failover [page 897]
Configuring Application Servers for Failover [page 898]
Configure HTTP Load Balancing for SAP HANA Extended Application Services, Classic Model [page 900]
Implementing a HA/DR Provider [page 905]
SAP HANA Host Auto-Failover
SAP HANA Storage Requirements

12.1.3.1 Configuring Clients for Failover

You can configure failover support for clients so that they continue to work in a transparent way to the user in the event of a failover.

SAP HANA clients that were configured to reach the original host need to be sent to the standby host after host auto-failover.

One way to handle this is using a network-based (IP or DNS) approach. Alternatively, SQL/MDX database clients can be configured with the connection information of multiple hosts, optionally including the standby host, by providing a list of hosts in the connection string. The client connection code uses a "round-robin" approach to reconnect, thus ensuring that the client can reach the SAP HANA database, even after failover.

To support failover with client libraries, you have to specify a list of host names separated by a semicolon instead of a single host name. Only hosts that have the role master or standby should be used.

To determine which hosts to use, execute the following SQL statement:

```
SELECT HOST FROM M_LANDSCAPE_HOST_CONFIGURATION WHERE NAMESERVER_CONFIG_ROLE LIKE
'MASTER%' ORDER BY NAMESERVER_CONFIG_ROLE
```

Since one of these master candidates will be active, only they have to be added. When hosts are added to a system, the master list is extended to three hosts, meaning there is one host configured as the actual master and two worker hosts are configured as master candidates. When the first standby host is added to the system, a worker host is removed from this list and replaced by the standby host. This is done because it is faster to fail over to an idle standby host than to an active worker host.

The client will choose one of these hosts to connect to. If a host is not available, the next host from the list will be used. Only in the case that none of the hosts are available will you get a connection error.

If a connection gets lost when a host is not available any longer, the client will reconnect to one of the host specified in the host list.

Example Configurations

Client	Example
JDBC	`Connect URL:` `jdbc:sap://host1:30015;host2:30015;host3:30015/`

SQLDBC	`SQLDBC_Connection`
	`*conn = env.createConnection();`
	`SQLDBC_Retcode rc = conn->connect`
	`("host1:30015;host2:30015;host3:30015", "", "user",`
	`"password");`

ODBC	`Connect URL:`
	`"DRIVER=HDBODBC32;`
	`UID=user;`
	`PWD=password;`
	`SERVERNODE=host1:30015,host2:30015,host3:30015";`

HTTP Client Access via SAP HANA Extended Application Services, Classic Model

To support HTTP (Web) clients accessing SAP HANA via the SAP HANA XS classic server, it is recommended to install an external, itself fault protected, HTTP load balancer (HLB), such as SAP Web Dispatcher, or a similar product from another vendor. The HLBs are configured to monitor the Web servers on all hosts on all sites. For more information see, *Configuring HTTP Load Balancing for SAP HANA Extended Application Services (XS)*.

If an SAP HANA instance fails, the HLB, which serves as a reverse web-proxy, redirects the HTTP clients to the running SAP HANA XS instance on an active host. HTTP clients are configured to use the IP address of the HLB itself, which is obtained via DNS, and remain unaware of any SAP HANA failover activity.

Related Information

Client Connection Recovery After Takeover [page 817]

12.1.3.2 Configuring Application Servers for Failover

You can configure failover support for application servers by using the secure user store of the SAP HANA client (`hdbuserstore`) to specify a list of host names that the server can connect to.

For the clients in a host auto-failover landscape, the use of virtual IP addresses is recommended. You can store user logon information, including passwords, in the secure user store of the SAP HANA client

(`hdbuserstore`). This allows client programs to connect to the database without having to enter a password explicitly.

The `hdbuserstore` can also be used to configure failover support for application servers (for example, for SAP Business Warehouse) by storing a list of all (virtual) host names to which the application server can connect. All nodes that are master candidates must be added to the `hdbuserstore`.

> → Tip
>
> For more information about how to find out the three master candidates in a distributed system, see SAP Note 1930853.

The application server will choose one of these hosts to connect to from the list. If a host is not available, the next host from the list will be used. Only if none of the hosts are available will you get a connection error. If a connection gets lost when a host is no longer available, the application server will reconnect to one of the hosts specified in the host list.

You can specify a list of host names in the secure user store using the following `hdbuserstore`:

```
hdbuserstore SET default "<hostname_node1>:3<system_number>15; ….. <
hostname_node(n)>: 3<system_number>15" SAP<sid> <Password>
```

> {·} Sample Code
>
> ```
> hdbuserstore SET default
> "ld9490:33315;ld9491:33315;ld9492:33315;ld9493:33315" SAPP20 <password>
> ```

KEY default

```
ENV : ld9490:33315;ld9491:33315;ld9492:33315;ld9493:33315
  USER: SAPP20
```

For more information about `hdbuserstore`, see the *SAP HANA Security Guide* and the *SAP HANA Client Interface Programming Reference*

Related Information

Client-Side Data Security

Use the User Store (hdbuserstore) (SAP HANA Client Interface)

SAP Note SAP Note 1930853

12.1.3.3 Configure HTTP Load Balancing for SAP HANA Extended Application Services, Classic Model

To enable load balancing for HTTP access to the SAP HANA XS classic sever, you need to set up a load balancer (for example, SAP Web Dispatcher).

Context

To support HTTP (Web) clients accessing SAP HANA via the SAP HANA XS classic server, it is recommended to install an external, itself fault protected, HTTP load balancer (HLB), such as SAP Web Dispatcher, or a similar product from another vendor. The HLBs are configured to monitor the Web servers on all hosts on all sites.

The SAP Web Dispatcher automatically reads the system topology from SAP HANA XS classic and is notified of changes to the topology, for example, when a host is no longer available or a standby host has taken over. The SAP Web Dispatcher then sends requests to a running XS instance on an active host. Third-party load balancers often use a static configuration with an additional server availability check.

The SAP Web Dispatcher can be configured with a list of the three master hosts. Once one of the master hosts is available the SAP Web Dispatcher acquires the topology information. HTTP clients can be configured to use the IP address of the HTTP load balancer itself, and remain unaware of any SAP HANA failover activity.

> ⓘ Note
>
> For more information about using and configuring the SAP Web Dispatcher for load balancing with SAP HANA multitenant database containers, see *Using SAP Web Dispatcher for Load Balancing with Tenant Databases*.

Procedure

1. Install SAP Web Dispatcher with a minimum release of 7.40 using the SAP NetWeaver Software Delivery Tool and update it to the latest version available on the SAP Software Download Center.
2. Log on to the SAP Web Dispatcher host as the `<SID>`adm user. Here the `<SID>` refers to the one of the SAP Web Dispatcher installation.
3. Open the instance profile of your SAP Web Dispatcher.

   The SAP Web Dispatcher profile can be found in the following location:

   ```
   usr/sap/<SID>/SYS/profile
   ```

4. Disable the ABAP system configuration, which is done automatically during the installation by commenting out the entries in this section of the profile:

   ```
   # Accessibility of Message Servers
   --------------------------------
   #rdisp/mshost = ldcialx
   #ms/http_port = 8110
   ```

5. Add a list of semicolon separated URLs and the base URL (without path) used for fetching topology information, to the XSSRV parameter in the profile.
   An example could be:

```
wdisp/system_0 = SID=HDX, XSSRV=http://ld9490:8089;http://ld9491:8089,
SRCSRV=*
```

Related Information

SAP Web Dispatcher
SAP Note 1855097
SAP Note SAP Note 1883147
Using SAP Web Dispatcher for Load Balancing with Tenant Databases [page 115]

12.1.3.4 Host Auto-Failover Parameters

This topic provides an overview of parameters available for Host Auto-Failover.

Master Failover without Standby Hosts

Distributed landscapes without standby hosts may also perform a failover to ensure that the master host is always available.

Parameter	`nameserver.ini/[failover]/enable_master_failover`
Description:	When set to false, the masterize check of the nameserver master candidates is disabled. Furthermore, adding a new host does not modify the master candidates list.
Online Change:	Yes
Default:	True

Failover Groups

During installation a failover group can be configured per host. If a failover target host is available in the same group, it will be preffered over hosts from other groups. This can be used to achieve better "locality" in large systems (for example, to use network or storage connection with less latency). It can also be used to separate differently sized hardware or storages.

Parameter	nameserver.ini/[failover]/cross_failover_group
Description:	When set to false, failover is restricted to the hosts in the same group.
Online Change:	Yes
Default:	True

Automatic Host Shutdown by Service Failures

For every service a fixed number of restarts can be defined after which the daemon stops. The nameserver is the only service that has the first parameter set to true as default. This means that any problem involving a constant nameserver crash will eventually stop the daemon.

Parameter	daemon.ini/[failover]/startup_error_shutdown_instance
Description:	When set to true, the daemon will shut down all services on the host if this service cannot start.
Online Change:	Yes
Default:	True

Parameter	daemon.ini/[failover]/startup_error_restart_retries
Description:	The number of retries if a service fails in the startup procedure.
Online Change:	Yes
Default:	4

12.1.3.5 Host Auto-Failover Setup with XS Advanced Run Time

Understand the host auto-failover setup for the XS advanced run time.

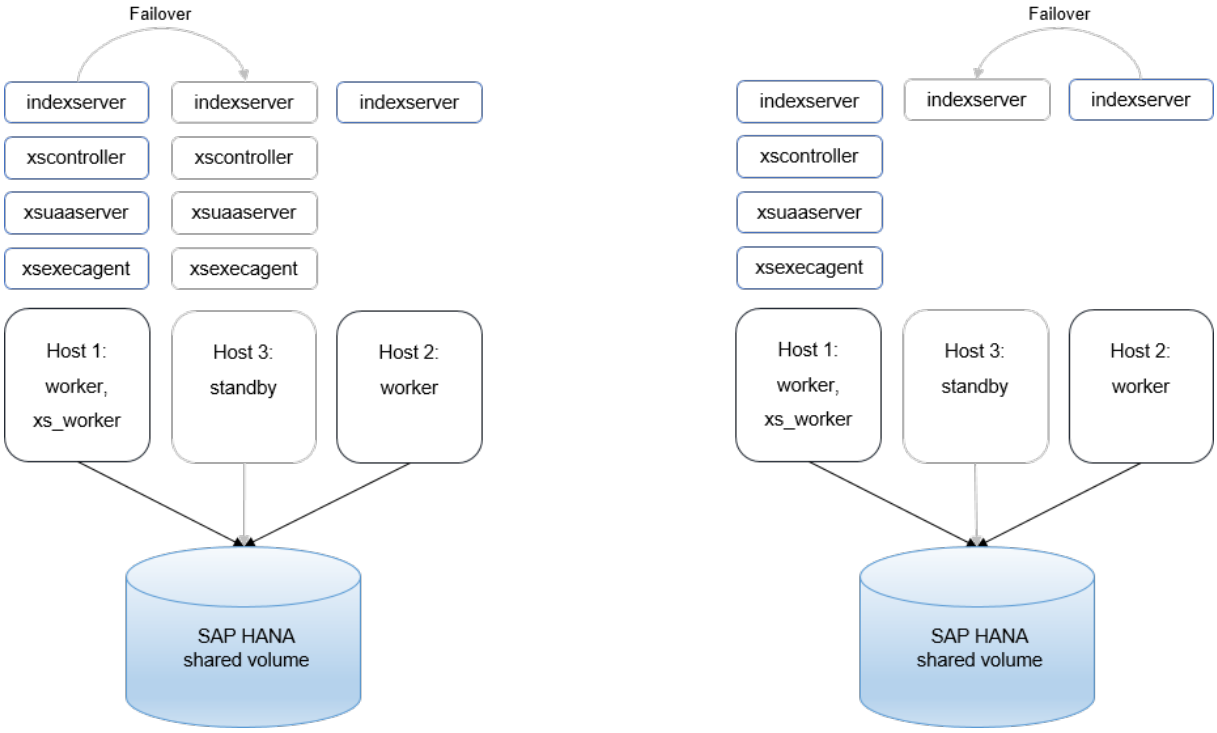
The host auto-failover setup for XS advanced integrates with failover mechanisms supported by SAP HANA. For more information, see *Setting Up Host Auto-Failover* in *Related Information*. Since host auto-failover always requires a multi-host setup, you have to fulfill all the prerequisites of a multi-host setup with XS advanced, as described in *Multi-Host Setup with XS Advanced* in *Related Information*. The following new host roles have been introduced for use with the XS advanced run time:

- `xs_worker`

- `xs_standby`.

Host Auto-Failover Without Separate `xs_worker` Hosts

A host with the role `standby` automatically takes over the load of a failing host with the role `worker`. At the same time, the standby host takes over any additional roles that were assigned to the failing worker host. That means that when the `xs_worker` role is assigned to a host with role `worker`, a host with role `standby` would automatically take over the role `xs_worker` from the worker host as well. One consequence of this is that it takes over the XS advanced processes as well, as illustrated in the scenario on the left-hand side of the following figure:

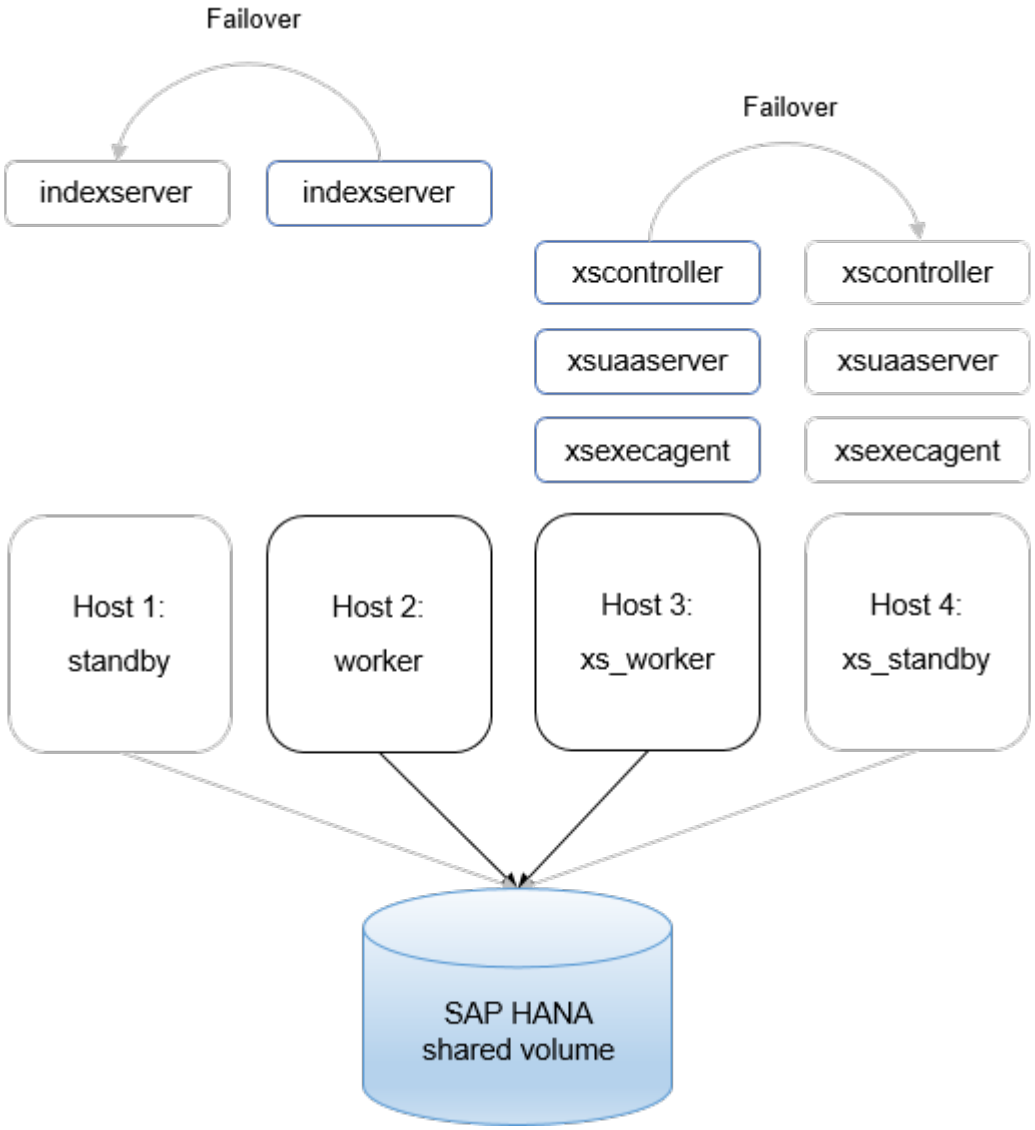


> **Note**
>
> A standby host only participates in the failover procedure once.

For example, in the scenario illustrated on the right-hand side of the figure above, the standby `Host 3` takes over the roles from `Host 2` as this host fails first. Since `Host 3` is busy as a result of the failover, there is no standby host left to take over XS advanced processes if `Host 1` also fails. So, in this example, if you want to have a failover host available to take over XS advanced processes after `Host 2` fails, you need to specify an additional host with a standby role (for example, `Host 4`, which is not depicted in the figure above) to cope with this situation.

Host Auto-Failover With Separate `xs_worker` Hosts

To set up failover hosts for separate hosts with the `xs_worker` role, hosts with the host role `xs_standby` are required. In the following example, separate hosts with the roles `standby` and `xs_standby` take over the load for either SAP HANA services (`worker`) or XS advanced services (`xs_worker`) and applications respectively.



It is only recommended to combine `standby` and `xs_standby` roles on one host if hosts with the `xs_worker` and `worker` role have the same hardware specifications. Otherwise, XS advanced processes could be assigned to hosts with SAP HANA database certified hardware after failover as shown by the following example:

- The host with the `worker` role uses SAP HANA database certified hardware.
- The host with the `xs_worker` role uses commodity hardware (but SAP HANA compliant).
- The host with the `xs_worker` role fails first.

In this scenario, the combined `standby` / `xs_standby` host takes over `xs_worker` only and cannot be used to take over the separate host with the `worker` role anymore. The XS advanced processes would then run on a standby host with over-sized hardware.

Related Information

Setting Up the XS Advanced Runtime Behind a Reverse Proxy

Setting Up Host Auto-Failover [page 896]
Multi-Host Setup with XS Advanced [page 1512]

12.1.4 Implementing a HA/DR Provider

The SAP HANA nameserver provides a Python-based API, which is called at important points of the host auto-failover and system replication takeover process.

API calls (so called "hooks" or HA/DR providers) can be used for many operations that need to be executed. One of the most important uses of the failover hooks is moving around a virtual IP address (in conjunction with STONITH). However, there are other purposes like starting tools and applications on certain hosts after failover, or even stopping DEV or QA SAP HANA instances on secondary sites before takeover. Multiple failover hooks can be installed and used in parallel with a defined execution order.

> **Note**
>
> When calling subprocesses within a HA/DR Provider implementation, please refrain from using the Python modules `subprocess` and `popen2`, as well as `os.popen()`, `os.popen2`, `os.popen3`, and `os.popen4`. Those methods allocate memory, which can cause a deadlock when forking the nameserver process. Use `os.system()` instead.

The topics in this section describe the Python API in detail and give an example how the hooks can be used. An implemented Python class is called a HA/DR provider. This script contains hook methods, which are called at certain events.

The failover hooks are included in SAP HANA - the failover hook API has its own version number. SAP HANA includes a Python interpreter which is used for interpreting the user defined failover hooks. Only Python 3 is supported in SAP HANA SPS 06; for information about migrating scripts from Python 2 refer to SAP Note 3070359 *Python 3 Migration Guide For SAP HANA*.

Related Information

SAP Note 3070359

12.1.4.1 Create a HA/DR Provider

You can adapt Python files delivered with SAP HANA to create your own HA/DR provider. This allows you to integrate, for example, SAP HANA failover mechanisms into your existing scripts.

Context

To create your own HA/DR provider, the following steps must be executed and then add the methods you want to use from those listed in *Hook Methods*:

Procedure

1. Create a new directory for the HA/DR provider

   The directory should be within the shared storage of the SAP HANA installation, but outside the <SID> directory structure (otherwise it is likely to be deleted/overwritten during a SAP HANA update).

   For example you could use the following location: `/hana/shared/myHooks`
2. Copy the `exe/python_support/hdb_ha_dr/HADRDummy.py` from an installed SAP HANA system to the new location.
   For example, copy the file to: `/hana/shared/myHooks/myFirstHook.py`

   > **Note**
   >
   > Do not copy the client.py, otherwise updates and new features will be missed when updating SAP HANA. When using the import statement as described below the client.py from the SAP HANA installation will be used.
3. Adapt the contents of the new file by renaming the Python class to the name of the file, for an example see Code Listing 1
4. Fill out the Python dict in the about() method, for an example see Code Listing 1.

   > **Sample Code**
   >
   > Code Listing 1

```
from hdb_ha_dr.client import HADRBase, Helper
import os, time
class myFirstHook(HADRBase):
    apiVersion = 1
    def __init__(self, *args, **kwargs):
        # delegate construction to base class
        super(myFirstHook, self).__init__(*args, **kwargs)
    def about(self):
        return {"provider_company" :        "SAP",
                "provider_description" :    "Template Dummy Provider",
                "provider_version" :        "1.0"}
```

Within the SAP HANA environment, the path `exe/python_support` is part of the PYTHONPATH setting. Therefore, `hdb_ha_dr` can be used as module for the import of the base class in helper class (shown in the first line of Code Listing 1).

The attribute `apiVersion` defines which HA/DR provider API version will be used.

The `__init__()` method should always call the super method with the parameters *args, **kwargs in order to ensure the correct initialization of the tracer and configuration file wrapper. If required, additional initialization steps can be used here.

Finally, the `about()` method must return a Python dict with the keys as shown. The values will be used for monitoring in the M_HA_DR_PROVIDERS view.

There are three class attributes defined in the HA/DR provider base class::

- `tracer` : a tracer that is available to all derived classes tracing to the nameserver's trace file
- `config` : a wrapper for easy access of optional configuration parameters in the global.ini
- `apiVersion` : the definition of the API version

More details can be found in the sections Additional Configuration Parameters and Tracing (see Related Information).

Next Steps

With the basic HA/DR provider now implemented you can continue by choosing and adding the methods listed in *Hook Methods* to your provider.

Related Information

Hook Methods [page 907]
Install and Configure a HA/DR Provider Script [page 914]

12.1.4.1.1 Hook Methods

There are a number of pre takeover, post takeover and general hooks available for you to use.

The following hook methods are available. Hooks are called from either the nameserver of the coordinator (previously referred to as the 'master') host or the nameserver of another host in the landscape. The hook acts on a host, site or service as indicated in the table.

Name	Trigger	Caller	Error behavior
startup()	Beginning of nameserver's start up phase	Called on the starting nameserver of any individual host	Nameserver aborts, start up is canceled

Name	Trigger	Caller	Error behavior
shutdown()	Just before the nameserver exits	Called on the stopping nameserver of any individual host	Error trace is written
failover() [Host Auto-Failover]	As soon as the nameserver made a decision about the new role	Called on each host that takes over a new role	Nameserver aborts, failover is canceled
stonith() [Host Auto-Failover]	As soon as the nameserver made the decision about the new role	Called on the master host nameserver for each failing host	Nameserver aborts, failover is canceled
preTakeover() [System Replication]	As soon as the hdbnsutil -sr_takeover command is issued	Called only once on the master host nameserver of the replicating site	Takeover is aborted
postTakeover() [System Replication]	As soon as all services with a volume return from their assign-call (open SQL port)	Called only once on the master host nameserver of the replicating site	Error trace is written
srConnectionChanged() [System Replication]	As soon as one of the replicating services loses or (re-)establishes the system replication connection	Called only once on the master host nameserver on the primary site for each replicating service	Error trace is written
srServiceStateChanged()	As soon as the nameserver made a decision about the new state	Called on the individual host where a service state has changed	Error trace is written
srReadAccessInitialized() [System Replication]	As soon as a tenant database or the system database is ready to accept SQL read queries on a read enabled secondary system	Called on master host on primary for each tenant	Error trace is written
srSecondaryUnregistered() [System Replication]	When the primary system is informed about a takeover or unregister of one of its secondary systems	Called only once on the master host on the primary site for a replicating site	Error trace is written

All hook methods receive a set of parameters, which can be used to identify the state and configuration of the calling nameserver. The nameserver expects the return code 0 in case of successful execution and codes other than 0 for the error case.

The hook methods are shown in detail in the following sections.

Startup hook method stub

Code Listing 3 shows the startup hook method stub.

```
 def startup(self, hostname, storage_partition, system_replication_mode,
**kwargs):
    """
    Hook description:
```

```
    * time of call: beginning of startup of the nameserver
    * caller: the starting host
    * landscape: each host calls it individually
    * behavior upon failure: nameserver aborts, startup is canceled
    @param hostname: the local hostname
    @type hostname:  string
    @param storage_partition: the storage partition number, 0 for standby hosts
    @type storage_partition: int
    @param system_replication_mode: mode of system replication
    @type system_replication_mode: string
    @param **kwargs: place holder for later usage (new parameters) to
             keep the interface stable
    @type **kwargs: dict
    @return: information about success
    @rtype: int
    """
    return 0
```

Shutdown hook method stub

Code Listing 4 shows the shutdown hook method stub.

> **Note**
>
> The invocation of the shutdown() method is not guaranteed. If the nameserver is terminated prematurely it cannot call the HA/DR provider, for example if a host fails or during SAP HANA shutdown, when the services have not had enough time to run through their shutdown routines. Therefore crucial tasks such as the deletion of virtual IPs need to be implemented in the startup() and failover() method.

```
  def shutdown(self, hostname, storage_partition, system_replication_mode,
**kwargs):
   """
   Hook description:

   * time of call: just before the nameserver exits
   * caller: the stopping host
   * landscape: each host calls it individually
   * behavior upon failure: error trace is written
   @param hostname: the local hostname
   @type hostname:  string
   @param storage_partition: the storage partition number, 0 for standby hosts
   @type storage_partition: int
   @param system_replication_mode: mode of system replication
   @type system_replication_mode: string
   @param **kwargs: place holder for later usage (new parameters) to
      keep the interface stable
   @type **kwargs: dict
   @return: information about success
   @rtype: int
   """
   return 0
```

Failover hook method stub

Code Listing 5 shows the failover hook method stub.

```
 def failover(self, hostname, storage_partition, system_replication_mode,
**kwargs):
   """
   Hook description:

   * time of call: when the nameserver made the decision about the new role
   * caller: the host that takes over the role
   * landscape: called on each host that gets a new role
   * behavior upon failure: nameserver aborts, failover is canceled
   @param hostname: the local hostname
   @type hostname:  string
   @param storage_partition: the storage partition number, 0 for standby hosts
   @type storage_partition: int
   @param system_replication_mode: mode of system replication
   @type system_replication_mode: string
   @param **kwargs: place holder for later usage (new parameters) to
      keep the interface stable
   @type **kwargs: dict
   @return: information about success
   @rtype: int
   """
   return 0
```

stonith hook method stub

Code Listing 6 shows the stonith hook method stub.

```
 def stonith(self, failing_host, **kwargs):
   """
   Hook description:

   * time of call: when the nameserver made the decision about the new role
   * caller: the master host
   * landscape: for each failed host
   * behavior upon failure: nameserver aborts, failover is canceled
   @param failing_host: the SAP HANA internal name of the failed host
   @type failing_host: string
   @param **kwargs: place holder for later usage (new parameters) to
      keep the interface stable
   @type **kwargs: dict
   @return: information about success
   @rtype: int
   """
   return 0
```

preTakeover hook method stub

Code Listing 7 shows the preTakeover hook method stub.

```
 def preTakeover(self, isForce, **kwargs):
   """
```

```
    Hook description:

    * time of call: as soon as the hdbnsutil -sr_takeover command is issued
    * caller: the master host
    * landscape: called only once on the master
    * behavior upon failure: nameserver aborts, takeover is canceled
    @param isForce: flag if it is a normal or forced takeover (as of today,
      takeover is always forced regardless of the value of the
      flag)
    @type isForce: bool
    @param **kwargs: place holder for later usage (new parameters) to
       keep the interface stable
    @type **kwargs: dict
    @return: information about success
    @rtype: int
    """
    return 0
```

postTakeover hook method stub

Code Listing 8 shows the postTakeover hook method stub.

```
def postTakeover(self, rc, **kwargs):
    """
    Hook description:

    * time of call: as soon as all services with a volume return from their
      assign-call (open SQL port)
    * caller: the master host
    * landscape: called only once on the master
    * behavior upon failure: error trace is written
    @param rc: the return code of the actual takeover process; 0=success,
      1=waiting for forced takeover, 2=failure
    @type rc: int
    @param **kwargs: place holder for later usage (new parameters) to
       keep the interface stable
    @type **kwargs: dict
    @return: information about success
    @rtype: int
    """
    return 0
```

srConnectionChanged hook method stub

Code Listing 9 shows the srConnectionChanged hook method stub.

```
def srConnectionChanged(self, parameters, **kwargs):
    """
    Hook description:
    * time of call: as soon as one of the replicating services loses or
      (re-)establishes the system replication connection
    * caller: master node on primary site
    * landscape: called only once on the master node on primary site
    * behavior upon failure: error trace is written
    * Possible return codes:
    * 0: Ok - continue processing
```

```
   * 1: Block - Further HANA processing is blocked if HANA lost sync (is_in_sync
= false).
         Every 5 sec. there will be a retry to call this hook.
   * If an HA/DR Provider shall not block SAP HANA processing or if a blocking
       situation caused by an HA/DR Provider shall (temporarily) be
       resolved use ‘hdbnsutil –sr_blockonconnectionchanged --disable’
   @param parameters: dict of parameters {hostname:string, port:string,
        database:string, status:int, database_status:int,
        system_status:int, timestamp:string, is_in_sync:bool, reason:string}
   @type parameters: dict
   @param **kwargs: place holder for later usage (new parameters) to keep the
      interface stable
   @type **kwargs: dict
   @return: information about success (0 = continue transaction, 1 = halt
       further transactions)
   @rtype: int
   * parameters:
   * -- hostname: host where the service is running
   * -- port: service's port
   * -- database: service's tenant database (MDC)
   * -- status: service replication status (10: NoHSR, 11: Error, 12: Unknown,
13: Initializing, 14: Syncing, 15: Active)
   * -- database_status: tenant database replicating status (10: NoHSR, 11:
Error, 12: Unknown, 13: Initializing, 14: Syncing, 15: Active)
   * -- system_status: HANA database overall replicating status (10: NoHSR, 11:
Error, 12: Unknown, 13: Initializing, 14: Syncing, 15: Active)
   * -- timestamp: date and time of the event
   * -- is_in_sync: true if service is in sync
   * -- reason: additional details (e.g. 'Starting', 'Stopping')
   * -- siteName: name of the replicating secondary site
   """
   return 0
```

srServiceStateChanged hook method stub

Code Listing 10 shows the srServiceStateChanged hook method stub.

```
 def srServiceStateChanged(self, parameters, **kwargs):
    """
    Hook description:
    * time of call: as soon as the nameserver made a decision about the new
state
    * caller: host that detects a local service change
    * landscape: each individual host
    * behavior upon failure: error trace is written
    @param parameters: dict of parameters {hostname:string, service_name:string,
service_port:string, service_status:string, timestamp:string}
    @type parameters: dict
    * parameters:
    * -- hostname: host where the service state has changed
    * -- service_name: name of the service
    * -- service_port: port of the service
    * -- service_status: (no, yes, unknown, starting, stopping)
    * -- timestamp: date and time of the service change event
    * -- daemon_status: the state of the host based on the daemon's runlevel.
values can be no, yes, starting, stopping.
    * -- database_id: numeric identifier of the tenant database (if it's a
tenant-related service)
    * -- database_name: name of the tenant database (if it's a tenant-related
service)
```

```
    * -- database_status: the state of the tenant database. values can be no,
yes, unknown, starting, stopping.
    * -- details: additional information, e.g. 'defunct', 'crash', ...

    """
    return 0
```

srReadAccessInitialized hook method stub

Code Listing 11 shows the srReadAccessInitialized hook method stub.

```
 def srReadAccessInitialized(self, parameters, **kwargs):
   """
   Hook description:
   * time of call: when a tenant database or the SystemDB is ready to accept SQL
read queries on a read enabled secondary system
   * caller: master node on primary site
   * landscape: called only once on the master node primary site
   * behavior upon failure: error trace is written
   @param parameters: dict of parameters {last_initialized_database:string,
databases_with_read_access_initialized:list<string>,
databases_without_read_access_initialized:list<string>, timestamp:string,
all_databases_initialized:bool}
   @type parameters: dict
   @param **kwargs: place holder for later usage (new parameters) to keep the
interface stable
   @type **kwargs: dict
   * parameters:
   * -- last_initialized_database: tenant database that has finished its read
access initialization and is ready to accept SQL requests
   * -- databases_with_read_access_initialized: list of tenant databases that
have finished their read access initialization
   * -- databases_without_read_access_initialized: list of tenant databases that
have NOT finished their read access initialization
   * -- timestamp: date and time of the event
   * -- all_databases_initialized: true if all tenant databases have finished
the read access initialization
   """
   return 0
```

srSecondaryUnregistered hook method stub

Code Listing 12 shows the srSecondaryUnregistered hook method stub.

```
 def srSecondaryUnregistered(self, parameters, **kwargs):
   """
   Hook description:

   * time of call: when a secondary sends an unregister request to the primary
(e.g. -sr_unregister, -sr_takeover)
   * caller: master node on primary site
   * landscape: called only once on the master node primary site
   * behavior upon failure: error trace is written

   @param parameters: dict of parameters {site_name:string, site_id:int,
reason:string}
```

```
   @type parameters: dict
   @param **kwargs: place holder for later usage (new parameters) to keep the
interface stable
   @type **kwargs: dict
   @return: information about success
   @rtype: int

   * parameters:
   * -- site_name: name of the secondary site that requests the unregister
   * -- site_id: internal ID of the secondary site that requests the unregister
   * -- reason: context in which the unregister happens (e.g. "unregister",
"takeover")
   """
   return 0
```

12.1.4.2 Install and Configure a HA/DR Provider Script

You can add, configure, and monitor your custom provider scripts in the SAP HANA studio.

If the HA/DR provider script is created, it can easily be installed on a SAP HANA system by adding a section called [ha_dr_provider_<classname>] to the global.ini with following parameters:

- provider : the class name
- path : location of the script
- execution_order : the ordering of the HA/DR provider if there is more than one; this is a number between 1 and 99

An example is shown in Code Listing 9.

Sample Code

Code Listing 9

```
[ha_dr_provider_myfirsthook]
provider = myFirstHook
path = /hana/shared/myHooks
execution_order = 50
```

It is possible to specify multiple HA/DR providers by adding multiple sections.

All scripts are loaded during the start up phase of the name server, alternatively, to avoid the need for a restart, run the following command to reload the scripts immediately:

```
hdbnsutil -reloadHADRProviders
```

Additional Configuration Parameters

If the HA/DR provider requires additional configurations parameters, arbitrary key value pairs can be added to the configuration parameter section. An example is shown in Code Listing 10.

Sample Code

Code Listing 10 - HA/DR Provider section and custom configuration parameters

```
    [ha_dr_provider_myfirsthook]
    provider = myFirstHook
    path = /hana/shared/myHooks
    execution_order = 50
    myparameter1 = somevalue
    myparameter2 = 42
```

To consume these parameters, the configuration parameter wrapper HADRBaseConfiguration (initialized in base class of the HA/DR Provider) can be used with following methods:

- self.config.hasKey(<name>)
- self.config.get(<name>)

Sample Code

Code Listing 11 - Using the configuration parameter wrapper

```
    def startup(self, hostname, storage_partition, system_replication_mode,
**kwargs):
        if self.config.hasKey("myparameter1"):
            self.tracer.debug("param2 is '%s'" %
self.config.get("myparameter2"))
        return 0
```

Access Rights

In many cases, the HA/DR provider needs additional rights granted in order to run operating system command and programs that require root access. Usually those rights are granted by adding a line to the `/etc/sudoers.d/sap_hdb_sudoers` file similar to this:

`<sid>adm ALL= NOPASSWD: /path/command, /path2/command2`

For example: `mmtadm ALL= NOPASSWD: /sbin/arping, /sbin/ip`

Execution Order

The order of execution is defined with the `execution_order` parameter in the `ha_dr_<classname>` section of the `global.ini` file by specifying a number between 1 and 99 – the lower the number, the higher the priority. For example:

Sample Code

Code Listing 12 - Configuration for two HA/DR Providers

```
    [ha_dr_provider_mySTONITH]
    provider = mySTONITH
    path = /hana/shared/myHooks
```

```
execution_order = 50
[ha_dr_provider_vIPMover]
provider = vIPMover
path = /hana/shared/myHooks
execution_order = 51
```

With the execution_order parameter, you can ensure that the mySTONITH provider is always called before the vIPMover provider

Monitoring with M_HA_DR_PROVIDERS

The monitoring view M_HA_DR_PROVIDERS contains all information about installed HA/DR Providers.

SQL | Result

```
select * from m_ha_dr_providers
```

	PROVIDER_NAME	PROVIDER_COMPANY	PROVIDER_DESCRIPTION	PROVIDER_VERSION	PROVIDER_TYPE	PROVIDER_PATH	EXECUTION_ORDER
1	vIPMover	SAP	vIP Mover	1.0	GENERIC	/hana/shared/myHooks	51

Tracing

There are a number of methods provided with the HA/DR provider base class that allow you to implement trace levels:

- self.tracer.debug(<text>)
- self.tracer.info(<text>)
- self.tracer.warning(<text>)
- self.tracer.error(<text>)
- self.tracer.fatal(<text>)

Everything will be traced to the component ha_dr_<classname>, in this example it would be ha_dr_myfirsthook. The default trace level is "info". You can override the level used by setting the parameter ha_dr_<classname>=<level> in the trace section of the global.ini file.

Additionally, the name server itself traces general information about the HA/DR provider calls and return code to the trace component ha_dr_provider. The default trace level is "info" as well.

12.1.4.3 Example HA/DR Provider Implementation

A full example showing how two HA/DR providers can be implemented for a sample landscape consisting of two SAP HANA systems with system replication enabled.

> **Note**
>
> This example does not make any statement or assumption about what kind of hardware and software set up is licensed and if it fulfills production SAP HANA requirements at all. This is only a showcase,

based on virtual machines, which was available during the development of this example and should not be considered for production use. Concepts such as virtual IPs are not always applicable since the network architecture within and across data centers needs to be considered.

The usage of virtual IPs to automatically reconnect to the master host after a failover or a system replication takeover only works if the virtual IP on the failing host is disabled through a controlled shut down. In a split brain situation (network problems separate parts of the landscape) or on host failures it cannot be guaranteed that the virtual IP is unique in the network causing severe routing issues. Therefore, for some use cases virtual hostnames or cluster manager software to control the assignment of virtual IPs might be an alternative.

The context of this example is based on two SAP HANA systems consisting of 16 virtual machines each. Two HA/DR providers are going to be implemented:

- A provider that sets up a virtual IP address on the master host of the primary system allowing all clients to use always the same IP address for connecting regardless of any HA or DR activities. This HA/DR provider will be called *vIPMover* .
- A provider that runs STONITH in order to ensure proper I/O fencing. STONITH will be called for host auto-failover (the failed host) and system replication takeover (all three master candidates of the other site). The latter one is usually the task of an external cluster manager, but for this simple example, we use the direct way (which is usually not possible in real data center set ups). This HA/DR provider will be called *mySTONITH* .

vIPMover HA/DR Provider

The purpose of this provider is to set up a virtual IP address every time the active master host moves. This can either happen when host auto-failover occurs or by a system replication takeover. For the failover case, the IP address move will simplify the SQL client connect by just having one IP address/hostname to specify. And for the system replication case, the client is able to find seamlessly the new system. However, proper fencing is a crucial part of moving an IP address around in order to avoid split-brain situations, because two hosts listen on the same address. The solution for this problem in this example will be the mySTONITH HA/DR provider, which will reboot the virtual machine, which has failed.

The HA/DR provider will make use of the Linux operating system commands `/sbin/arping` and `/sbin/ip`, which need to be added to the `/etc/sudoers.d/sap_hdb_sudoers` file for SAP HANA's `<sid>adm` user.

Listing 13 shows the class definition, constructor and the about() method of the vIPMover class.

Sample Code

Code Listing 13 - Class definition, constructor and about() method of the vIPMover class

```
from hdb_ha_dr.client import HADRBase
import os
class vIPMover(HADRBase):
    apiVersion = 1
    def __init__(self, *args, **kwargs):
        super(vIPMover, self).__init__(*args, **kwargs)
        self.vIP = self.config.get("vip")
        self.eth = self.config.get("eth")
        self.netMask = self.config.get("netmask")
    def about(self):
        return {"provider_company" :        "SAP Documentation Example",
```

```
                "provider_description" :    "vIP Mover",
                "provider_version" :        "1.0"}
```

Using apiVersion = 1, the `__init__()` method calls its super method and additionally reads the three attributes vIP, eth and netMask from the configuration file. More details later. The next step is to define helper methods for the actual setup of the virtual IP address. This example uses the standard Linux command `ip` and `arping` to set up and shut down an IP address:

Sample Code

Helper methods for virtual IP address setup and shut down

```
    def setupIP(self):
        # setup IP
        command1 = "sudo /sbin/ip addr add %s/%s dev %s" % (self.vIP,
self.netMask, self.eth)
        rc1 = os.system(command1)
        self.tracer.info("command '%s' returned with rc=%s" % (command1, rc1))
        command2 = "sudo /sbin/arping -U -c 5 %s" % self.vIP
        rc2 = os.system(command2)
        self.tracer.info("command '%s' returned with rc=%s" % (command2, rc2))
        return rc1 + rc2
    def shutdownIP(self):
        command = "sudo /sbin/ip addr del %s/%s dev %s" % (self.vIP,
self.netMask, self.eth)
        rc = os.system(command)
        self.tracer.info("comand '%s' returned with rc=%s" % (command, rc))
        return rc
```

The commands that will be executed in this example would be:

```
sudo /sbin/ip addr add 10.208.155.179/20 dev eth0
sudo /sbin/arping –U –c 5 10.208.155.179
```

Finally, Listing 15 shows the implementation of the hook methods.

Sample Code

Code Listing 15 - The hook method implementation

```
    def startup(self, hostname, storage_partition, system_replication_mode,
**kwargs):
        self.shutdownIP()
        # only setup vIP on the primary system
        if system_replication_mode not in ["", "primary"]:
            return 0
        # only setup vIP on the master host
        if storage_partition == 1:
        return self.setupIP()

         return 0
    def shutdown(self, hostname, storage_partition, system_replication_mode,
**kwargs):
        if system_replication_mode not in ["", "primary"]:
            return 0
        if storage_partition == 1:
            return self.shutdownIP()
        return 0
    def failover(self, hostname, storage_partition, system_replication_mode,
**kwargs):
        if system_replication_mode not in ["", "primary"]:
            return 0
```

```
        if storage_partition == 1:
            return self.setupIP()
        return 0
    def preTakeover(self, isForce):
        """Pre takeover hook."""
        return self.setupIP()
```

The three methods startup(), shutdown() and failover() have the same structure. The condition `if system_replication_mode not in ["", "primary"]` checks if there is no system replication configured or if it is a system replication primary system.

In the preTakeover() method, the virtual IP address is started just before the internal takeover process begins.

The HA/DR provider is configured in the global.ini with the three user-defined parameters:

Sample Code

Code Listing 16 - The vIPMover configuration in the SAP HANA system

```
[ha_dr_provider_vIPMover]
provider = vIPMover
path = /hana/shared/myHooks
execution_order = 51
vip = 10.208.155.179
eth = eth0
netmask = 20
```

mySTONITH HA/DR Provider

The second HA/DR provider offers STONITH to the SAP HANA system. For this example the virtual machine sends a hard reboot command with a locally installed API to a management node outside the SAP HANA system. As the system is installed with virtual hostnames, but the API requires the public hostnames, a mapping of both is defined in the global.ini. Another option would be to resolve those names via naming convention if applicable.

Listing 17 shows the HA/DR Provider implementation.

Sample Code

Code Listing 17 - Implementation of the mySTONITH HA/DR provider

```
from hdb_ha_dr.client import HADRBase
import os
class mySTONITH(HADRBase):
    apiVersion = 1
    def __init__(self, *args, **kwargs):
        super(mySTONITH, self).__init__(*args, **kwargs)
        self.hsrStonith = self.config.get("hsr_stonith").split()
    def about(self):
        return {"provider_company" :        "SAP Documentation Example",
                "provider_description" :    "Basic virtual machine STONITH",
                "provider_version" :        "1.0"}

    def stonith(self, failingHost, **kwargs):
        vmName = self.config.get("map_%s" % failingHost)
        if vmName == "":
            raise Exception("hostname for virtual machine not configured")
```

```
        self.tracer.info("calling STONITH for %s (%s)" % (vmName,
failingHost))
        return os.system(("<script> <logon credentials> --name %s <hard
reboot> <servicing host>" % (vmName)))

def preTakeover(self, isForce):
        rc = 0
        for h in self.hsrStonith:
            rc = rc + self.stonith(h)
        return rc
```

The __init__() and about() methods are filled similar to the example above. The stonith() method looks up the SAP HANA internal hostname via configuration parameter and executes the STONITH command. This is specific to the type of the virtual environment. For bare metal servers, an IPMI-based call is a typical implementation.

The preTakeover() method sends a STONITH command to all master hosts of the other site defined via configuration parameter hsr_stonith , a space-separated list of host names.

As a result the configuration entries look like this:

Sample Code

```
    [ha_dr_provider_mySTONITH]
    provider = mySTONITH
    path = /hana/shared/myHooks
    execution_order = 50
    hsr_stonith = hananode17 hananode28 hananode32
    map_hananode17 = DEWDFTVU3017
    map_hananode28 = DEWDFTVU3028
    map_hananode32 = DEWDFTVU3032
    map_hananode01 = DEWDFTVU3001
    map_hananode02 = DEWDFTVU3002
    map_hananode03 = DEWDFTVU3003
    map_hananode04 = DEWDFTVU3004
    map_hananode05 = DEWDFTVU3005
    map_hananode06 = DEWDFTVU3006
    map_hananode07 = DEWDFTVU3007
    map_hananode08 = DEWDFTVU3008
    map_hananode09 = DEWDFTVU3009
    map_hananode10 = DEWDFTVU3010
    map_hananode11 = DEWDFTVU3011
    map_hananode12 = DEWDFTVU3012
    map_hananode13 = DEWDFTVU3013
    map_hananode14 = DEWDFTVU3014
    map_hananode15 = DEWDFTVU3015
    map_hananode16 = DEWDFTVU3016
```

12.2 SAP HANA Database Backup and Recovery

SAP HANA offers comprehensive functionality to safeguard your database and ensure that it can be recovered speedily and with maximum business continuity.

SAP HANA supports the following backup and recovery capabilities:

- Manual and automated backups

- Extensive configuration options
- Integrity checks for backups
- Backup lifecycle management (housekeeping)
- Recovery to the most recent state or to a specific point-in-time
- Recovery to a specific data backup or data snapshot
- Database copy using backup and recovery
- Full integration with third-party backup tools
- When using the local secure store (LSS), its backups are automatically integrated into the SAP HANA backup (an extra secure store backup is not needed).

> **Note**
>
> This documentation only covers backup and recovery of an SAP HANA database. It does not describe how to back up and recover all the components that can be part of an SAP system.

This documentation does not cover SAP HANA studio. For information about working with SAP HANA studio, see the separate guide *Backup and Recovery Using SAP HANA Studio*.

Related Information

Creating Backups
Scheduling Backups
Change the Backup Configuration Settings
Checking Whether a Recovery is Possible [page 1030]
Housekeeping: Deleting and Archiving Backups
Recovering an SAP HANA Database
Copying a Database Using Backup and Recovery [page 1065]
Working with Third-Party Backup Tools [page 969]
Backup and Recovery Using SAP HANA Studio

12.2.1 Planning Your Backup and Recovery Strategy

When you are planning a backup strategy, consider using a combination of backup types for different recovery scenarios to minimize the risk of data loss, and to ensure that, in the event of a fault, a recovery can be completed speedily and with minimal downtime.

When to Create a Data Backup

It is recommended that you create a data backup in the following situations:

- After the initial load

- At regular intervals
  You can schedule backups to run at specific times without additional user intervention.

  > → Tip
  >
  > You can use less recent data backups for a recovery, provided that the subsequent log backups are available. If more log backups have to be replayed, the recovery takes longer to complete.
  >
  > For this reason, we recommended that you use the most recent data backup and subsequent log backups to recover the database.
  >
  > The more frequently a database is backed up, the faster the recovery will be.

- Before the database software is upgraded to a new version
  If a software upgrade fails, it is possible to use the backup to recover the database to its state before the upgrade.

  > ⓘ Note
  >
  > After an SAP HANA upgrade, the backup history is not broken. A full backup is not necessary to ensure that the backup history is intact.

- After any situation that causes log writing to be interrupted
  For example, immediately after the log mode was changed.

Scheduling Regular Backups

It is strongly recommended to schedule regular data backups from the data area of your SAP HANA database to a secure location.

A backup plan could look like this:

- Data snapshot: daily
  For more information, see *Points to Note: Backup with Data Snapshots*.

  > ⓘ Note
  >
  > If you have a backup and recovery strategy that is based on **data snapshots**, you must ensure that all the data snapshots that you wish to use for a recovery are replicated outside of the SAP HANA storage system.

- Data backup (in the file system or using third-party backup tools): once a week
- Automatic log backups

For more information about scheduling backups using SAP HANA cockpit, see *Schedule Backups*.

Lifecycle Management (Housekeeping)

It is recommended to regularly delete full backups and backup generations that are no longer needed to recover your SAP HANA database.

For more information, see *Housekeeping: Deleting and Archiving Backups*.

Related Information

Points to Note About Backup and Recovery [page 923]
Points to Note: Backup with Data Snapshots [page 926]
SAP HANA Backup Types [page 941]
SAP HANA Backup [page 983]
Scheduling Backups
Housekeeping: Deleting and Archiving Backups
SAP HANA Recovery [page 1029]
Recovery Scenarios [page 1059]

12.2.2 Points to Note About Backup and Recovery

Before you begin preparing a backup strategy for your SAP HANA installation, you should be aware of specific considerations that apply to backup and recovery.

12.2.2.1 Points to Note: SAP HANA Backups

When you plan your backup strategy, you should be aware of several important points concerning SAP HANA data backups, data snapshots, delta backups, and log backups.

- Backups can only be created when SAP HANA is online.
  All the configured SAP HANA services must be running.
  While full backups (data backups and data snapshots), delta backups (differential and incremental backups), and log backups are being created, the impact on system performance is negligible, and users can continue to work normally.
  For more information, see *SAP HANA Backup Types*.

  > ⚠ Caution
  >
  > Do not create a full backup after a database fault or other failure has occurred.

- With a data backup, only the actual data is backed up.
  Unused space in the database is not backed up.
  A full data backup contains all the data that is required to recover the database to a consistent state. This includes both business data and administrative data.

  > ⓘ Note
  >
  > A full data backup does not include the log area.

- The system database can initiate backups of both the system database itself and of individual tenant databases.

A tenant database can create its own backups without needing to connect through the system database.

- A data backup reflects the consistent database state from the time at which the data backup was started. Changes made to the database after a data backup was started are not included in the data backup. If a data backup is recovered without any log backups, open transactions in the data backup are rolled back to the start time of the data backup.
- If a new full backup (data backup or data snapshot) is started before the previous full backup is finished, SAP HANA handles the situation as follows:
  - The first full backup continues as normal.
  - The second full backup does not start, and an error message is displayed.
- It is not possible to back up and recover individual database objects. Backup and recovery always apply to the whole database. For more information, see *Data Backups*.
- If the local secure store (LSS) is used, creating backups will fail if the root key backup password is not set and not given within the SQL statement.

SAP HANA Dynamic Tiering and SAP HANA Backup

If you are planning a backup and recovery strategy for a landscape that makes use of SAP HANA dynamic tiering, see *SAP Note 2375865 (SAP HANA Dynamic Tiering 2.0: Backup and Restore Functional Restrictions)* for information about considerations for dynamic tiering.

> **Note**
>
> `hdbbackupcheck` does not support SAP HANA Dynamic Tiering.

Worker Groups

If you have defined worker group sub-roles, information about the worker groups for each volume is stored as part of SAP HANA full backups (complete data backups and data snapshots).

SAP HANA Cockpit and Backup

From SAP HANA cockpit 2.0, to schedule backups for an SAP HANA 1.0 database, you must be logged onto that database. You cannot schedule backups for SAP HANA 1.0 databases through the system database.

Related Information

SAP HANA Backup Types [page 941]
Data Backups [page 942]

Tenant Databases [page 17]
Scheduling Backups (SAP HANA Administration with SAP HANA Cockpit)
SAP Note 2375865

12.2.2.1.1 Points to Note: File-Based Backups

When you plan your backup strategy, you should be aware of several important points concerning how SAP HANA handles backups to the file system.

- The configured destination for data and log backups must be valid throughout the whole system, not only for specific hosts.
  Backups of tenant databases are always created in subdirectories of this location.
- To make the backup area available to all the nodes in a database, it is strongly recommended to use shared backup storage.
  In addition, shared storage offers support for database copy.

> ⓘ Note
>
> **Shared Backup Storage**
>
> Shared backup storage allows the system database or the master index server of a tenant database to perform availability checks for file-based backups at the beginning of the recovery.
>
> If you are working with file-based backups, and shared storage is not used for backups, the master name server has no access to the backup storage of the other servers. As a consequence, the master name server cannot check whether backups are available. This means that the availability checks cannot be performed at the beginning of the recovery. If you have started a recovery that cannot be completed because one or more of the required backups are not available, this will only be detected later, when each service checks the availability of its own backups.
>
> If the complete recovery needs to be repeated because log backups or delta backups are missing, this may cause significant disruption to work with the database.

Related Information

Change the Backup Configuration Settings (SAP HANA Administration with SAP HANA Cockpit)
Parameters for Data Backup Settings [page 985]
Naming Conventions [page 963]
Parameters for Backing Up the Backup Catalog [page 1018]

12.2.2.1.2 Points to Note: Backup with Data Snapshots

If you are planning a backup strategy that makes use of data snapshots, you should be aware of several important points.

- You can create a data snapshot of an SAP HANA system (the system database and all its tenant databases).
- To **create** a data snapshot, you need to use native SQL.
  For more information, see *Create a Data Snapshot (Native SQL)*.
- If your backup and recovery strategy is based on data snapshots, you must ensure that all the data snapshots that you wish to use for a recovery are replicated outside of the SAP HANA storage system.
- Data snapshots using Backint are not supported by SAP HANA.
  If you are using a third-party backup tool, check with your vendor whether storage-level snapshot technologies are supported.

Related Information

Data Snapshots [page 945]
Create a Data Snapshot (Native SQL) (SAP HANA Administration with SAP HANA Cockpit)
Comparison of Data Backups and Data Snapshots [page 947]
Points to Note: Recovery with Data Snapshots [page 932]

12.2.2.1.3 Points to Note: Log Modes

SAP HANA uses two log modes: `normal` and `overwrite`. By default, SAP HANA runs in log mode `normal`.

After installation, SAP HANA temporarily runs in log mode `overwrite`.

In log mode `overwrite`, the log segments are not backed up. When a log segment is full, it is closed. After a successful savepoint, that log segment can be reused. In this way, log mode `overwrite` ensures that the log area does not grow excessively.

After you create the first full data backup, SAP HANA automatically switches to the default log mode `normal`, and log backups are created.

> → Tip
>
> If you change the log mode from `overwrite` – where log backups are not created – to log mode `normal`, **you must create a full data backup** to ensure that log backups are created again, and that the database can be recovered to the most recent point in time.

SAP HANA Dynamic Tiering and Log Mode Overwrite

If you are planning a backup and recovery strategy for a landscape that makes use of SAP HANA dynamic tiering, see *SAP Note 2375865 (SAP HANA Dynamic Tiering 2.0: Backup and Restore Functional Restrictions)* for information about considerations for dynamic tiering.

Related Information

Log Modes [page 1009]
Change Log Modes [page 1011]
SAP Note 2375865

12.2.2.1.4 Points to Note: Backup Compression

You can create compressed data backups, delta backups (differential or incremental backups), and log backups using native SAP HANA compression.

Backup Compression with SAP HANA Cockpit

With SAP HANA cockpit, you can enable compression of data backups and log backups and define the degree of compression.

When you create or schedule backups using SAP HANA cockpit, you can specify whether to compress the backup. The degree of compression for backups can be changed in the SAP HANA cockpit configuration.

For more information, see *Backup Configuration Settings* in *SAP HANA Administration with SAP HANA Cockpit (Backup and Recovery)*.

Backup Compression Parameters

You can use SAP HANA parameters to configure data backup and log backup compression. By default, backup compression is disabled.

For more information, see *Configure Data Backup Compression* and *Configure Log Backup Compression*.

Backup Compression and the Backup Catalog

If backup compression is enabled, the backup catalog records the original and compressed sizes.

The backup catalog itself cannot be compressed.

For more information, see *M_BACKUP_CATALOG_FILES System View* in the *SAP HANA SQL Reference Guide for SAP HANA Platform* and *What Information is in the Backup Catalog?*.

Backup Compression with Native SQL

To create compressed backups, you can use the BACKUP DATA statement with the option COMPRESSED.

To recover compressed backups, no additional action is required, as the RECOVER DATA and RECOVER DATABASE statements automatically recognize if a backup is compressed.

For more information, see *BACKUP DATA Statement (Backup and Recovery)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Related Information

Backup Configuration Settings (SAP HANA Administration with SAP HANA Cockpit)
Create Data Backups (SAP HANA Administration with SAP HANA Cockpit)
Configure Data Backup Compression [page 991]
Configure Log Backup Compression [page 1017]
What Information is in the Backup Catalog? [page 957]
BACKUP DATA Statement (Backup and Recovery)
RECOVER DATA Statement (Backup and Recovery)
RECOVER DATABASE Statement (Backup and Recovery)
M_BACKUP_CATALOG_FILES System View

12.2.2.1.5 Points to Note: Third-Party Backup Tools

Third-party backup tools can be fully integrated with SAP HANA to enable you to perform backup and recovery operations.

- The implementation of the API of a third-party backup tool that uses the `Backint for SAP HANA` interface must be certified by SAP.
  For more information, see *Working with Third-Party Backup Tools*.
- To recover a database, it is possible to use a combination of backups from a third-party backup tool and backups from the file system, provided that the backups originate from the same SAP HANA database.
- SAP HANA supports high isolation scenarios for third-party backup tools.
  For more information, see *Isolation Level High for Backups and Third-Party Backup Tools*.

SAP HANA Dynamic Tiering and Third-Party Backup Tools

If you are planning a backup and recovery strategy for a landscape that makes use of SAP HANA dynamic tiering, see *SAP Note 2375865 (SAP HANA Dynamic Tiering 2.0: Backup and Restore Functional Restrictions)* for information about considerations for dynamic tiering.

Related Information

Working with Third-Party Backup Tools [page 969]
Copying a Database Using Backup and Recovery [page 1065]
Isolation Level High for Backups and Third-Party Backup Tools [page 979]
SAP Note 2031547
SAP Note 2375865

12.2.2.1.6 Points to Note: Release Compatibility of SAP HANA Backups

In some situations, backups from earlier SAP HANA releases can be used for a recovery.

- SAP HANA backups created with release 1.0 SPS10 or newer can be used to recover or copy to SAP HANA 2.0.
  This applies to both SAP HANA single-container systems and tenant databases.
  For more information about using backups from SAP HANA 1.x to recover or copy to SAP HANA 2.x releases, see *SAP Note 2372809 (Mandatory Preparation Steps for Upgrading a SAP HANA 1 System to SAP HANA 2)*.
  For SAP HANA running on IBM Power systems, different release compatibilities apply.
  For more information, see *Points to Note: SAP HANA on IBM Power Systems*.
- A backup of an SAP HANA single-container system can only be recovered to a tenant database.
  A backup of an SAP HANA single-container system **cannot be recovered to a system database**.

Related Information

SAP Note 2372809

12.2.2.2 Points to Note: SAP HANA Recovery

Before you plan your backup and recovery strategy, you should be aware of several important points with regard to recovering an SAP HANA database.

- The SAP HANA version used for the recovery must always be the same or newer than the SAP HANA version used to create the data and log backup.
  Data and log backups cannot be used to recover to a lower SAP HANA version.
  If you are upgrade your SAP HANA from a maintenance revision to a support package, certain restrictions apply. For more information, see SAP Note 1948334 (*SAP HANA Database Update Paths for Maintenance Revisions*) and SAP Note 2378962 (*SAP HANA 2.0 Revision and Maintenance Strategy*).
- A system database only needs to be recovered if it is corrupted.
  If only a tenant database is corrupted, the system database does not need to be recovered.
- To perform a recovery, an SAP HANA database needs to be shut down.
  For this reason, during recovery, a database cannot be accessed by end users or applications.

The **system database** is shut down for recovery	All its tenant databases are automatically shut down as well. The whole SAP HANA system is not available until the recovery of the system database has been completed. **Note** When you recover and restart a system database, its tenant databases are not automatically restarted. You should first check that the system database was recovered successfully, then restart the tenant databases manually.
A **tenant database** is shut down for recovery	The system database and any other tenant databases remain online.

Note

Recovery from a data snapshot:

Before you make the data snapshot available in the data area of the storage system, you must first shut down the database.

- To recover a complete SAP HANA system:
  A recovery of a tenant database is always initiated from the system database.
  You first need to recover the system database. After the system database has been recovered successfully, each tenant database is recovered separately. The tenant databases cannot be recovered together in one single operation.
- Before a SAP HANA is recovered, all target volumes are deleted and recreated with the recovered data.
  When you recover a tenant database that is being encrypted on the fly, recovery ensures that there are no unencrypted shadow pages in the recovered database. For this reason, it is not necessary to drop and recreate a tenant database before starting the recovery.
- Using SAP HANA cockpit, a **tenant database** can be recovered from a full database backup only or to a user-specified point in time.
  Using SAP HANA cockpit, a **system database** can only be recovered to its most recent state.

To recover a system database to a point in time, use the `recoverSys.py` tool.
For more information, see *Recover a System Database to a Point in Time*.

- If using the local secure store (LSS), you need the root keys backup password for a successful SAP HANA database recovery.

SAP HANA Dynamic Tiering

If you are planning a backup and recovery strategy for a landscape that makes use of SAP HANA dynamic tiering, see *SAP Note 2375865 (SAP HANA Dynamic Tiering 2.0: Backup and Restore Functional Restrictions)* for information about considerations for dynamic tiering.

Worker Groups

If you have defined worker group sub-roles, information about the worker groups for each volume is stored as part of SAP HANA full backups (complete data backups and data snapshots).

Ensure that the target system for a database recovery has the same number of worker groups as the source system.

> ⚠ Caution
>
> If the target system is not configured correctly, SAP HANA cannot be recovered.

Before you start a recovery, you should normally ensure that the worker groups in the SAP HANA system and the backups have the same names. However, under certain circumstances, it is possible to override this restriction.

For more information, see the `<IGNORE WORKERGROUPS>` option for *RECOVER DATABASE Statement (Backup and Recovery)* in the SAP HANA SQL Reference Guide.

SAP HANA Cockpit and Recovery

SAP HANA cockpit 2.0 SP05 can be used to recover only SAP HANA 2.0 databases.

For more information, see *SAP Note 2616241 (Recovery of SAP HANA 1.0 with SAP HANA Cockpit 2.0)*.

Related Information

SAP HANA Recovery [page 1029]
Recover a System Database to a Point in Time (recoverSys.py) [page 1053]
Checking Whether a Recovery is Possible [page 1030]
RECOVER DATABASE Statement (Backup and Recovery)

SAP Note 1948334
SAP Note 2378962
SAP Note 2375865
SAP Note 2616241

12.2.2.2.1 Points to Note: Delta Backups and Recovery

SAP HANA supports both differential and incremental backups.

- You can recover an SAP HANA database using a full data backup (complete data backup or data snapshot) and a combination of a differential backup and one or more incremental backups.
- By default, when SAP HANA computes a recovery strategy, it gives preference to differential and incremental backups over log backups.
  To recover using only a full data backup and log backups, specify the appropriate options in the recovery dialog.
- If you recover an SAP HANA database, and do not immediately create a full data backup, the delta backups subsequently created are based on the data backup that was used for the recovery.

> **Note**
>
> If you wish to recover SAP HANA using differential or incremental backups, you must also use log backups. If log backups are not available, you can only recover using a full data backup.

Related Information

Delta Backups [page 943]

12.2.2.2.2 Points to Note: Recovery with Data Snapshots

Data snapshots offer an additional option to safeguard and recover an SAP HANA database. If you are planning a backup strategy that makes use of data snapshots, you should be aware of several important points.

> **Caution**
>
> Unlike a database recovery from data backups and log backups, the process of recovering SAP HANA from a data snapshot is not entirely under the control of SAP HANA. Manual steps are necessary, for example, to restore the data snapshot to the correct location before the recovery can begin.
>
> To ensure that it succeeds, recovering SAP HANA from a data snapshot requires expert knowledge.
>
> If you are in doubt as to how to ensure that the prerequisites for a recovery from a data snapshot are met, we recommend that you instead consider recovering SAP HANA using a standard recovery from data backups and log backups.

- Using SAP HANA cockpit and storage system tools, you can recover a complete SAP HANA system (the system database and all its tenant databases) from a data snapshot.
- To recover SAP HANA from a data snapshot, you need to recover the system database, and then recover each tenant database separately.
  The tenant databases cannot be recovered together in one single operation.
- To recover a database from a data snapshot, you can optionally also use delta backups and log backups.
- While a data snapshot is being created, no further data integrity checks are performed (checksum calculation) on page or block level.
  If a data snapshot has been changed before it is used for a recovery, SAP HANA will only detect this when the database is started.
- It is **not possible** to use a data snapshot of an SAP HANA single-container system to recover an SAP HANA multitenant database system.

Related Information

Data Snapshots [page 945]
Recover SAP HANA From a Data Snapshot (SAP HANA Administration with SAP HANA Cockpit)

12.2.2.2.3 Points to Note: License Keys and Recovery

When you recover an SAP HANA database, you should be aware of certain license key requirements.

The license key for an SAP HANA database is based on the system ID and the hardware ID. After a recovery, if the SID or hardware ID has changed, an SAP HANA license key becomes invalid.

SAP HANA licenses can be installed for the system database (global) or for a single tenant database (local). Global licenses are for the system database and all the tenant databases, but a license installed in a tenant database governs only that tenant database. If a local license key is removed, that tenant database reverts to the global license key installed in the system database.

During recovery, a temporary license key is installed automatically if the backup used for recovery had a permanent license that is still valid. You can work with the automatically installed temporary license for up to 90 days. During this time, you need to apply to SAP to have the license from the source database transferred to a new license key. You then need to install the new license key in the recovered SAP HANA database.

For more information, see *License Keys for SAP HANA Database* in *SAP HANA Administration Guide (Licensing)*.

Lockdown

If SAP HANA is in lockdown immediately after a recovery or a database copy, this can be because:

- The SID or hardware key of the target SAP HANA system does not match the SID or hardware key of the source SAP HANA system, and
- The backup contains an invalid license key.

This could be a permanent license key that expired more than 28 days ago (grace period), a temporary license key that expired, or a temporary license key that was already auto-generated during a previous recovery.

If the system is in lockdown, you need to request and install a new license key for the current SID or hardware key.

If the backup has a valid permanent license key, a new temporary license key is generated automatically when the target system is restarted after recovery, even if the SID and/or the hardware key have changed.

Related Information

License Keys for SAP HANA Database [page 137]

Recovering an SAP HANA Database (SAP HANA Administration with SAP HANA Cockpit)

Prerequisites for Copying a Database Using Backup and Recovery (SAP HANA Administration with SAP HANA Cockpit)

12.2.2.2.4 Points to Note: Extension Nodes for SAP Business Warehouse

If you are using extension nodes for Business Warehouse, you need to consider some important points with regard to SAP HANA recovery.

Data Temperature

Ensure that warm data is recovered to a service that is configured for warm data.

Likewise, hot data must be recovered to a service that is configured for hot data.

This means that you need to set up the same number and type of services for hot and warm data on the target host that were running on the source host.

For more information, see *Extension Node* in *SAP HANA Administration Guide for SAP HANA Platform*, and SAP Note 2453736 (*How-To: Configuring SAP HANA for SAP BW Extension Node in SAP HANA 2.0*).

Worker Groups

If you have defined worker group sub-roles, information about the worker groups for each volume is stored as part of SAP HANA full backups (complete data backups and data snapshots).

Ensure that the target system for a database recovery has the same number of worker groups as the source system.

> ⚠ Caution
>
> If the target system is not configured correctly, SAP HANA cannot be recovered.

Before you start a recovery, you should normally ensure that the worker groups in the SAP HANA system and the backups have the same names. However, under certain circumstances, it is possible to override this restriction.

For more information, see the `<IGNORE WORKERGROUPS>` option for *RECOVER DATABASE Statement (Backup and Recovery)* in the SAP HANA SQL Reference Guide.

Related Information

Extension Node
SAP Note 2453736
RECOVER DATABASE Statement (Backup and Recovery)

12.2.2.3 Points to Note: Copying a Database Using Backup and Recovery

You can use backup and recovery to copy a tenant database to the same or a different SAP HANA system, or to copy a system database to a different SAP HANA system. When you copy an SAP HANA database, you should be aware of certain important considerations.

You can copy an SAP HANA database using file-based backups or backups created using third-party tools.

> ⓘ Note
>
> To copy a database, it is also possible to use a backup catalog from a different source than the backups:
>
> - Backups from a third-party backup tool with a backup catalog from the file system
> - Backups from the file system with a backup catalog from a third-party backup tool

> ⓘ Note
>
> To create a database copy using differential or incremental backups, you must also use log backups. If log backups are not available, you can only create a database copy using a full data backup.

Database Copy and Data Snapshots

- You can create a data snapshot of an SAP HANA database with one or more tenant databases.
- For a database copy using data snapshots, the number of hosts and the number and type of services assigned to each host must be the same for the source database and the target database

Also, the mountpoint IDs must be identical.

- When you copy SAP HANA from a data snapshot, you first need to recover the system database. After the system database has been recovered successfully, each tenant database is recovered separately.
  The tenant databases cannot be recovered together in one single operation.
- Before you make the data snapshot available in the data area of the storage system, you must first shut down the database.

Database Copy and System Replication

If you have system replication configured, and require near-zero downtime, consider using system replication to copy a tenant database.

For more information, see *Copying and Moving Tenant Databases Between Systems* in the *SAP HANA Administration Guide*.

Related Information

Copying a Database Using Backup and Recovery [page 1065]
Copying and Moving Tenant Databases
Copying a Database Using Backup and Recovery (SAP HANA Administration with SAP HANA Cockpit)

12.2.2.4 Points to Note: System Replication

Data backups and log backups can only be written on the primary system.

After a Takeover

> **Note**
>
> After a takeover, it is **not** necessary to create a new full data backup (data backup or data snapshot) of the now active system. Backups of the former primary system can be used to recover the database, but **not** in combination with delta backups from the current primary.
>
> Delta backups must be based on the complete data backup of the current primary. After a takeover, if you wish to create delta backups, you must first have created a complete data backup of the current primary.
>
> If you wish to recover SAP HANA using differential or incremental backups, you must also use log backups. If log backups are not available, you can only recover using a full data backup.

After a takeover, ensure the following:

- Backups from the former primary system are not being written to the same location as backups from the now active system.

  > ⚠ Caution
  >
  > If backups from different systems are mixed, it will not be possible to recover the database.

  This can be achieved in either of the following ways:
  - Disable automatic log backups and any scheduled data backups in the former primary system.
  - Shut down the former primary system to ensure that it creates no new data backups and no new log backups.
- Any backups scheduled in the now active system are configured in accordance with your requirements. For more information, see *Schedule Backups*.

Before a Recovery

Before a recovery, disable the FULL SYNC option.

If you are running system replication with replication mode `SYNC` and the `FULL SYNC` option enabled, the system will not start after a recovery, because no write operations are possible.

To prevent this from happening, before you perform a recovery, manually disable the `FULL SYNC` option in `global.ini`.

You can use the following command as `<sid>adm`:

```
hdbnsutil -sr_fullsync --disable
```

For more information, see SAP Note 2165547 (*FAQ: SAP HANA Database Backup & Recovery in an SAP HANA System Replication Landscape*) and *Recovery with System Replication*.

System Replication and Third-Party Backup Tools

- If backups are managed using a third-party tool, the `Backint for SAP HANA` API must be accessed by both the active system and the original primary system.
- If SAP HANA is recovered from backups that were created with different UIDs, some third-party backup tools may prevent the recovery from being started.
  For more information, contact your tool vendor or ensure that the same UID is used for all the backups used for a recovery.

Related Information

Scheduling Backups

Recovery with System Replication [page 1062]
SAP Note 2165547

12.2.3 Authorizations Needed for Backup and Recovery

Backup and recovery operations can only be performed by users that have the appropriate authorizations. The authorization required depends on whether administrative tasks are performed at system level or at database level.

> → Tip
>
> We recommend that you create your own dedicated database users with only the specific authorizations required for backup and recovery.

The following authorizations are required to administer SAP HANA:

Backup and Recovery (SAP HANA Cockpit)

Authorizations for Backup and Recovery (SAP HANA Cockpit)

To Perform This Task...	You Need These SAP HANA System Privileges...		
	System Database	Tenant Database	Tenant Database (Through the System Database)
View, create, and cancel database backups	`BACKUP ADMIN` or `BACKUP OPERATOR` (recommended for batch users only)	`BACKUP ADMIN` or `BACKUP OPERATOR` (recommended for batch users only)	`DATABASE BACKUP OPERATOR` or `DATABASE BACKUP ADMIN` or `DATABASE ADMIN`
Delete database backups and backups of the backup catalog	`BACKUP ADMIN`	`BACKUP ADMIN`	`DATABASE BACKUP ADMIN` or `DATABASE ADMIN`
Recover or copy a database	Operating system user `<sid>adm`	(not possible)	`DATABASE RECOVERY OPERATOR` or `DATABASE ADMIN`

To Perform This Task...	You Need These SAP HANA System Privileges...		
	System Database	Tenant Database	Tenant Database (Through the System Database)
Schedule backups	BACKUP ADMIN or BACKUP OPERATOR	BACKUP ADMIN or BACKUP OPERATOR	DATABASE BACKUP OPERATOR or DATABASE BACKUP ADMIN or DATABASE ADMIN
Configure backups	Display the backup configuration settings: BACKUP ADMIN	Display the backup configuration settings: BACKUP ADMIN	DATABASE BACKUP ADMIN or DATABASE ADMIN
Define a Backup Retention Policy Change the status for Retained backups	BACKUP ADMIN	BACKUP ADMIN	DATABASE BACKUP ADMIN or DATABASE ADMIN

System-wide Backup Configuration

To change default configuration settings for all the tenant databases, you need DATABASE ADMIN.

Native SQL

Authorizations for Backup and Recovery (Native SQL)

To Perform This Task...	You Need These SAP HANA System Privileges...		
	System Database	Tenant Database	Tenant Database (Through the System Database)
View, create, and cancel database backups	BACKUP ADMIN or BACKUP OPERATOR (recommended for batch users only)	BACKUP ADMIN or BACKUP OPERATOR (recommended for batch users only)	DATABASE BACKUP OPERATOR DATABASE BACKUP ADMIN DATABASE ADMIN
Check that the disk space required for a backup is available.	BACKUP OPERATOR, BACKUP ADMIN		DATABASE BACKUP OPERATOR, DATABASE BACKUP ADMIN

To Perform This Task...	You Need These SAP HANA System Privileges...		
	System Database	Tenant Database	Tenant Database (Through the System Database)
Recover or copy a database		(not possible)	`DATABASE RECOVERY OPERATOR` `DATABASE ADMIN`
Check whether backups are accessible			`DATABASE RECOVERY OPERATOR`
Create data snapshots	`BACKUP OPERATOR`, `BACKUP ADMIN`	(not possible)	(not possible)
Delete database backups and backups of the backup catalog	`BACKUP ADMIN`	`BACKUP ADMIN`	`DATABASE BACKUP ADMIN` `DATABASE ADMIN`
Change the status for Retained backups	`BACKUP ADMIN`	`BACKUP ADMIN`	`DATABASE BACKUP ADMIN` or `DATABASE ADMIN`

Backup Encryption

Authorizations for Backup and Recovery (Backup Encryption)

	System Database	Tenant Database	Tenant Database (Through the System Database)
Enable or Disable Backup Encryption (To encrypt backups, no additional authorization is required.)	`ENCRYPTION ROOT KEY ADMIN`	`ENCRYPTION ROOT KEY ADMIN`	`ENCRYPTION ROOT KEY ADMIN` **Note** Encryption for a tenant database can only be enabled or disabled from the system database if the system database has control.

For more information about backup encryption, see *Enable Encryption of Data and Log Backups* in the *SAP HANA Administration Guide (Encryption)* and *ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)* in the *SAP HANA SQL and System Views Reference*.

BACKUP ADMIN Versus BACKUP OPERATOR

The system privileges `BACKUP ADMIN` and `BACKUP OPERATOR` allow you to implement a more specific separation of user roles.

System Privilege	Permitted Operations
`BACKUP ADMIN`	All backup-related operations, including deleting backups and backup configuration.
`BACKUP OPERATOR`	• Create backups • Cancel backups • Check available space • Query views

For more information about SAP HANA authorization, see *SAP HANA Authorization* in the *SAP HANA Security Guide* and *GRANT Statement (Access Control)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Related Information

Operating System User <sid>adm [page 552]
SAP HANA Authorization
SAP HANA Backup Encryption [page 950]
Enable Encryption of Data and Log Backups
GRANT Statement (Access Control)

12.2.4 SAP HANA Backup Types

SAP HANA supports the following backup types:

- Full backups:
  - Complete data backups
  - Data snapshots

    > **Note**
    >
    > To make use of storage snapshot-based SAP HANA backups, first create an SAP HANA data snapshot.
    >
    > For more information, see *Data Snapshots and Database Snapshots*
- Delta backups (incremental backups and differential backups)
- Redo log backups
- Native encryption of backups

- Backups created using third-party backup tools

Related Information

Data Backups [page 942]
Delta Backups [page 943]
Data Snapshots [page 945]
Data Snapshots and Database Snapshots [page 946]
Log Backups [page 948]
SAP HANA Backup Encryption [page 950]
Working with Third-Party Backup Tools [page 969]
Backup Catalog [page 956]
Backing Up Customer-Specific Configuration Settings [page 994]
Naming Conventions [page 963]
Diagnosis Files for Backup and Recovery [page 1070]

12.2.4.1 Data Backups

A data backup includes all the data that is required to recover the database to a consistent state.

A data backup is comprised of both business data and administrative data. Administrative data can be roles, models, information models, and topology information.

With a data backup, only the actual data is backed up. A data backup does not include:

- Unused space in the database
- The log area

The data area is backed up in parallel for each of the SAP HANA services. If SAP HANA is running on multiple hosts, a data backup includes all the service-specific backup parts for all the hosts.

While a data backup is running, some data integrity checks are performed. These integrity checks are performed on block level (page level on disk) only, and do not analyze the content of each data block.

If these checks are successful, the data is written to the backup destination.

> **Note**
>
> To ensure the safety of your data, data backups should be stored on multiple different backup destinations outside the SAP HANA database.

Related Information

Creating Backups (SAP HANA Administration with SAP HANA Cockpit)

Backing Up Customer-Specific Configuration Settings [page 994]
Comparison of Data Backups and Data Snapshots [page 947]

12.2.4.2 Delta Backups

Delta backups contain only the data that has been changed since the last full data backup (complete data backup or data snapshot) or the last delta backup.

> ⓘ Note
>
> Delta backups can only be created after a data backup has been created.

With a delta backup, changed data means changes to the physical representation of the data in the SAP HANA persistent storage. This is not always the data that was actually changed by an application. An internal reorganization can change the physical representation **without changing the actual data**.

> ✤ Example
>
> In a delta merge of a column store partition, only a small amount of the data may have been changed. Nevertheless, all the data is restructured and rewritten. This means that a delta merge can be as large as a full data backup. If a delta backup is created in this situation, the whole partition is backed up in the delta data backup, even if only a small amount of the actual data was changed.

Benefits of Delta Backups

Delta backups allow you to reduce the amount of data that is backed up, compared to full data backups.

In turn, this means that delta backups are normally faster to create than full data backups.

In addition, a database recovery using delta backups is normally faster than a recovery using log backups. With delta backups, only the changed data is recovered, whereas with log backups, each log entry needs to be processed separately before it is recovered. Recovering many log backups is normally more CPU-intensive than recovering a small number of delta backups.

> → Tip
>
> To keep data that is frequently changed separate from data that is not frequently changed, consider partitioning column store data. Partitioning column store data can reduce the size of delta backups.

Related Information

Delta Backup Types [page 944]

12.2.4.2.1 Delta Backup Types

SAP HANA supports both differential backups and incremental backups.

> **Note**
>
> Delta backups can only be created after a data backup has been created.

Comparison of Delta Backup Types

	Differential Backup	Incremental Backup
What Data is Backed Up?	The data changed since the last full data backup (complete data backup or data snapshot).	The data changed since the last full data backup or the last delta backup (incremental or differential).
Backup Size	The amount of data to be saved with each differential backup **increases**.	If data remains unchanged, it is not saved to more than one backup. For this reason, incremental backups are the smallest of the backup types.
Backup and Recovery Strategy	If your backup strategy is based on only full data backups and differential backups, only two backups are needed for a recovery: one full data backup and one differential backup.	If your backup strategy is based on only full data backups and incremental backups, to recover the database, SAP HANA needs the following backups: • The full data backup on which the incremental backups are based • Each incremental backup made since the full data backup In some situations, many incremental backups may be needed for a recovery.

Recovery Using Delta Backups

> **Note**
>
> A recovery can use multiple incremental backups, but only one differential backup.
>
> If you wish to recover SAP HANA using differential or incremental backups, you must also use log backups. If log backups are not available, you can only recover using a full data backup.
>
> **System replication:** Delta backups must be based on the complete data backup of the current primary. After a takeover, if you wish to create delta backups, you must first have created a complete data backup of the current primary.

To recover SAP HANA, you can combine a differential backup with one or more incremental backups.

> **Example**
>
> You could recover SAP HANA to a specific point in time using the following sequence of backups:
>
> 1. Full data backup
> 2. Differential backup
> 3. Incremental backup 1
> 4. Incremental backup 2
> 5. Log backups

Related Information

Recovering an SAP HANA Database
Recovery with System Replication [page 1062]

12.2.4.3 Data Snapshots

A data snapshot captures the data persisted in the data area at a particular point in time. A data snapshot includes all the data that is required to recover each individual SAP HANA database to a consistent state.

Benefits of Data Snapshots

Data snapshots offer an additional option to safeguard the SAP HANA data area and to recover an SAP HANA database.

Data snapshots have the following benefits:

- Data snapshots can be created with minimal impact on database performance.
  Data snapshots are created in the storage system and do not consume database resources.
- Recovery from a data snapshot is faster than a recovery from a data backup.
  The data snapshot only needs to be made available in the data area of the storage system.

For more information about the relative benefits of data snapshots, see *Comparison of Data Backups and Data Snapshots*.

> **Note**
>
> To make use of storage snapshot-based SAP HANA backups, first create an SAP HANA data snapshot.
>
> For more information, see *Data Snapshots and Database Snapshots*

Data Snapshots and External Storage Systems

The external storage system must copy each data volume in an atomic operation in order to ensure the I/O consistency of the data snapshot. Multiple data volumes do not need to be copied in parallel; data volumes can be copied one at a time.

Data Snapshots and SAP HANA Dynamic Tiering

If you are planning a backup and recovery strategy for a landscape that makes use of SAP HANA dynamic tiering, see *SAP Note 2375865 (SAP HANA Dynamic Tiering 2.0: Backup and Restore Functional Restrictions)* for information about considerations for dynamic tiering.

Related Information

Data Snapshots and Database Snapshots [page 946]
Comparison of Data Backups and Data Snapshots [page 947]
Create a Data Snapshot (Native SQL)
Encryption of Data Snapshots [page 951]
Recover SAP HANA From a Data Snapshot
SAP Note 2375865

12.2.4.3.1 Data Snapshots and Database Snapshots

A **data snapshot** is created by first creating an **internal database snapshot** of the file system of the data area. The internal database snapshot is then used to create a data snapshot.

The internal database snapshot captures the state of the SAP HANA database at the point in time that the snapshot was started. In this way, the internal database snapshot ensures that the state of the data snapshot is consistent.

> **Note**
>
> While a **data snapshot** is being created, no further data integrity checks are performed (checksum calculation) on page or block level.
>
> (With **data backups**, the block-level integrity of the data to be backed up is checked automatically while the backups are being created. The content of each data block is not checked.)

Internal Database Snapshot and System Replication

An internal database snapshot used to create a data snapshot does not conflict with an internal database snapshot used for system replication. There is no relation between these two types of internal database snapshots.

Related Information

Create a Data Snapshot (Native SQL)
Encryption of Data Snapshots [page 951]

12.2.4.3.2 Comparison of Data Backups and Data Snapshots

You can use this overview to help assess the benefits of using data backups and data snapshots as part of your backup strategy.

Comparison of Data Backups and Data Snapshots

	Data Backup to File	Data Backup Using Backint	Data Snapshot
Advantages	• Integrity checks at block level For more information, see *Manually Checking Whether a Recovery is Possible*. • Can be encrypted For more information, see *SAP HANA Backup Encryption*.	• Integrity checks at block level • Integrated into existing data center infrastructure • Third-party backup tool offers additional features. For example, deduplication. • Backups are immediately available for recovery	• Fast • Negligible impact on network • Can be encrypted For more information, see *Encryption of Data Snapshots*.
Disadvantages	• Requires additional storage • Generates additional network load • File system needs to be monitored (fill level) • More time is needed to make backups available for recovery	• Generates additional network load	• No integrity checks on page or block level (Checksum calculation)

	Data Backup to File	Data Backup Using Backint	Data Snapshot
Backup Size	• Payload only	• Payload only	• Size of the data area (but is usually compressed or deduplicated by the storage tool)
Backup Duration	• IO-bound (reading from data volume, writing to target) • Network-bound (writing to target file system)	• IO-bound (reading from data volume) • Network-bound (writing to backup server)	• Negligible (depending on the storage tool)

Related Information

Checking Whether a Recovery is Possible [page 1030]
SAP HANA Backup Encryption [page 950]
Encryption of Data Snapshots [page 951]
Working with Third-Party Backup Tools [page 969]

12.2.4.4 Log Backups

By default, SAP HANA log segments are backed up automatically.

Log segments are backed up for each service that has persistence. When a log backup is created, only the actual data (the "payload") of the log segments is written from the log area to service-specific log backups in the file system or to a third-party backup tool.

> ⓘ Note
>
> If you use the local secure store (LSS), root key actions (rotation, key management configuration changes, encryption settings) trigger LSS backup to be written to log area, and later also into log area. This requires the root keys backup password to be set, otherwise the backup will fail.

Interrupted Log Backups

If an SAP HANA service stops, log backups for that service also stop. The stopped service must be immediately restarted.

> ⚠ Caution
>
> If a stopped service is not restarted, a database recovery will only be possible to a point in time before this service stopped. That is, only to a point in time for which log backups for **all services** exist.
>
> If log backups for any service are missing, it will not be possible to recover the database to its most recent state.

Log Area Unavailable

If the log area becomes temporarily unavailable, once it is available again, SAP HANA automatically continues creating log backups for all the log segments that have not so far been backed up.

Removing a Service

If you need to remove a service, use the procedure described in the section *Steps After Copying a Database*, as this ensures that the log area is backed up and can be used to recover the database.

Related Information

Log Modes [page 1009]
Naming Conventions for Log Backups [page 968]
Parameters for Log Backup Settings [page 999]
Troubleshooting: Slow Log Backups [page 1006]

12.2.4.4.1 Savepoints and Redo Logs

To maintain optimal performance, an SAP HANA database holds the bulk of its data in memory. However, SAP HANA also uses persistent storage to provide a fallback in the event of a fault or a failure.

During normal database operation, changed data is automatically saved from memory to disk at regular **savepoints**. By default, savepoints are created every five minutes, including during a backup.

With a system running on properly configured hardware, the impact on performance of savepoints is negligible. Savepoints do not affect the processing of transactions. During a savepoint, transactions continue to run as normal, and new transactions can be started as normal.

Additionally, all data changes are recorded in the **log segments in the redo log buffer**. When a database transaction is committed, these log segments are saved to disk. Also, if the redo log buffer fills at any time, the redo log buffer is written to disk anyway, even if no commit has been sent.

Related Information

Persistent Data Storage in the SAP HANA Database [page 180]
Database Restart and Savepoints [page 950]
M_SAVEPOINTS System View

12.2.4.4.2 Database Restart and Savepoints

An SAP HANA database can be restarted in the same way as a disk-based database, and returned to its most recent consistent state by replaying the redo logs from the log area (not the log backups).

The log records in the log area only need to be processed from the last savepoint position, rather than from the beginning of the log area. In this way, savepoints help to speed up database restarts.

While savepoints and logs can protect your data against some failures, these mechanisms offer no protection if the persistent storage itself is damaged or if a logical error occurs. To be able to react appropriately and quickly to a hardware failure, as well as to protect your data against logical errors and the possibility of corruption caused by software changes, it is essential to have a well-planned strategy for backup and recovery.

Related Information

Planning Your Backup and Recovery Strategy [page 921]
Savepoints and Redo Logs [page 949]

12.2.4.5 SAP HANA Backup Encryption

SAP HANA supports native encryption of backups.

Backup encryption safeguards the privacy of the SAP HANA backup data by preventing unauthorized parties from reading the content of backups.

When backup encryption is enabled, the backup data is transferred encrypted to the backup location, both for file-based backups and for backups created using third-party backup tools.

SAP HANA backups are encrypted using AES 256-bit encryption.

Which Backup Types Can Be Encrypted?

It is not possible to enable encryption for an individual data backup. If backup encryption is enabled, all backups are encrypted, except the backup catalog, which is never encrypted.

> ⓘ Note
>
> To encrypt data snapshots, additional steps are necessary.
>
> For more information, see *Encryption of Data Snapshots* in *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.

Related Information

M_ENCRYPTION_OVERVIEW System View
Encryption of Data Snapshots [page 951]
Prerequisites: Recovering an Encrypted SAP HANA Database
Change the I/O Buffer Size [page 989]
Set Parallel Backup Encryption [page 990]
Configure Multistreaming with Third-Party Backup Tools [page 976]
Data Storage Security in SAP HANA
Encryption Key Management
Root Key Backup
Changing Encryption Root Keys
Import Backed-Up Root Keys or LSS Backup Before Database Recovery
Enable Encryption
Disable Encryption
M_ENCRYPTION_OVERVIEW System View

12.2.4.5.1 Encryption of Data Snapshots

If backup encryption is enabled, data snapshots are not automatically encrypted. To encrypt data snapshots, additional steps are necessary.

A data snapshot can be encrypted if the following conditions are met:

- Backup encryption is enabled.
- Log encryption is enabled.
- Data volume encryption is enabled.
  For more information, see the *SAP HANA Administration Guide (Encryption)* and the *SAP HANA Security Guide*.
- The data conversion status is **not active**.
  The SAP HANA database must not be in the process of encrypting or decrypting data.
  To check the data conversion status, execute the following SQL statement:
  `SELECT DATA_CONVERSION_ACTIVE FROM M_PERSISTENCE_ENCRYPTION_STATUS`
  To encrypt a data snapshot, `DATA_CONVERSION_ACTIVE` must be set to `false`.
  For more information, see *M_PERSISTENCE_ENCRYPTION_STATUS System View* in the SAP HANA SQL Reference Guide for SAP HANA Platform.

Related Information

SAP HANA Backup Encryption [page 950]
M_PERSISTENCE_ENCRYPTION_STATUS System View

12.2.4.5.2 Resource Consumption With Backup Encryption

With backup encryption enabled, SAP HANA backups require additional main memory and CPU. Depending on how much main memory and CPU are available, you can offset an impact on backup performance by adjusting buffer sizes, enabling parallel encryption, or multistreaming (for third-party backup tools).

Typically, backup encryption does not impact the performance of backup and recovery, since the backup and recovery operations, such as encryption, checksum calculation, and I/O, run in parallel. However, with backup encryption, higher CPU usage is to be expected. Depending on the sizing of your SAP HANA system, you may notice a negative impact on overall system performance.

To create encrypted backups with optimal performance, SAP HANA consumes up to three times as much main memory and CPU than is needed to create unencrypted backups. Resource consumption is compared below:

Backup Type	Buffer Allocated
Unencrypted backups	The memory consumed by backup is determined by the configured I/O buffer size to read/write backups. For an unencrypted data backup, this amount of memory is allocated twice, one buffer for reading data and one buffer for writing data. This is the case when multistreaming is not used. To configure I/O buffer size, you can use the parameter `data_backup_buffer_size` in the `backup` section of the `global.ini` parameter file. For more information, see *Change the I/O Buffer Size*.
Multistreaming backups	The configured I/O buffer size is also allocated to each channel. With multistreamed backups, SAP HANA allocates one additional I/O buffer for each additional channel. For this reason, the memory consumption is calculated as `data_backup_buffer_size` * (1 + `parallel_data_backup_backint_channels`). To configure multistreaming backups, you can use the parameter `parallel_data_backup_backint_channels` in the `backup` section of the `global.ini` parameter file. For more information, see *Configure Multistreaming with Third-Party Backup Tools*.

Backup Type	Buffer Allocated
Encrypted backups	The configured I/O buffer size is also allocated to two additional I/O buffers for each channel for each persistent service. For this reason, the memory consumption is calculated as `data_backup_buffer_size` * (1 + 3 * `parallel_data_backup_backint_channels`) **Note** The size of encrypted SAP HANA backups is the same as unencrypted backups (except for the checksum).

Considerations for Backup Encryption

Depending on how much main memory and CPU are available on an SAP HANA system, consider the following options:

- Configure SAP HANA to run backup (and recovery) with sub-optimal performance.
  You can configure SAP HANA to perform backup and recovery operations with encryption enabled, but with the same resource consumption as without encryption. As a consequence, backup and recovery operations will typically take longer to complete.
  Set the parameter `enable_parallel_backup_encryption` from **`true`** (default) to **`false`**.
- Install additional CPUs in your SAP HANA system.
- Reduce the backup I/O buffer size allocated by SAP HANA.
  If the I/O buffer size is reduced, backup will typically have a lower throughput, and, as a consequence,will take longer to complete.
  If no additional physical main memory is available, the backup I/O buffer size for a database can be reduced to approximately one third of the backup I/O buffer size configured for unencrypted backups.
  For more information, see *Change the I/O Buffer Size* in the *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.
- If you are working with a third-party backup tool, it is not recommended to use data deduplication with encrypted backups, as this will increase backup times with no actual benefit.

Calculating Backup Memory Consumption

The following examples illustrate how SAP HANA allocates main memory for data backup.

Example

Backup without Multistreaming

Criteria	Configuration	Memory Allocated for Data Backup
Configuration parameter setting:	`parallel_data_backup_backint_channels = 1`	

Criteria	Configuration	Memory Allocated for Data Backup
Backup encryption enabled?	No	1024 MB + 1024 MB = **2 GB**
Backup encryption enabled?	Yes	1024 MB + (3 * 1024 MB) = **4 GB**

Example

Backup with Multistreaming

Criteria	Configuration	Memory Allocated for Data Backup
Configuration parameter setting:	`parallel_data_backup_backint_channels = 32`	
Backup encryption enabled?	No	1024 MB + (32 * 1024 MB) = **33 GB**
Backup encryption enabled?	Yes	1024 MB + (3 * 32 * 1024 MB) = **97 GB**

In this scenario, SAP HANA allocates approximately three times as much main memory with backup encryption enabled, compared with backup encryption disabled.

Related Information

SAP Note 1999997
Change the Backup Configuration Settings
Change the I/O Buffer Size [page 989]
Configure Multistreaming with Third-Party Backup Tools [page 976]

12.2.4.5.3 Backup Encryption Root Key

If you enable encryption (either for backup, log, or data volume), the SAP HANA backups are encrypted and decrypted using a backup encryption root key.

The same backup encryption root key is used for data backups and log backups.

Note

Data snapshots are not encrypted using the backup encryption root key.

For more information, see *Encryption of Data Snapshots* in *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.

The backup encryption root key is encrypted and stored in the secure store in the file system (instance SSFS) together with other encryption root keys. For example, application encryption root keys.

When a new tenant database is created, a new backup encryption root key is generated for that tenant database.

Other tenant databases retain their existing backup encryption root key.

> ⚠ Caution
>
> If backup encryption is enabled, database administrators must ensure that the current backup encryption root key and older versions are backed up.
>
> Whenever the backup encryption root key is changed, you must back it up. Without a current backup of the backup encryption root key, some data will be lost after a recovery.
>
> There is always one active backup encryption root key. If (repeated) root key rotation is used, previously active keys can also exist, and may be needed to recovery SAP HANA from older encrypted backups.
>
> For more information, see *Root Key Backup* in the *SAP HANA Security Guide*.

> ⓘ Note
>
> The block-level integrity of encrypted backups can still be checked without access to the backup encryption root key.
>
> For more information, see *Checking Whether a Recovery is Possible* in the *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.

For more information about working with encryption root keys, see *Encryption Key Management* in the *SAP HANA Security Guide* and *Changing Encryption Root Keys* in the *SAP HANA Administration Guide (Encryption)*.

Related Information

Checking Whether a Recovery is Possible [page 1030]
Points to Note: License Keys and Recovery [page 933]
Backup Encryption
Encryption Key Management
Changing Encryption Root Keys
Encryption of Data Snapshots [page 951]
Root Key Backup

12.2.4.5.4 SAP HANA Dynamic Tiering and Backup Encryption

If you are planning a backup and recovery strategy for a landscape that makes use of SAP HANA dynamic tiering, see *SAP Note 2375865 (SAP HANA Dynamic Tiering 2.0: Backup and Restore Functional Restrictions)* for information about considerations for dynamic tiering.

Related Information

SAP Note 2375865

12.2.4.6 Backup Catalog

The backup catalog contains information about the backup history of an SAP HANA database.

The backup catalog enables SAP HANA to determine the following:

- Whether a recovery is possible
- Which backups to use to recover a database
- Which backups are no longer needed for a recovery

The system database and each tenant database have their own backup catalog.

> **Note**
>
> Each time a backup of any type is created, the backup catalog is backed up and versioned. In this way, the latest version of the backup catalog always contains the entire backup history.

Recovery Without the Backup Catalog

It is possible to recover SAP HANA without using a backup catalog, or using data backups that are not recorded in the backup catalog.

Without a backup catalog, a point-in-time recovery cannot be done. Without a backup catalog, it is only possible to recover SAP HANA to a specific data backup.

If you recover SAP HANA using a backup that is not recorded in the backup catalog, you need to manually specify the backup type (File or Backint), the location of the backup, and its prefix.

Related Information

Display Information in the Backup Catalog
What Information is in the Backup Catalog? [page 957]
Monitoring Views for the Backup Catalog [page 960]

Rebuilding the Backup Catalog [page 962]
Housekeeping: Deleting and Archiving Backups
Backup Configuration Settings
Naming Conventions for the Backup Catalog [page 969]

12.2.4.6.1 What Information is in the Backup Catalog?

The backup catalog contains information about the backup history.

In the Backup Catalog	Description
Backups created for a database	This includes data backups, data snapshots, delta backups (differential and incremental backups), and log backups. Each recovery is recorded in the backup catalog, but not displayed in the monitoring views. For more information, see *Monitoring Views for the Backup Catalog*.
Start and completion times of the backups	The backup catalog records local server times and UTC times. The start and end times of a recovery are recorded in the backup catalog as local server time. Points in time for a recovery in the SQL statements are specified as UTC. The start time of a backup does not reflect the exact state of the database when the backup was created. The savepoint that determines the database state is always taken after the start time of a data backup. Changes made to the database after this savepoint are not included in a data backup.
Whether a backup is still running	The backup catalog does not show the progress of a backup. The progress of a backup is recorded in the `backup.log`, and can be seen using the view `M_BACKUP_PROGRESS`. For more information, see *backup.log* and *Monitoring Views for the Backup Catalog*.

In the Backup Catalog	Description
Status of a backup	Records whether a backup was completed successfully or not. The status can be: • Successful • Failed • Running • Cancel pending • Canceled • Prepared For data snapshots only: A data snapshot has been prepared, but has not been confirmed or abandoned.
Volumes that were backed up	Backups are created in separate backup destinations for each volume.
Log backups and the log positions they contain	For data backups, one exact redo log position is recorded. This redo log position corresponds to its savepoint. During a recovery, log replay starts at the redo position for the data backup. For log backups, a range of redo log positions is recorded, from the oldest log entry to the newest log entry.
Backup sizes	The size of each backup is shown. If backup compression is enabled, the compressed size is also shown.
Destination type	Data backups can be written to different destinations. However, all the parts of the same data backup are written to the same destination. The destination type can be: • File • Backint • Snapshot For data backups created in the file system, you can change the default destination. For backups created using third-party backup tools, a named pipe is created in the file system for the system database or tenant database. The default backup destination cannot be changed.

In the Backup Catalog	Description
Backup ID	ID of a data backup or a log backup. All backup files of a single data backup share the same BACKUP_ID. If you are working with a third-party backup tool, an external backup ID (`ebid`) is also included.
Encryption Root Key	If encryption is enabled, the key used for encrypting a backup.
Comment	Additional user-supplied information. A comment can help to identify a particular backup in the backup catalog.
Configuration Included	For data backups and delta backups: Shows whether customer-specific configuration changes (.ini files changed from the default values) are included in the backup. The *Details* view also shows which of the files for a backup contains the configuration data. For more information, see *Include Configuration Settings in Backups*.

In the Backup Catalog	Description
Retained	For complete data backups: Shows whether the backup is flagged to be retained. Individual complete data backups that are flagged as Retained cannot be deleted in SAP HANA, either individually (by deleting them from the backup catalog overview), by scheduled housekeeping tasks, or by the SQL statement `BACKUP CATALOG DELETE`. Retained data backups remain recorded in the backup catalog and in their physical locations. You can change the Retained setting from the *Backup Details* view. **Retained Backups and Backup Generations** After a **backup generation** has been deleted, a Retained data backup still exists in its physical location, and is still recorded in the backup catalog. However, the Retained data backup may no longer be at the beginning of a new backup generation. There may be no longer be any related subsequent log backups, delta backups, or backups of the backup catalog. For more information, see *Delete Backup Generations* in *SAP HANA Administration with SAP HANA Cockpit (Backup and Recovery)*.

Related Information

Monitoring Views for the Backup Catalog [page 960]
backup.log [page 1071]
Savepoints and Redo Logs [page 949]
Include Configuration Settings in Backups [page 996]
Data Snapshots [page 945]
Delete Backup Generations

12.2.4.6.2 Monitoring Views for the Backup Catalog

You can use monitoring views to display information from the backup catalog. Monitoring views are stored in the SYS schema.

The monitoring views `M_BACKUP_CATALOG`, `M_BACKUP_CATALOG_FILES`, and `M_BACKUP_PROGRESS` provide different overviews of information from the backup catalog.

Monitoring View	Description
M_BACKUP_CATALOG	Provides an overview of information about backup activities. Each row in the view provides information about a separate catalog entry identified by a backup ID. This information includes the type (for example, data backup), and start and completion times. **Note** The backup ID is used to reference the parts of a backup in the M_BACKUP_CATALOG_FILES monitoring view.
M_BACKUP_CATALOG_FILES	Provides information about the backups created, and the backup destinations for data and log backups. Each row in the view has a corresponding entry in the M_BACKUP_CATALOG monitoring view. Each row is identified by a backup ID. The M_BACKUP_CATALOG_FILES monitoring view provides additional information about each database service that was involved in a backup. For example, with a data backup, each database service is listed with its specific backup information such as destination path and redo log position.
M_BACKUP_PROGRESS	Provides detailed information about the most recent data backup. Each row contains information about one service that is part of the data backup, identified by host name and port number.

Comparison of the Monitoring Views for the Backup Catalog

M_BACKUP_CATALOG	M_BACKUP_CATALOG_FILES	M_BACKUP_PROGRESS
All types of backups (data backup, log backup, delta backups, and data snapshots, if available)		Only for data backups, differential backups, and incremental backups
All completed and currently running backups since the database was created		Currently running and last finished backups only
Persistent		Cleared at database restart
Total amount of data for completed backups only		Total and already transferred amount of data for all backups

For more information, see the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Example: Search for a Data Snapshot Using M_BACKUP_CATALOG

To search for a data snapshot in the backup catalog, you can use either the backup ID or the comment.

To search for a backup ID, use the following command:

```
SELECT * FROM "SYS"."M_BACKUP_CATALOG" WHERE BACKUP_ID = backup_id
```

Related Information

M_BACKUP_PROGRESS System View
M_BACKUP_CATALOG System View
M_BACKUP_CATALOG_FILES System View

12.2.4.6.3 Rebuilding the Backup Catalog

In exceptional situations that are outside of the control of SAP HANA, the backup catalog may not be available at the time of a recovery. If the backup catalog is not available, it can be largely rebuilt using the existing data and log backups from the file system.

To rebuild the backup catalog, use the `hdbbackupdiag` tool.

For more information, see SAP Note 1812057 (*Reconstruction of the backup catalog using hdbbackupdiag*).

> ⚠ Caution
>
> If the backup catalog is rebuilt:
>
> - The backup catalog only contains information about the data backups and log backups that you provide through the `hdbbackupdiag` tool.
>   Any parts of the database backup history that you do not specify are lost.
> - The backup catalog no longer contains data snapshots or backups created using third-party tools.
>   As a consequence, those backups cannot be used for a recovery.
> - Data and log backup directories must contain **only SAP HANA data**.
> - A log is generated and written to the SAP HANA working directories.
>   If you rebuild the backup catalog **more than once**, this log will also be read, and an error will be caused. For this reason, if you need to rebuild the backup catalog a second time, remove the log from the first rebuild.
> - Only the backup of the backup catalog is affected; the persistent SAP HANA backup catalog is not changed.
>   The backup of the backup catalog is used to recover SAP HANA.

Related Information

SAP Note 1812057
Checking Whether a Recovery is Possible [page 1030]
Housekeeping: Deleting and Archiving Backups

12.2.4.7 Naming Conventions

When you plan your backup strategy, you need to be familiar with the naming conventions and recommendations for file-based data backups, delta backups, and third-party backup tools.

Related Information

Naming Conventions for Data Backups [page 963]
Naming Conventions for Delta Backups [page 967]
Naming Conventions for Log Backups [page 968]
Naming Conventions for the Backup Catalog [page 969]
Temporary Names for File-Based Backups [page 969]

12.2.4.7.1 Naming Conventions for Data Backups

When you plan your backup strategy, you need to be familiar with the naming conventions and recommendations for data backups.

Each data backup name is comprised of the following elements:

```
<path><prefix>_<suffix>
```

> **Note**
>
> The naming conventions apply to data backups created in the file system and data backups created using third-party tools. With third-party tools, you cannot change the backup path.

Elements of Data Backup Names

Name Element	Description
`<path>` For example: `/backup/data/`	Optional. **For file-based backups**: If no complete path is specified, the default backup destination is prepended to the backup name. **For backups created using third-party tools**: A named pipe is created in the file system for the system database or tenant database: • `/usr/sap/<SID>/SYS/global/hdb/backint/SYSTEM` • `/usr/sap/<SID>/SYS/global/hdb/backint/DB_<tenant_database_name>` A third-party backup tool reads data to be backed up from the named pipe, and writes the backup data in accordance with the tool configuration.

Name Element	Description
`<prefix>`	Optional. You can use the prefix proposed by the system or you can specify a different prefix for the backup name. **→ Tip** To be able to more easily identify archived file-based backups, it is strongly recommended that you use a unique prefix for each data backup name. For example, a timestamp. By default, the name of each backup is prefixed with the timestamp of the start of the backup. The placeholders **`[date]`** and **`[time]`** are automatically converted to the current timestamp. If you use the same fixed prefixes, it is recommended that you replicate a data backup to a new destination as soon as the backup is created. Otherwise, if your SAP HANA system is configured to overwrite existing data backups, an existing backup will be overwritten by the next data backup with the same prefix. For backups created using **third-party tools**, data backups are not overwritten. The Backint for SAP HANA interface is able to identify multiple versions of backups with the same name. Nevertheless, for easier identification and versioning, it is strongly recommended to assign unique backup names. **ⓘ Note** It is **not** possible to change the prefix of a backup after it has been created.
`<Suffix>`	The system adds a suffix to each backup name. For complete data backups, the suffix is unique only within a specific backup. For delta backups and log backups, the suffix is unique for a specific database. As a suffix is assigned for each service that is included in the backup, you only need to specify the name (`<path><prefix>`) for all the backups on all the hosts in the system. The next time a service is backed up, the system assigns the same suffix to the backup to that service.

> **Note**
>
> Once backups have been created, it is strongly recommended that you **do not change** their names.
>
> When backups are created, their names are stored in the backup catalog. For a recovery, specific backup components are located using the names stored in the backup catalog. If the name of a backup was changed after it was recorded in the backup catalog, it will not be possible to locate it using the backup catalog, and it will not be possible to use it for a point-in-time recovery. A renamed backup can only be used to recover the database without replaying log backups (RECOVER DATA).
>
> You can copy or move file-based backups to a different location. If you use a moved backup for a recovery, you need to specify its current location in the recovery dialog.

Example: Names for Parts of a Data Backup

During backup, each service backs up its data to the specified backup destination.

Below is an example of a set of backups from one data backup.

```
/backup/data/COMPLETE_DATA_BACKUP_databackup_0_1
/backup/data/COMPLETE_DATA_BACKUP_databackup_1_1
/backup/data/COMPLETE_DATA_BACKUP_databackup_2_1
...
```

In the above example:

Element	In Example
`<path>`	`/backup/data/`
`<prefix>`	`COMPLETE_DATA_BACKUP`
`<suffix>`	`databackup_0_1` The suffix is automatically added by SAP HANA. `0` is the volume ID. `1` is the partition ID.

12.2.4.7.2 Naming Conventions for Delta Backups

When you plan your backup strategy, you need to be familiar with the naming conventions and recommendations for differential and incremental backups.

Structure of File Names for Delta Backups

	Differential Backups	Incremental Backups
Prefix	User-defined. A timestamp is recommended. For example: `2022-02-23`	User-defined. A timestamp is recommended. For example: `2022-02-23`
String	`databackup_differential`	`databackup_incremental`
Backup ID	The backup ID of the **full data backup** on which the differential backup is based	The backup ID of the **full data backup or the delta backup** on which the incremental backup is based
Delta Backup ID	ID of the differential backup	ID of the incremental backup
Volume ID	Volume ID as with complete data backups	Volume ID as with complete data backups
Partition ID	Partition ID as with complete data backups	Partition ID as with complete data backups

Example: File Names for Differential Backups

The SQL statement `BACKUP DATA DIFFERENTIAL USING FILE ('2022-02-23')` creates a differential backup based on the previously created full data backup.

Example names of incremental backup files:

```
2022-02-23_databackup_differential_1426237023821_1426237780534_0_1
2022-02-23_databackup_differential_1426237023821_1426237780534_1_1
2022-02-23_databackup_differential_1426237023821_1426237780534_2_1
2022-02-23_databackup_differential_1426237023821_1426237780534_3_1
```

Example: File Names for Incremental Backups

The SQL statement `BACKUP DATA INCREMENTAL USING FILE ('2022-02-23')` creates an incremental backup based on the previously created full data backup or differential backup.

Example names of incremental backup files:

```
2022-02-23_databackup_incremental_1426237023821_1426237028496_0_1
2022-02-23_databackup_incremental_1426237023821_1426237028496_1_1
2022-02-23_databackup_incremental_1426237023821_1426237028496_2_1
2022-02-23_databackup_incremental_1426237023821_1426237028496_3_1
```

12.2.4.7.3 Naming Conventions for Log Backups

Log backup names are generated automatically. Unlike the data backup names, no parts of the log backup names are user-defined.

The names of log backups are assigned in accordance with specific naming conventions.

Each log backup name comprises the following elements:

```
<log_backup>_<volume ID>_<log partition ID>_<first redo log position>_<last redo log position>.<backup_ID>
```

The elements of log backup names are separated by an underscore. A period ('.') separates the appended backup ID from the log name.

Elements of Log Backup Names

Name Element	Description
`<log_backup>`	All log backups begin with the string `<log_backup>`.
`<volume ID>`	The volume ID for the SAP HANA service. For example, name server, index server, script server, or XS engine.
`<log partition ID>`	Only one log partition is supported for each service.
`<first redo log position>`	The oldest entry in log backup
`<last redo log position>`	The most recent entry in the log backup
`<backup_ID>`	Uniquely identifies the log backup **Note** `<backup_ID>` is calculated automatically by SAP HANA. `<backup_ID>` is only used for file-based backups, not for backups with third-party backup tools.

Example

A log backup name could look like this:

`log_backup_1_0_1234567_1238567.1380740407446`

12.2.4.7.4 Naming Conventions for the Backup Catalog

Different names are assigned to the backup catalog with file-based backups and when using a third-party backup tool.

The backup catalog is assigned a name in the following format:

`log_backup_0_0_0_0.<Backup_ID>`

With a third-party tool, the backup catalog is assigned a name in the following format:

`log_backup_0_0_0_0`

12.2.4.7.5 Temporary Names for File-Based Backups

When file-based backups are written, they are first written using a temporary name.

After a part of a backup has been written successfully, it is renamed to the final name used for the SAP HANA service. If your SAP HANA system is configured to overwrite existing data backups, data backups with the same name are only overwritten after the backup for the service was completed successfully.

> ⓘ Note
>
> If existing backups are overwritten by backups with the same names, at least **twice the space** in the backup location is needed, because the old backup and the new backup exist for a time in parallel.

12.2.5 Working with Third-Party Backup Tools

Third-party backup tools can be fully integrated with SAP HANA to enable you to perform backup and recovery operations from SAP HANA cockpit and using native SQL.

Backint for SAP HANA Interface

In addition to the file system, you can back up and recover an SAP HANA database using an SAP-certified third-party tool that supports the `Backint for SAP HANA interface`, which is used to communicate with an SAP HANA database.

Each active host in a distributed SAP HANA system may have one or more volumes to be backed up. When `Backint for SAP HANA` is used to back up a database, several communication processes are started, one for each volume. Backint-based data backups and log backups can be created in parallel.

A third-party backup tool reads data to be backed up from named pipes, and writes the data in accordance with the tool configuration. For a third-party backup tool, the only objects created in the file system are named pipes. Named pipes occupy no space in the file system.

Prerequisites for Using Third-Party Backup Tools

- You have a support contract with the tool vendor that permits you to use the third-party backup tool with SAP HANA.

More information:

- *SAP Note 1730932 (Using Backup Tools with Backint for SAP HANA)*
- For a current overview of certified third-party backup tools, got to the *SAP Certified Solutions Directory*. Use the search term `HANA-BRINT` and select a partner name to display more details.
- *SAP Note 1730998 (Unrecommended versions of backup tools)* contains a list of backup tool versions that should **not** be installed or activated in an SAP HANA appliance.
- For more information about installing and configuring a third-party backup tool, consult the documentation provided by the tool vendor.

Related Information

SAP Note 2031547
SAP Note 1730932
SAP Note 1730998
SAP Certified Solutions Directory

12.2.5.1 Configuring a Third-Party Backup Tool

After a third-party backup tool has been installed, you can back up and recover an SAP HANA database without making any further changes to the default configuration. However, you have the option to specify a Backint parameter file, and to change some of the tool configuration settings.

Directories for Third-Party Backup Tools

For third-party backup tools, the following directories are used for all types of backups (including delta backups and backups of the backup catalog):

- System database: `/usr/sap/<SID>/SYS/global/hdb/backint/SYSTEMDB`
- Tenant database: `/usr/sap/<SID>/SYS/global/hdb/backint/DB_<tenant_database_name>`

The content of the backups is not necessarily written to these directories. The backup tool decides where the content of the backups is written to.

A third-party backup tool reads data to be backed up from named pipes, and writes the data in accordance with the tool configuration. For a third-party backup tool, the only objects created in the file system are named pipes. Named pipes occupy no space in the file system.

> ⓘ Note
>
> For third-party backup tools, the backup destinations cannot be changed. During a recovery, SAP HANA retrieves the backup content from the locations provided by the third-party backup tool.

Backint Parameter File

If required by the third-party backup tool, you can specify Backint parameter files for data backup, log backups, and the backup catalog. The content and syntax of the parameter files is tool-specific and defined by the tool vendor.

For more information, see the vendor documentation for the third-party backup tool.

To specify Backint parameter files for SAP HANA, use SAP HANA cockpit.

For more information, see *Change the Backup Configuration Settings*.

SAP HANA Parameters for a Third-Party Backup Tool

SAP HANA offers several parameters for configuring a third-party backup tool.

For more information, see *Configure SAP HANA Parameters for a Third-Party Backup Tool*

Related Information

Change the Backup Configuration Settings
Configure SAP HANA Parameters for a Third-Party Backup Tool [page 972]
Database Configuration in SAP HANA Cockpit

12.2.5.1.1 Configure SAP HANA Parameters for a Third-Party Backup Tool

SAP HANA offers several parameters for configuring a third-party backup tool.

Prerequisites

The Backint agent has been installed.

If the Backint agent is not installed, you cannot change the Backint parameter files.

SAP HANA expects the Backint agent executable (`hdbbackint`) to be in the following path:

`/usr/sap/<SID>/SYS/global/hdb/opt/hdbbackint`

If the Backint agent executable is not installed in this path, a symbolic link must be created during the installation of a third-party backup tool. This symbolic link points from `/usr/sap/<SID>/SYS/global/hdb/opt/hdbbackint` to the actual location of the Backint agent executable.

> **Note**
>
> You cannot change the `Backint` agent using SAP HANA cockpit.

Procedure

You can display and change configuration parameters using SAP HANA cockpit.

For more information, see *Configuring System Properties in SAP HANA Cockpit*.

To display all the parameters for third-party backup tools, search for `backint` from the system properties overview.

The following parameters are available for third-party backup tools:

Option	Parameter
Specify a timeout for log backups for third-party tools.	`backint_response_timeout` For more information, see *Timeout for Log Backups (Backint)*.
Write backups of the backup catalog using a third-party backup tool.	`catalog_backup_using_backint` For more information, see *Destination for Backups of the Backup Catalog*.
Enable or disable log backup using a third-party tool.	`log_backup_using_backint` For more information, see *Change the Log Backup Destination Type*.
Specify the number of channels to be used for multistreaming.	`parallel_data_backup_backint_channels` This parameter applies to complete data backups and delta backups. For more information, see *Configure Multistreaming with Third-Party Backup Tools* and *Prerequisites: Recovery Using Multistreamed Backups*.

Specify the Backint protocol version used by SAP HANA to communicate with the third-party backup tools.	`backint_protocol_version`
	Depending on the Backint protocol version, SAP HANA uses different parameters and flags. This parameter must be set to the protocol version supported by the third-party backup tool.
	Default:
	`backint_protocol_version = 1.0`
	Possible values: **`1.0`**, **`1.5`**

Related Information

Timeout for Log Backups (Backint) [page 978]
Destination for Backups of the Backup Catalog [page 1020]
Change the Log Backup Destination Type [page 1015]
Configure Multistreaming with Third-Party Backup Tools [page 976]
Backup Configuration Settings

12.2.5.1.2 Automatic Log Backup Fallback for Third-Party Backup Tools

If an external storage system unexpectedly becomes unavailable, SAP HANA can continue writing log backups. SAP HANA can automatically detect whether an external storage system can no longer be accessed, and then redirect log backups to the file system instead.

When using a third-party backup tool to back up SAP HANA log segments, if the external storage system cannot be accessed, log backups will fail repeatedly. For example, access to an external storage system could be lost as a result of impacted network response times, an outage due to planned maintenance, or hardware failure.

In high-load scenarios, this can quickly cause the log area to grow until the file system is full. If, as a consequence, no more log segments can be created, this will cause SAP HANA to freeze.

You can enable log backup fallback to automatically detect downtimes in a third-party backup tool, and to temporarily redirect the writing of log backups to the file system. If the backup catalog is also configured to be backed up to the third-party storage system, backups of the backup catalog are also automatically redirected to the file system while log backup fallback is active.

> ⓘ Note
>
> If backups of the backup catalog to an external storage system fail, but log backups continue to work, a fallback is **not** triggered.

Behavior of Automatic Log Backup Fallback

When automatic log backup fallback is enabled, it is enabled for **all** the persistent services in a database.

If one service fails to complete a log backup three times in succession, log backup fallback is triggered for all the services in the database. After a service performs four log backups to the file system, the fallback is temporarily disabled to allow SAP HANA to recheck the availability of the external storage system.

The fallback continues as long as the third-party storage is unavailable. When the external storage system is available again, the fallback ends. From then on, log backups for all services are written to the third-party backup tool again.

Enabling Automatic Log Backup Fallback

You can enable automatic log backup fallback using SAP HANA configuration parameters.

Prerequisites

To enable automatic log backup fallback, you need the SAP HANA system privilege DATABASE ADMIN.

Enable Automatic Log Backup Fallback

To enable automatic log backup fallback for third-party backup tools, the following parameters **must** be configured:

To Enable...	Parameter
Log backup using a third-party tool	`log_backup_using_backint=true`
Automatic log backup fallback	`log_backup_backint_fallback=true`

For more information about changing parameter values, see *Database Configuration in SAP HANA Cockpit*.

Log Backup Fallback Location

If automatic log backup fallback is active, log backups are written to the path in the file system defined by the parameter `basepath_logbackup`. This path must be accessible by SAP HANA.

If the backup catalog is configured to be backed up to a third-party backup tool, if a fallback is active, backups of the backup catalog are written to the path in the file system defined by the parameter `basepath_catalogbackup`.

Log Backup Frequency

The log backup frequency is defined by the following parameters:

To Change...	Parameter
The log backup interval	`log_backup_timeout_s` For more information, see *Change the Log Backup Interval*.
The interval mode for log backups	`log_backup_interval_mode` For more information, see *Set the Interval Mode for Log Backups*.

How Log Backup Fallback is Recorded

Alert for Automatic Log Backup Fallback

When a log backup fallback is triggered, alert 143 is raised. When the fallback is stopped, either automatically when log backups are being written again to the external storage system, or by disabling automatic log backup fallback, alert 143 is canceled.

Identifying Which Log Backups Used Fallback

To check whether a log backup was written to the file system due to a fallback or not, refer to the monitoring view `M_BACKUP_CATALOG_FILES`. Column BACKINT_FALLBACK_USED has the following values:

Value	Indicates...
FALSE	A log backup was written to the file system as configured.
TRUE	A log backup was originally configured to be written to a third-party backup tool, but was written to the file system due to an automatic log backup fallback.

For more information, see *M_BACKUP_CATALOG_FILES System View* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Related Information

Database Configuration in SAP HANA Cockpit
Change the Log Backup Interval [page 1013]
Set the Interval Mode for Log Backups [page 1012]
M_BACKUP_CATALOG_FILES System View
Change the Log Backup Destination [page 1016]
Destination for Backups of the Backup Catalog [page 1020]

12.2.5.1.3 Configure Multistreaming with Third-Party Backup Tools

Multistreaming allows backup data to be distributed in parallel to multiple devices. When creating complete data backups and delta backups, a third-party backup tool can use multiple channels to write the backup data for each service.

Context

By default, multistreaming is disabled, which means that SAP HANA uses a single channel for data backups and delta backups. If required, you can configure SAP HANA to use additional channels. When multiple channels are used, SAP HANA distributes the data equally across the available channels. All the parts of a multistreamed backup are approximately the same size.

Multistreamed data backups are configured individually for each database.

> ⓘ Note
>
> To create multistreamed data backups and delta backups of an SAP HANA database, the third-party backup tool must also be configured to use multiple channels. To get optimal performance, ensure that the third-party backup tool is configured correctly.
>
> For more information about the configuration of the backup tool, consult the vendor documentation.

Procedure

1. To enable multistreaming for data backups and delta backups, configure the parameters `parallel_data_backup_backint_channels` and `parallel_data_backup_backint_size_threshold` for each database.

   The parameters are in the `backup` section of `global.ini`.

   For more information, see *Configuring System Properties in SAP HANA Cockpit*.

2. Specify the appropriate values for each service.

Parameter	Possible Values
`parallel_data_backup_backint_channels`	Specify the number of channels to be used for multi-streaming. You can specify a value between 1 and 32. The maximum number of channels permitted for each service is 32. The default value is **1**. A value of 1 means that data backups with third-party backup tools are created through a single channel. This disables multistreaming. To enable multistreaming, specify a value greater than 1. **Note** Each additional channel requires an additional IO buffer (by default, 512 MB). Ensure that increasing the number of channels does not have a negative impact on memory consumption. If backup encryption is enabled, three additional IO buffers are needed for each channel. For more information, see *SAP HANA Backup Encryption*.
`parallel_data_backup_backint_size_threshold`	Specify the minimum data backup size for multistreaming to be enabled. The value is specified in GB. You can specify a value of 1 GB or more. The default value is **`128 GB`**. The threshold applies to each service individually. If the data backup size for a service is below the threshold, that service will not be multistreamed, even if multistreaming is enabled (`parallel_data_backup_backint_channels` > 1).

3. (Optional) Ensure that all buffers fit into memory.

You can optimize the configuration of the following parameters:

- Reduce the backup buffer size.
  Reconfigure `data_backup_buffer_size`.
  For more information, see *Change the Backup Buffer Size*.
- Reduce the number of channels.
  Reconfigure `parallel_data_backup_backint_channels`.

4. Save.

   The changes take effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
Prerequisites: Recovery Using Multistreamed Backups [page 1049]
Change the I/O Buffer Size [page 989]
SAP HANA Backup Encryption [page 950]

12.2.5.1.4 Timeout for Log Backups (Backint)

If the Backint agent does not respond for a user-specified time when writing log backups, SAP HANA cancels the Backint process.

The timeout for log backups for third-party tools is specified using the following parameter:

```
backint_response_timeout=<seconds>
```

The parameter is set on the tenant database coordinator node in the backup section of the `indexserver.ini` parameter file.

The default timeout is 600 s.

> **Note**
>
> This parameter is used for log backups and backups of the backup catalog.

If the Backint process terminates as the result of a timeout, it may be recorded in the `backint.log` as having terminated with an error.

The following error message is written to the trace file for the service:

```
Backint did not respond for 600 seconds, killing pid
```

Related Information

Set the Interval Mode for Log Backups [page 1012]
SAP Note 2571163

12.2.5.1.5 Isolation Level High for Backups and Third-Party Backup Tools

SAP HANA supports high isolation for third-party backup tools.

In an SAP HANA database, it is necessary to ensure that a tenant database cannot access the backups of other tenant databases. If your third-party backup tool does not support high isolation, you can set up separate Backint parameter files for each tenant database. These Backint parameter files are owned and managed by the operating system user (`<sid>adm`), which has read and write access. The tenant databases have only read access to the Backint parameter file through the tenant-specific group.

For more information about the access permissions required for system and tenant databases, see *Set the Isolation Level to High for Backups with Third-Party Backup Tools*.

> ⓘ Note
>
> To ensure high isolation in an SAP HANA database with many tenant databases, many Backint parameter files may be needed.

> → Tip
>
> Check with your third-party backup tool vendor whether any tool-specific restrictions apply.

Related Information

Set the Isolation Level to High for Backups with Third-Party Backup Tools [page 979]
Increase the System Isolation Level [page 78]
Database Isolation [page 81]
Overview of SAP HANA Security Functions

12.2.5.1.5.1 Set the Isolation Level to High for Backups with Third-Party Backup Tools

By default, an SAP HANA tenant database has isolation level low. You can increase the isolation level to high to ensure that one tenant database cannot access the backups of other tenant databases.

For the System Database

Procedure

1. On operating system level, create a database-specific directory for the Backint parameter files.

The database-specific directory must be owned by the operating system user `<sid>adm` and the group `sapsys`.

2. Assign the access permissions 700 to the directory.

   700 allows user `<sid>adm` read, write, and execute access to the directory; the group has no access permission; others have no access permission.

3. In the database-specific directory, create a Backint parameter file for backups.

   If required, also create a database-specific Backint parameter file for log backups.

   The parameter file must be owned by the operating system user `<sid>adm` and group `sapsys`.

4. Assign the access permissions 600 to the parameter file.

   600 allows user `<sid>adm` read and write access to the file; the group has no access permission; others have no access permission.

5. Use the parameters `data_backup_parameter_file`, `log_backup_parameter_file`, and `catalog_backup_parameter_file` to specify the Backint parameter files.

For Each Tenant Database

Context

To grant the system administrator access to the tenant database backup files and directories, you need to add the `<sid>adm` user to the operating system group of each tenant. For more information, see *File and Directory Permissions with High Isolation* in the *SAP HANA Administration Guide*.

Procedure

1. On operating system level, create a database-specific directory for the Backint parameter files.

   The database-specific directory must be owned by the operating system user `<sid>adm` and by the group of the tenant database.

2. Assign the access permissions 750 to the directory.

   750 allows user `<sid>adm` read, write, and execute access to the directory; the group has read and execute permission; others have no access permission.

3. In each tenant-specific directory, create a tenant-specific Backint parameter file for backups.

   If required, also create a tenant-specific Backint parameter file for log backups and backups of the backup catalog.

   The parameter file must be owned by the operating system user `<sid>adm` and group of the tenant database.

4. Assign the access permissions 640 to the parameter file.

   640 allows user `<sid>adm` read and write access to the file; the group has read permission; others have no access permission.

5. Assign the tenant-specific Backint parameter file(s) in the SAP HANA system.
   a. Locate the parameter `data_backup_parameter_file` in the `backup` section of `global.ini`.

      For more information, see *Database-Specific Configuration Parameters (SAP HANA Administration with SAP HANA Cockpit)*.
   b. Open the change dialog.
   c. Specify the new path to the parameter file.
   d. Save.

      When you save, the change takes effect immediately.

      To change the Backint parameter file for log backups, repeat this procedure for the parameter `log_backup_parameter_file`.

      To change the Backint parameter file for backups of the backup catalog, repeat this procedure for the parameter `catalog_backup_parameter_file`.

   Alternatively, to change the Backint parameter file setting, you can execute the following SQL statement:

   Sample Code

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE',
   '<database_name>') SET ('backup', '<backup_parameter_file>') =
   '<absolute_path_and_name>' WITH RECONFIGURE
   ```

   This statement changes one parameter. If you need to assign a tenant-specific Backint parameter file for both data backups and log backups, you need to execute the statement once for each Backint parameter file.

   Example

   Assume that you want to assign new Backint parameter files for a tenant database called `<TENANT1>` in an SAP HANA database called `<PR2>`.

   With the following statement, you can assign a new parameter file for data backups:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE', 'TENANT1')
   SET ('backup', 'data_backup_parameter_file') = '/usr/sap/PR2/SYS/
   global/hdb/opt/config/DB_TENANT1/PR2_TENANT1_data.utl' WITH RECONFIGURE
   ```

   With the following statement, you can assign a new parameter file for log backups:

   ```
   ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE', 'TENANT1') SET
   ('backup', 'log_backup_parameter_file') = '/usr/sap/PR2/SYS/global/hdb/opt/
   config/DB_TENANT1/PR2_TENANT1_log.utl' WITH RECONFIGURE
   ```

Related Information

Database-Specific Configuration Parameters
Database Configuration in SAP HANA Cockpit
File and Directory Permissions with High Isolation [page 83]

12.2.5.2 Delta Backups and Third-Party Backup Tools

SAP HANA supports seamless integration of SAP-certified third-party backup tools.

Normally, delta backups will work using the default configuration settings. In some situations, additional steps may be required to create delta backups with a third-party backup tool.

If you are using a third-party tool to create delta backups, consider the following points:

Configuring the Backint Agent

Ensure that the Backint agent executable (`hdbbackint`) is configured correctly.

For more information about configuring your third-party backup tool, see *Configuring a Third-Party Backup Tool* and consult the tool vendor documentation.

–l LOG Option

For delta backups, SAP HANA uses the `hdbbackint` level log (option `-l LOG`) in combination with the `hdbbackint` parameter file for data backups. This `hdbbackint` call is sent internally by SAP HANA and is recorded in the `backint.log` file.

> ⚠ Caution
>
> If a third-party tool uses the option `-l LOG` to specify the log backup container, the log backup container could unintentionally be used for delta backups as well as for log backups. This can potentially cause an error situation.
>
> For this reason, we recommend that you set up two dedicated `hdbbackint` parameter files: one for data backups and one for log backups
>
> If you are in doubt, check with your tool vendor for more details **before** you use delta backups as part of your backup strategy.

Related Information

Configuring a Third-Party Backup Tool [page 971]
Configure SAP HANA Parameters for a Third-Party Backup Tool [page 972]

12.2.5.3 Upgrading a Third-Party Backup Tool

When you upgrade third-party backup software, the following procedure is strongly recommended.

1. Before you start a software upgrade in your production system:
   - Test the upgrade in a test system.
   - Disable automatic log backups through Backint.
     Disabling automatic log backup ensures that the backup history is not disrupted.
     For more information, see *Log Settings: Log Backup: Create Log Backups* in *Backup Configuration Settings*, and *SAP Note 2009486 (Disable SAP HANA log backups during upgrade of third-party backup tool that supports the Backint for SAP HANA interface)*.
2. While you are upgrading the third-party backup software:
   - Do **not** perform a data backup of the SAP HANA system.
   - Do not make any changes to the backup catalog.
3. After you have upgraded the third-party backup software, enable automatic log backups again.

> → Tip
>
> Monitor the log area to ensure that enough space is available.
>
> If your third-party backup tool is unavailable for an extended period, consider writing the log backups to the file system.

Related Information

Change the Backup Configuration Settings
SAP Note 2009486

12.2.6 SAP HANA Backup

There are different option to configure and create backups of an SAP HANA database.

You can:

- Display SAP HANA backup configuration settings
- Configure SAP HANA backups
- Back up your SAP HANA database
- Schedule SAP HANA backups
- Audit the creation and cancelation of a backup
- Delete old full backups and backup generations (housekeeping)

> ⓘ Note
>
> SAP HANA database backups include the secure stores and therefore require the root keys backup password only.

For more information, see SAP HANA Administration with SAP HANA Cockpit.

Related Information

Creating Backups (SAP HANA Administration with SAP HANA Cockpit)
Change the Backup Configuration Settings (SAP HANA Administration with SAP HANA Cockpit)
Housekeeping: Deleting and Archiving Backups (SAP HANA Administration with SAP HANA Cockpit)
SAP HANA Backup Types [page 941]
Backup Audit Actions for Security [page 1024]
Persistent Data Storage in the SAP HANA Database [page 180]
BACKUP DATA Statement (Backup and Recovery) (SAP HANA SQL Reference Guide for SAP HANA Platform)
Create Data Backups and Delta Backups (SAP HANA Studio)

12.2.6.1 Backup Configuration Parameters

In addition to displaying and changing backup configuration settings using SAP HANA cockpit, you can also configure the underlying parameters.

For more information about changing parameter values, see *Database Configuration in SAP HANA Cockpit*.

Related Information

Parameters for Data Backup Settings [page 985]
Parameters for Log Backup Settings [page 999]
Parameters for Backing Up the Backup Catalog [page 1018]
Database Configuration in SAP HANA Cockpit
Configure SAP HANA Parameters for a Third-Party Backup Tool [page 972]
Backing Up Customer-Specific Configuration Settings [page 994]
Backup Configuration Settings

12.2.6.1.1 Parameters for Data Backup Settings

You can use parameters to change default configuration settings for data backups.

Related Information

Change the Default Destination for File-Based Data Backups [page 985]
Set the Maximum File Size for File-Based Backups [page 987]
Change the I/O Buffer Size [page 989]
Set Parallel Backup Encryption [page 990]
Configure Data Backup Compression [page 991]
Prevent or Allow Data Backup Overwrite [page 993]
Include Configuration Settings in Backups [page 996]
Change the Backup Configuration Settings
Configure SAP HANA Parameters for a Third-Party Backup Tool [page 972]

12.2.6.1.1.1 Change the Default Destination for File-Based Data Backups

Each time you start a file-based data backup, you have the option to change the default backup destination or to specify a different destination for the current backup only.

Prerequisites

You require the following authorization:

For Database	Privilege
The database you are logged onto. This can be either the system database or a tenant database.	`BACKUP ADMIN`
A tenant database through the system database	`DATABASE BACKUP ADMIN` or `DATABASE ADMIN`

Context

Data backups can be written to different destinations. However, all the parts of the same data backup are written to the same destination.

For file-based data backups, you can change the default destination.

By default, file-based data backups are written to `$DIR_INSTANCE/backup/data`.

> → Tip
>
> For file-based backups, it is recommended that you create the destination directory structures **before the backup is started**.

For backups created using third-party backup tools, the default backup destination cannot be changed.

Procedure

1. Locate the parameter `basepath_databackup` in the `persistence` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.

   > → Tip
   >
   > For improved data safety, it is recommended that you specify a path to a secure backup destination. The data area, log area, and backups should never be on the same physical storage devices.

2. Open the change dialog.
3. Specify the new default destination.
4. Save.

Results

The change takes effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit

12.2.6.1.1.2 Set the Maximum File Size for File-Based Backups

For file-based data backups, you may need to limit the maximum size of a single backup file. For example, due to file system limitations.

Context

If the size of a data backup file for a service exceeds the specified limit, SAP HANA splits the file into multiple smaller files.

Procedure

1. Locate the parameter `data_backup_max_chunk_size` in the `backup` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Specify the new maximum size (in MB).

   You can specify the maximum file size of data backups up to 2048000 MB (2000 GB).

   The actual size of data backups may be smaller than the specified maximum size.

   > ### Note
   >
   > If existing backups are overwritten by backups with the same names, at least **twice the space** in the backup location is needed, because the old backup and the new backup exist for a time in parallel.
4. Save.

   > ### Note
   >
   > Changes take effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
Change the Backup Configuration Settings

12.2.6.1.1.3 Configure Minimum File Size and Channel Count for File Multistreaming

File based backups utilize multistreaming to speed up the backup process. If a file exceeds a certain size, the backup will be written using the configured number of parallel file channels.

Context

If the size of a data backup file for a service exceeds the specified limit, SAP HANA can be configured to use multiple file channels to back up the data for that service, as relying on a single file channel may result in slower performance, whereas multistreaming enhances the speed of the backup process.

Procedure

1. In the backup section of the global.ini parameter file,locate the parameters:

Parameter	Value
parallel_data_backup_file_size_threshold	Specifies the minimum size for file-based data backups to enable multistreaming. If a file backup exceeds this threshold, it will be processed using the configured number of parallel file channels specified in `parallel_data_backup_file_channels`.
parallel_data_backup_file_channels	This parameter defines the number of file channels to be used in parallel during a backup. Each channel requires a buffer, the size of which is specified by the `data_backup_buffer_size` parameter. The default buffer size is 512MB, and it should be within the range of 128MB to 4096MB, plus one additional buffer for the backup process.

2. Open the change dialog.
3. Specify the new minimum size for `parallel_data_backup_file_size_threshold` which must be in the range of [1,-] in GB. The default size is 128GB.
4. Specify the number of parallel channels to be utilized in `parallel_data_backup_file_channels`. The acceptable range is between 1 and 32 with a default value of 1.
5. Save.

Related Information

Change the Backup Configuration Settings

SAP HANA Parameter Configuration Reference
Set Parallel Backup Encryption [page 990]

12.2.6.1.1.4 Change the I/O Buffer Size

To increase or decrease the memory consumption of SAP HANA backup and recovery, you can change the I/O buffer size that is allocated by SAP HANA for data backups and log backups.

Context

To back up the data from main memory, SAP HANA allocates different numbers of buffers to data backups, depending on whether the data backups are singlestreamed or multistreamed:

Data Backup Streaming	Buffer Allocation
Singlestreamed data backups:	Allocates two I/O buffers for each persistent service
Multistreamed data backups:	Allocates n+1 I/O buffers for each persistent service. n is the number of streams defined by the parameter `parallel_data_backup_backint_channels`. For more information, see *Configure Multistreaming with Third-Party Backup Tools*.

> **Note**
>
> Changing the I/O buffer memory consumption affects each persistent service for the database that is being backed up.
>
> If you change the size of the backup I/O buffer, check to see if the change has any effect on backup runtime. If you do not notice an improvement in backup runtime, consider restoring the SAP HANA default size.

Procedure

1. In the `backup` section of the `global.ini` parameter file, locate the parameter `data_backup_buffer_size`.
2. Open the change dialog.
3. Specify the new I/O buffer size for data backups.

Possible values:	128 - 4096 MB

Default:	512 MB for each buffer

4. Save.

Results

The change takes effect immediately.

Related Information

SAP HANA Backup Encryption [page 950]
Configure Multistreaming with Third-Party Backup Tools [page 976]
Prerequisites: Recovery Using Multistreamed Backups [page 1049]
Configuring System Properties in SAP HANA Studio
Database Configuration in SAP HANA Cockpit

12.2.6.1.1.5 Set Parallel Backup Encryption

You can enable or disable parallel backup encryption.

Context

When backup encryption is enabled, additional operations have to be performed for each backup buffer (encryption, checksum). These can be executed either serially, or in parallel for maximum throughput (default).

If parallel backup encryption is enabled, backup and recovery with encryption is executed with maximum throughput. This requires higher CPU and memory consumption.

If parallel backup encryption is disabled, backup and recovery with encryption enabled is executed with the same resource consumption as without encryption. Typically, backup and recovery operations will take longer to complete.

Procedure

1. In the `backup` section of the `global.ini` parameter file, locate the parameter `enable_parallel_backup_encryption`.
2. Open the change dialog.

3. Specify the new setting.

Enable parallel backup encryption:	`enable_parallel_backup_encryption = true` (Default)
Disable parallel backup encryption:	`enable_parallel_backup_encryption = false`

4. Save.

Related Information

SAP HANA Backup Encryption [page 950]
Database Configuration in SAP HANA Cockpit

12.2.6.1.1.6 Configure Data Backup Compression

You can use parameters to configure the data backup compression level and the compression algorithm.

Context

Currently, SAP HANA supports the LZ4 compression algorithm only.

For more information about how to change parameter settings, see *Database Configuration in SAP HANA Cockpit*.

By default, data backup compression is disabled.

You can enable compression of data backups using SQL. (Compression of data backups is not enabled using a parameter.) For more information, see *BACKUP DATA Statement (Backup and Recovery)* in the *SAP HANA SQL and System Views Reference*.

> **Note**
>
> If you have enabled data backup encryption using the COMPRESSED clause of the BACKUP DATA statement, the data backups are automatically compressed before they are encrypted.
>
> For more information about backup encryption, see *SAP HANA Backup Encryption*.

Procedure

1. Locate the following parameters in the `backup` section of the `global.ini` parameter file.

Parameter	Value
`data_backup_compression_algorithm`	Specifies the compression algorithm to be used. Currently, only LZ4 is supported: `data_backup_compression_algorithm = lz4`
`data_backup_compression_level`	Specifies the compression level. Possible values: 0 - 65537 With LZ4, a larger compression level value means a **faster compression time**, but a correspondingly larger backup size. The default setting is: `data_backup_compression_level = 0` 0 means maximum possible compression, and the longest compression time.

2. Make your changes and *Save*.

   Changes take effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
SAP HANA Backup Encryption [page 950]
BACKUP DATA Statement (Backup and Recovery)

12.2.6.1.1.7 Prevent or Allow Data Backup Overwrite

By default, SAP HANA overwrites existing data backups in the file system with new backups with the same prefix. You can change this configuration to prevent existing backups from being overwritten.

Prerequisites

You require the following authorization:

For Database	Privilege
The database you are logged onto. This can be either the system database or a tenant database.	`BACKUP ADMIN`
A tenant database through the system database	`DATABASE BACKUP ADMIN` or `DATABASE ADMIN`

Context

You can configure SAP HANA to prevent existing complete data backups from being overwritten by new backups with the same prefix.

Procedure

1. In the `backup` section of the `global.ini` parameter file, locate the parameter `enable_data_backup_overwrite`.
2. Open the change dialog.
3. Specify the new setting.

Allow existing complete data backups to be overwritten:	`enable_data_backup_overwrite = true` (Default)
Prevent existing complete data backups from being overwritten:	`enable_data_backup_overwrite = false`

4. Save.

Related Information

Database Configuration in SAP HANA Cockpit
Backup Configuration Settings
Authorizations Needed for Backup and Recovery [page 938]

12.2.6.1.1.8 Backing Up Customer-Specific Configuration Settings

By default, customer-specific configuration settings (`.ini` files that have been changed from the default) are not backed up as part of a data backup.

The customer-specific configuration settings are not essential to perform a database recovery. If the `.ini` files are not available, SAP HANA automatically recreates the default configuration.

To backup up customer-specific configuration files, you can:

Option	Steps
During a database backup	Configure SAP HANA to include customer-specific configuration files in data backups. For more information, see *Include Configuration Settings in Backups* and *BACKUP DATA Statement (Backup and Recovery)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.
Manually	You can use an SQL script to generate an overview of configuration files that have been changed from the SAP HANA default settings. For more information, see *Back Up Customer-Specific Configuration Settings*.

Locations of the SAP HANA Configuration Files

By default, configuration files for SAP HANA are written to specific directories.

Locations of the SAP HANA Configuration Files

Configuration Settings	Location
Global configuration settings	`/usr/sap/$SID/global/hdb/custom/config/`

Configuration Settings	Location
Configuration settings for a tenant database	`/usr/sap/$SID/global/hdb/custom/config/DB_<database_name>` For more information, see *Database-Specific Configuration Parameters* in the *SAP HANA Administration Guide*.
Host-specific configuration settings	`/usr/sap/<SID>/HDB<instance no.>/<host>/`

> **Note**
>
> Configuration files (`.ini` files) are only created if customer-specific changes are made to them after installation. If no customer-specific changes have been made, the directories may be empty.

Default SAP HANA Configuration Files

During installation of SAP HANA database, the following configuration files are created:

Content of the Main SAP HANA Configuration Files

Configuration File	Description
`sapprofile.ini`	Contains system identification information, such as the system name (SID) or the instance number. After installation, `sapprofile.ini` is not changed again. **Caution** `sapprofile.ini` contains information that is specific to each host. For this reason, in a recovery situation, the `sapprofile.ini` file must not be copied manually to a different host, as it will not be compatible with a new landscape. The `sapprofile.ini` file can be found in the following directory: `/usr/sap/<SID>/HDB<instance no.>/<host>/` **Note** `sapprofile.ini` is not displayed in the *Configuration of System Properties* overview in SAP HANA cockpit.

Configuration File	Description
`daemon.ini`	Contains information about which database services to start. The `daemon.ini` file can be found in the following directory: `/usr/sap/<SID>/HDB<instance no.>/<host>/`
`nameserver.ini`	The `nameserver.ini` file contains global configuration settings for each SAP HANA installation. The landscape section contains the system-specific landscape ID and assignments of hosts to roles MASTER, WORKER, and STANDBY. It also contains configuration settings for system replication and for the SAP HANA Storage Connector API. If the system landscape is changed, for example, hosts are added or removed, the landscape section of the `nameserver.ini` is also changed.

Related Information

Back Up Customer-Specific Configuration Settings [page 998]
Include Configuration Settings in Backups [page 996]
Database Configuration in SAP HANA Cockpit
Change the Backup Configuration Settings
Configuring SAP HANA System Properties (INI Files) [page 126]
Database-Specific Configuration Parameters
Check Individual Backups Inside an SAP HANA Installation (hdbbackupdiag, hdbbackupcheck) [page 1042]
BACKUP DATA Statement (Backup and Recovery)

12.2.6.1.1.8.1 Include Configuration Settings in Backups

You can configure SAP HANA to include customer-specific configuration settings (.ini files changed from their default values) in data backups and delta backups.

Context

By default, a data backup does not include customer-specific configuration settings.

Also by default, a recovery does not restore customer-specific configuration settings.

Procedure

To configure SAP HANA to include or exclude customer-specific configuration settings in data backups and delta backups:

1. In the `backup` section of the `global.ini` parameter file, locate the parameter `include_configuration_backup`.
2. Open the change dialog.
3. Specify the new setting.

Include customer-specific configuration files in data backups and delta backups:	`include_configuration_backup = true` Default: `false`

4. Save.

Results

The change takes effect immediately.

You can verify that user configuration files are backed up as follows:

To...	Do the Following...
Confirm that a backup contains backups of user configuration files...	The M_BACKUP_CATALOG_FILES system view includes the following: CONFIGURATION_INCLUDED For more information, see *M_BACKUP_CATALOG_FILES System View* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.
View the content of the user configuration in a data backup...	Use the `--dump -c <.ini file>` option of the `hdbbackupcheck` tool. For more information, see *Check Individual Backups Inside an SAP HANA Installation (hdbbackupdiag, hdbbackupcheck)*.

Related Information

BACKUP DATA Statement (Backup and Recovery)

RECOVER DATA Statement (Backup and Recovery)
RECOVER DATABASE Statement (Backup and Recovery)
M_BACKUP_CATALOG_FILES System View
Change the Backup Configuration Settings
Check Individual Backups Inside an SAP HANA Installation (hdbbackupdiag, hdbbackupcheck) [page 1042]

12.2.6.1.1.8.2 Back Up Customer-Specific Configuration Settings

You can generate an overview of customer-specific configuration settings (`.ini` files that have been changed from the default), and use this information to restore your configuration settings after a recovery or a database copy.

Context

A backup of changes to the configuration settings can be helpful to more easily identify and restore customer-specific changes to the default settings. You can back up, and also restore, customer-specific changes using SQL.

Procedure

1. To generate a list of customer-specific configuration settings, execute the following SQL script on the system database:

   Code Syntax

   ```
   select 'ALTER SYSTEM ALTER CONFIGURATION  ('''|| file_name ||''', '''||
          case layer_name
              when 'SYSTEM' then layer_name
              when 'DATABASE' then layer_name ||''', '''|| database_name
              when 'HOST' then layer_name ||''', '''|| host
          end ||
          ''') SET ('''|| section ||''', '''|| key ||''') = '''||value ||'''
   WITH RECONFIGURE;'
          as "Configuration File Backup"
     from sys_databases.m_inifile_contents
   where layer_name != 'DEFAULT';
   ```

2. The SQL script generates a list of configuration settings that have been changed from the SAP HANA default settings.

   Each parameter change is output as an `ALTER SYSTEM ALTER CONFIGURATION` statement.

For example:

Output Code

```
ALTER SYSTEM ALTER CONFIGURATION  ('nameserver.ini', 'SYSTEM') SET
('execution', 'max_concurrency') = '20' WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION  ('nameserver.ini', 'SYSTEM') SET
('execution', 'single_thread_group') = '1' WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION  ('nameserver.ini', 'SYSTEM') SET
('httpserver', 'embedded') = 'true' WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION  ('nameserver.ini', 'SYSTEM') SET
('httpserver', 'listenport') = '3$(SAPSYSTEM)14' WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION  ('nameserver.ini', 'SYSTEM') SET
('httpserver', 'maxthreads') = '10' WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION  ('nameserver.ini', 'SYSTEM') SET
('httpserver', 'workerpoolsize') = '5' WITH RECONFIGURE;
```

3. Check which of the parameters you want to restore to your new SAP HANA database.

   Caution

   Before you restore the parameter changes, ensure that the parameter changes are compatible with a different system.

   For example, if a landscape ID is used, ensure that it is the landscape ID of the new system.

4. Copy the SQL statements for the parameter changes that you want to restore to the new system.
5. Execute the desired SQL statements on the new system.

   The configuration settings are now updated in the new system.

Related Information

ALTER SYSTEM ALTER CONFIGURATION Statement (System Management)
Include Configuration Settings in Backups [page 996]

12.2.6.1.2 Parameters for Log Backup Settings

You can configure parameters to control the behavior of log backups.

Related Information

Find the Optimal Log Backup Configuration [page 1000]
Enable and Disable Automatic Log Backup [page 1007]
Log Modes [page 1009]
Change Log Modes [page 1011]
Set the Interval Mode for Log Backups [page 1012]

Change the Log Backup Interval [page 1013]
Change the Log Backup Destination Type [page 1015]
Change the Log Backup Destination [page 1016]
Consolidated Log Backups [page 1002]
Configure Log Backup Compression [page 1017]

12.2.6.1.2.1 Find the Optimal Log Backup Configuration

To ensure that there is an optimal balance between log backup performance and the safety of your production data, you can adjust the log backup configuration.

Log Backup Configuration

The following configuration options influence log backup behavior:

Log Backup Option	See...
Log backup interval	*Change the Log Backup Interval*
Log backup interval mode	*Set the Interval Mode for Log Backups* Parameter `log_backup_interval_mode`
Consolidated log backups	*Consolidated Log Backups* Use the parameter `max_log_backup_size` in combination with the log backup interval mode (parameter `log_backup_interval_mode`).
Log segment size	Parameter `log_segment_size_mb` in the *SAP HANA Configuration Parameter Reference*.

Considerations for an Optimal Log Backup Interval

Issue	Consider...
Does a large number of small log backup files cause performance issues?	Reducing the frequency of log backups could enhance log backup performance. To reduce the number of log backup files, consider: • Setting the log backup interval mode to service. service offers maximum benefit when using consolidated log backups. With service, each log backup is created only after a service-specific timeout has been reached. • Increasing the log segment size and leaving the log backup interval mode immediate. A log backup is created immediately after a log segment becomes full, or when the service-specific timeout for a log segment has been reached.
If log backups are created infrequently, and the log area fails, many log segments may be irretrievably lost. How much data loss is acceptable to your recovery point objective?	A lower frequency of creating log backups can negatively impact your recovery point objective if the log area fails. To ensure minimum data loss, consider keeping `log_backup_interval_mode` set to the default value immediate.
If the log area is corrupted, can your recovery point objective still be met?	Consolidated log backups, with a small number of relatively large log backups, can enhance log backup performance. However, as a consequence, if the log area fails, a lot of data may be irretrievably lost. For example, collecting 10 GB of log segments in one log backup will mean that fewer log backup are created and less frequently. If the log area is corrupted, it may mean that 10 GB of log data is irretrievably lost. If it is acceptable to have a small number of large backup files, consider setting `log_backup_interval_mode` to service. With log backup interval mode service, each log backup is created only after a service-specific timeout has been reached.
What file size can be handled with the highest I/O throughput in your landscape?	To keep the overhead low, you can reduce the number of log segments. Consider increasing the log segment size (parameter `log_segment_size_mb`).

Related Information

Change the Log Backup Interval [page 1013]
Set the Interval Mode for Log Backups [page 1012]
Consolidated Log Backups [page 1002]
log_segment_size_mb in the SAP HANA Configuration Parameter Reference

12.2.6.1.2.1.1 Consolidated Log Backups

To improve the performance of log backups, SAP HANA can write multiple log segments of a service to a single consolidated log backup. You configure the maximum size of a consolidated log backup to be processed by a single backup operation.

Context

Why Consolidate Log Backups?

Log segments in the log area are only released after a log backup was completed successfully. After log segments have been released, the space they occupy can then be overwritten with new log segments.

If a single log backup operation takes a long time, during that time, several other log segments may be queued for backup. During periods of high load, log segments may be closed and queued for backup faster than a single backup operation can be completed.

If many log segments are waiting to be backed up, there can be a delay in releasing them. As a result of this delay, the log area may grow.

If the log segments cannot be backed up and released faster than the log area is growing, the log area could become full.

To remedy this issue, SAP HANA can write all the log segments of a service that are ready to be backed up at a particular time to a single consolidated log backup. A consolidated log backup contains multiple log segments.

> ⓘ Note
>
> The option to consolidate log backups is primarily intended for use with third-party backup tools that create a large number of small backups in a short time. However, consolidated log backups are supported for both third-party backup tools and file system backups.

Log Backup Interval Mode

The behavior of the log backup process is also influenced by the log backup interval mode, which can be immediate (default) or service.

immediate means that a log backup is created immediately when a log segment becomes full, or when the service-specific timeout for a log segment has been reached. With log backup interval mode immediate, there are typically only a few log segments within one log backup.

In the SAP HANA log area, the log segments that are part of a log backup are only released after the last of the queued log segments has been backed up and if they are no longer needed to restart the database.

For more information, see *Set the Interval Mode for Log Backups*.

Procedure

Configure the maximum size of a consolidated log backup to be processed by a single backup operation.

1. Locate the parameter `max_log_backup_size` in the `backup` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Select the database or the host.

   You can configure the size of a consolidated log backup for one or more tenant databases and for one or more hosts.
4. Set the size in GB.

   The default value is **16** GB.

   This means that one (consolidated) log backup is filled with log segments up to a maximum size of 16 GB.

   The maximum size allowed is 64 GB.

   To reset to the default value, choose *Restore Default*.
5. Save.

Results

The backup catalog is written and backed up **once** for each log backup.

To check that consolidated log backups are enabled, you can use the `hdbbackupdiag` tool or the M_BACKUP_CATALOG_FILES system view.

For more information, see *Verify That Log Backups are Being Consolidated (hdbbackupdiag)* or *Verify That Log Backups are Being Consolidated (M_BACKUP_CATALOG_FILES)*.

> ⓘ Note
>
> In the backup catalog, separate entries are maintained for the log segments in a log backup. However, if you subsequently want to remove log backups from a consolidated log backup, you can only remove all the consolidated log backups together, and then only if none of them is still needed for recovery.

Related Information

Verify That Log Backups are Being Consolidated (hdbbackupdiag) [page 1004]

Verify That Log Backups are Being Consolidated (M_BACKUP_CATALOG_FILES) [page 1005]
Change the Log Backup Interval [page 1013]
Set the Interval Mode for Log Backups [page 1012]
Find the Optimal Log Backup Configuration [page 1000]
Database Configuration in SAP HANA Cockpit

12.2.6.1.2.1.1.1 Verify That Log Backups are Being Consolidated (hdbbackupdiag)

You can check whether log backup consolidation is enabled or disabled.

Context

If a log backup contains multiple log segments, log backup consolidation is enabled.

To check log whether a log backup contain more than one log segment, you can use the `hdbbackupdiag` tool.

Procedure

Execute `hdbbackupdiag -d <log_backup_directory> -b <log_backup_name>`.

`<log_backup_directory>` is the directory where the log backups are located.

Results

Information is displayed about the log segments in the log backup.

If the log backup contains more than one log segment, log backup consolidation is enabled.

Example

In this example, `hdbbackupdiag` is used to check the content of a log backup.

```
hdbbackupdiag -d /usr/sap/HD2/HDB02/backup/log/DB_HD2 -b
log_backup_2_0_7246592_7267136.1613127042627
```

Returns the following results:

Output Code

```
Backup file: /usr/sap/HD2/HDB02/backup/log/DB_HD2/
log_backup_2_0_7246592_7267136.1613127042627
Backup time: 2021-02-12T10:50:42+00:00
```

```
LogSegmentBackupInfo:
StartPosition: 7246592
EndPosition: 7266816
LogSegmentBackupInfo:
StartPosition: 7266816
EndPosition: 7267136
```

Here, the log backup contains two log segments. This indicates that log backup consolidation is enabled and working correctly.

Related Information

Check the Backups Required for a Recovery (hdbbackupdiag) [page 1036]
Verify That Log Backups are Being Consolidated (M_BACKUP_CATALOG_FILES) [page 1005]

12.2.6.1.2.1.1.2 Verify That Log Backups are Being Consolidated (M_BACKUP_CATALOG_FILES)

You can check whether log backup consolidation is enabled or disabled.

Context

If a log backup contains multiple log segments, log backup consolidation is enabled.

To check whether log backup consolidation is enabled, you can use the system view M_BACKUP_CATALOG_FILES.

If the LOG_SEGMENT_COUNT is greater than 1, log backup consolidation is enabled.

For more information, see *M_BACKUP_CATALOG_FILES System View*.

Procedure

To check the log segment count in M_BACKUP_CATALOG_FILES, use the following SQL statement:

SELECT * from M_BACKUP_CATALOG_FILES WHERE LOG_SEGMENT_COUNT > 1

Results

If more than one row is returned, log backup consolidation is enabled.

Example

The SQL statement could return the following results:

Output Code

```
ENTRY_ID,BACKUP_ID,SOURCE_ID,SOURCE_TYPE_NAME,HOST,SERVICE_TYPE_NAME,REDO_L
OG_POSITION,FIRST_REDO_LOG_POSITION,LAST_REDO_LOG_POSITION,BACKUP_SIZE,DEST
INATION_PATH,DESTINATION_TYPE_NAME,EXTERNAL_BACKUP_ID,LOG_SEGMENT_COUNT,BAC
KUP_CATALOG_BACKUP_ID,UTC_LAST_COMMIT_TIME,BACKINT_FALLBACK_USED,COMPRESSED
_SIZE,COMPRESSION_ALGORITHM
1613126753637,1613126753637,2,"volume","cc-hostd2-v","indexserver",?
,7228096,7246592,1183744,"/usr/sap/HD2/HDB02/backup/log/DB_HD2/
log_backup_2_0_7228096_7246592.1613126753637","file","",2,1613126754280,"20
21-02-12 10:45:53.423639000","FALSE",? ,?
1613127042627,1613127042627,2,"volume","cc-hostd2-v","indexserver",?
,7246592,7267136,1314816,"/usr/sap/HD2/HDB02/backup/log/DB_HD2/
log_backup_2_0_7246592_7267136.1613127042627","file","",2,1613127042929,"20
21-02-12 10:50:42.373881000","FALSE",? ,?
2 rows selected (overall time 3090 usec; server time 2096 usec)
```

Here, the log backup contains two log segments. This indicates that log backup consolidation is enabled and working correctly.

Related Information

Check the Backups Required for a Recovery (hdbbackupdiag) [page 1036]
M_BACKUP_CATALOG_FILES System View
Verify That Log Backups are Being Consolidated (hdbbackupdiag) [page 1004]

12.2.6.1.2.1.2 Troubleshooting: Slow Log Backups

If log backups are delayed or interrupted, as a consequence, space will not be freed up in the log area fast enough to accommodate new log segments.

Context

If logs are being backed up too slowly, over time, the log area can grow until the file system is full. If a disk full error occurs, no more log segments can be created, and the database freezes.

→ Tip

It is recommended that you proactively check the performance of log backups.

Procedure

To check the performance of log backups:

1. Locate the SQL statement `LogBackupDelays` in the *Statement Library*.

   For more information, see *Use the Statement Library to Administer Your Database*.
2. Check the values returned for log backups.

Performance Indicator	Description
`NOT_BACKED_UP`	A value greater than 2 indicates that logs are being backed up slowly. For more information, see *SAP Note 1835075 (Analyze backup and recovery performance issues).*
`TIME_SINCE_BACKUP_H`	If the time continues to increase, this indicates a delay in logs being backed up. If automatic log backups are disabled, ensure that they are enabled. For more information, see *Enable and Disable Automatic Log Backup*. If the log backup thread is blocked, contact SAP HANA support.

Related Information

Find the Optimal Log Backup Configuration [page 1000]
Parameters for Log Backup Settings [page 999]
Enable and Disable Automatic Log Backup [page 1007]
SAP Note 1835075

12.2.6.1.2.2 Enable and Disable Automatic Log Backup

By default, SAP HANA creates redo log backups automatically at regular intervals. You can disable and enable automatic log backups.

Prerequisites

To enable automatic log backups, the log mode must be set to `normal`.

For more information, see *Log Modes*.

> ⚠ Caution
>
> During normal system operation (log mode `normal`), it is strongly recommended that you enable automatic log backups. When log segments are backed up, the space they occupied in the log area can be freed. SAP HANA can overwrite the newly freed space in the log area with new log entries. In this way, automatic log backups can prevent the log area from filling. If automatic log backups are disabled, the log area grows until the file system is full. If the file system is full, and no more log segments can be created, the database freezes.

Procedure

1. Locate the parameter `enable_auto_log_backup` in the `persistence` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Specify whether to enable or disable automatic log backups.

   The default setting is `yes` (automatic log backup is active).

   You can specify either `yes` or `no` to enable or disable automatic log backups.

   To reset to enable automatic log backups, choose *Restore Default*.
4. Save.

Results

The change takes effect immediately.

If any log backups are running, they will first be completed before automatic log backups are disabled.

Related Information

Log Modes [page 1009]
Database Configuration in SAP HANA Cockpit
Change the Backup Configuration Settings

12.2.6.1.2.3 Log Modes

SAP HANA can run either in log mode `normal` or `overwrite`.

After installation, SAP HANA temporarily runs in log mode `overwrite`.

In log mode `overwrite`, the log segments are not backed up. When a log segment is full, it is closed. After a successful savepoint, that log segment can be reused. In this way, log mode `overwrite` ensures that the log area does not grow excessively.

After you create the first full data backup, SAP HANA automatically switches to the default log mode `normal`, and log backups are created.

> → Tip
>
> If you change the log mode from `overwrite` – where log backups are not created – to log mode `normal`, **you must create a full data backup** to ensure that log backups are created again, and that the database can be recovered to the most recent point in time.

SAP HANA Log Modes

Log Mode	Description
`normal` (Default)	In log mode `normal`, log segments are backed up automatically if automatic log backups are enabled. For more information, see *Enable and Disable Automatic Log Backup* and *Changing the Backup Configuration Settings*. → Tip Log mode `normal` is recommended to provide support for point-in-time recovery. After a log segment has been backed up, closed, and a savepoint has been written, the space it occupied can be freed. SAP HANA can then overwrite the newly freed space in the log area with new log entries. In this way, automatic log backups can prevent the log area from filling. ⚠ Caution If the log area becomes full and no more log segments can be created in the file system, the database freezes. No more log entries can be written until a log backup has been completed, and the log segments are no longer needed to restart the database.

Log Mode	Description
`overwrite`	No log backups are created. When savepoints are written, log segments are immediately freed to be overwritten by new log entries. When log mode `overwrite` is active, the *Log Backup Settings* in the Backup Console cannot be changed. Log mode overwrite can be useful for installations that do not need to be backed up or recovered. For example, for test installations. **⚠ Caution** Log mode `overwrite` is not recommended for production systems. With log mode `overwrite`, a point-in-time recovery is not possible. Delta backups created in log mode `overwrite` cannot be used for a point-in-time recovery. **ⓘ Note** Even if no log backups are written, with each data backup, the **backup catalog** is still backed up.

⚠ Caution

Do not delete log segments at operating system level, as this makes the log area unusable. As a consequence, the database may stop working immediately, and it will not be possible to restart the database.

For more information about switching log modes, see *Backup Configuration Settings* or *Change Log Modes*.

Related Information

Change the Backup Configuration Settings
Change Log Modes [page 1011]
Enable and Disable Automatic Log Backup [page 1007]
Change a System Property in SAP HANA Studio
Savepoints and Redo Logs [page 949]
SAP Note 2375865
Change the Log Backup Interval [page 1013]
Set the Interval Mode for Log Backups [page 1012]

12.2.6.1.2.3.1 Change Log Modes

You can switch between the SAP HANA log modes `normal` and `overwrite`.

Context

> → Tip
>
> If you change the log mode from `overwrite` – where log backups are not created – to log mode `normal`, **you must create a full data backup** to ensure that log backups are created again, and that the database can be recovered to the most recent point in time.

System Replication

If you are using system replication, ensure that the log mode is `normal`. If the log mode is `overwrite`, log segments will not be backed up.

For more information, see *General Prerequisites for Configuring SAP HANA System Replication*.

Procedure

1. Locate the parameter `log_mode` in the `persistence` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Specify the new log mode.

   The log mode can be either `normal` or `overwrite`.

   For more information, see *Log Modes*.

   To reset SAP HANA to the default log mode `normal`, choose *Restore Default*.
4. Save.

   The change takes effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
Log Modes [page 1009]
General Prerequisites for Configuring SAP HANA System Replication [page 765]

12.2.6.1.2.4 Set the Interval Mode for Log Backups

You can specify an interval mode for log backups.

Context

> → Tip
>
> You can also specify the log backup interval mode using the Backup Configuration App in SAP HANA cockpit.

Procedure

1. Locate the parameter `log_backup_interval_mode` in the `backup` section of the `global.ini` parameter file.

   For more information about how to change parameter settings, see *Database Configuration in SAP HANA Cockpit*.

   `log_backup_interval_mode` controls the use of the log backup timeout.

   For more information, see *Change the Log Backup Interval*.
2. Specify the desired log backup interval mode.

   You can set the following interval modes:

Log Backup Interval Mode	Description
immediate (Default)	A log backup is created immediately after a log segment becomes full, or when the service-specific timeout for a log segment has been reached. If you have configured consolidated log backups (parameter `max_log_backup_size`), the log backup process does not wait for the configured size of log backups to be queued. If fewer log backups are queued, all the queued log backups are processed. For more information, see *Consolidated Log Backups*.
service	A log backup is created only after a service-specific timeout has been reached. This backup includes all the log segments in states **writing**, **closed**, and **truncated**.

3. *Save*.

Changes take effect immediately.

> **Note**
>
> The log interval mode and timeout are included in the system view `M_BACKUP_CONFIGURATION`.
>
> For more information, see *M_BACKUP_CONFIGURATION*.

Related Information

Database Configuration in SAP HANA Cockpit
Change the Backup Configuration Settings
Change the Log Backup Interval [page 1013]
Timeout for Log Backups (Backint) [page 978]
Consolidated Log Backups [page 1002]
M_BACKUP_CONFIGURATION System View

12.2.6.1.2.5 Change the Log Backup Interval

You can change the interval at which log backups are created.

Prerequisites

The log backup interval takes effect only if **automatic log backups** are enabled.

To enable automatic log backups, the log mode must be `normal`.

For more information, see *Log Modes*.

Context

Specifying an appropriate interval for log backups enables you to recover an SAP HANA database with a good Recovery Point Objective (RPO). In the event of database failure, the RPO is the maximum time span from which data will be lost if the log area cannot be used for recovery, and if only data backups, delta backups, and log backups are available.

> **Note**
>
> If the log segments become full, they are backed up immediately, even if the log backup interval has not been reached.

> → Tip
>
> You can specify the log backup interval using the Backup Configuration App in SAP HANA cockpit.

Procedure

Change the log backup interval.

1. Locate the parameter `log_backup_timeout_s` in the `persistence` section of the `global.ini` parameter file.

   For more information about how to change parameter settings, see *Database Configuration in SAP HANA Cockpit*.
2. Specify the desired log backup interval.

   By default, the log backup interval is 15 minutes (900 s).

   A log backup interval of 15 minutes (or less) is recommended for production systems.

   For test systems, you can set a longer log backup interval, depending on what data loss is acceptable to you if a fault occurs.

   > ⓘ Note
   >
   > If you specify a timeout of 0, log backups are created only when a log segment is full and when services are restarted.
3. Save.

Results

The change takes effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
Log Modes [page 1009]
Set the Interval Mode for Log Backups [page 1012]

12.2.6.1.2.6 Change the Log Backup Destination Type

You can change parameters to the destination type for file-based log backups.

Procedure

1. Locate the parameter `log_backup_using_backint` in the `backup` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Specify the destination type.

   Possible values:

   - true (back up using Backint)
   - false (back up to the file system)

   By default, log backups are written to the file system.

   > ### Note
   >
   > The destination type *Backint* is only available if the Backint agent is installed.
4. Save.

Results

The change takes effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
Destination for Backups of the Backup Catalog [page 1020]

12.2.6.1.2.7 Change the Log Backup Destination

You can change parameters to the default destination for file-based log backups.

Context

> ⓘ Note
>
> The log backup destination can only be changed for file-based backups.
>
> It is not possible to change the backup destinations for third-party backup tools.
>
> Backups created using third-party backup tools always use the destination: `/usr/sap/<SID>/SYS/global/hdb/backint`
>
> A third-party backup tool reads data to be backed up from named pipes, and writes the data in accordance with the tool configuration. For a third-party backup tool, the only objects created in the file system are named pipes. Named pipes occupy no space in the file system.

Procedure

1. Locate the parameter `basepath_logbackup` in the `persistence` section of the `global.ini` parameter file.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Specify the destination for log backups.

   By default, file-based log backups are written to: `$DIR_INSTANCE/backup/log`

   > → Tip
   >
   > For improved data safety, it is recommended that you specify a path to a secure backup destination.
   >
   > The data area, log area, data backups, and log backups should never be on the same physical storage devices.
4. Save.

Results

The change takes effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit

12.2.6.1.2.8 Configure Log Backup Compression

You can use parameters to configure log backup compression.

Context

Currently, SAP HANA supports the LZ4 compression algorithm only.

For more information about how to change parameter settings, see *Database Configuration in SAP HANA Cockpit*.

> **Note**
>
> If you have enabled log backup encryption together with log backup compression, the log backups are automatically compressed before they are encrypted.
>
> For more information about backup encryption, see *SAP HANA Backup Encryption*.

Procedure

1. Locate the following parameters in the `backup` section of the `global.ini` parameter file.

Parameter	Value
`enable_log_backup_compression`	By default, log backup compression is disabled. `enable_log_backup_compression = true` enables log backup compression.
`log_backup_compression_algorithm`	Specifies the compression algorithm to be used. Currently, only LZ4 is supported: `log_backup_compression_algorithm = lz4`

Parameter	Value
`log_backup_compression_level`	Specifies the compression level. Possible values: 0 - 65537 With LZ4, a larger compression level value means a **faster compression time**, but a correspondingly larger backup size. The default setting is: `log_backup_compression_level = 0` 0 means maximum possible compression, and the longest compression time.

2. Make your changes and *Save*.

   Changes take effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit
SAP HANA Backup Encryption [page 950]

12.2.6.1.3 Parameters for Backing Up the Backup Catalog

Each time a backup of any type is created, the backup catalog is backed up and versioned. In this way, the latest version of the backup catalog always contains the entire backup history.

Even in situations such as when `log_mode = overwrite` is set, where log backups are not created, the backup catalog is still backed up.

If the backup catalog is backed up using a third-party tool, the tool also handles the versioning of the backup catalog.

You can manually change the following settings for the backup catalog:

- The destination type
  The destination type can be the file system or a third-party backup tool.
  For more information, see *Destination Type for Backups of the Backup Catalog*.
- Where the backup catalog is backed up
  For more information, see *Destination for Backups of the Backup Catalog*.
- Whether backups of the backup catalog are accumulated
  For more information, see *Accumulated Backups of the Backup Catalog*.

Related Information

Destination Type for Backups of the Backup Catalog [page 1019]
Destination for Backups of the Backup Catalog [page 1020]
Accumulated Backups of the Backup Catalog [page 1022]
Change the Backup Configuration Settings
Configure SAP HANA Parameters for a Third-Party Backup Tool [page 972]
Log Modes [page 1009]

12.2.6.1.3.1 Destination Type for Backups of the Backup Catalog

For backups of the backup catalog, you can configure the backup destination type.

Parameter to Configure the Destination Type for Backups of the Backup Catalog

Task	Parameter
Destination Type:	Write backups of the backup catalog to the file system or using a third-party backup tool. `catalog_backup_using_backint` in `global.ini/[backup]/` Values: • true (back up using Backint) • false (back up to the file system) Default value: false (Backups of the backup catalog are written to the file system.)

For more information about changing parameter settings, see *Database Configuration in SAP HANA Cockpit*.

Related Information

Database Configuration in SAP HANA Cockpit

12.2.6.1.3.2 Destination for Backups of the Backup Catalog

You can change the default destination for backups of the backup catalog.

By default, the backup catalog is backed up to the same destination as the **log backups**. If required, you can configure a separate destination for backups of the backup catalog.

> ⚠ Caution
>
> If you change the default destination for the log backups, the backup catalog is **not automatically backed up to the same location**.
>
> For this reason, if you change the default destination for the log backups, you must also check that the backup catalog is being backed up to the desired destination.

For more information about changing parameter settings, see *Database Configuration in SAP HANA Cockpit*.

Parameters to Configure the Destination for Backups of the Backup Catalog

Task	Parameter
File-based Backups:	Specify the directory to which to write backups of the backup catalog. `basepath_catalogbackup` in `global.ini/[persistence]/` Default value: empty (Backups of the backup catalog are written to the default directory for log backups: `$DIR_INSTANCE/backup/log`)
Backint:	Specify a Backint parameter file for the backup catalog. `catalog_backup_parameter_file` in `global.ini/[backup]/` Default value: empty (No Backint parameter file is used.)

Considerations for the Backup Catalog and Recovery

When you recover SAP HANA, you are prompted to select a directory or destination type for the backup catalog, regardless of the location of the backup catalog that is currently configured. SAP HANA then only searches the specified directory (not the subdirectories) or Backint, and selects the most recent backup catalog available there.

Recovery and the Backup Catalog Using SQL

To recover SAP HANA using native SQL, the location of the backup catalog in the file system can be specified in the syntax for `RECOVER DATABASE` and `RECOVER DATA` using the following clause:

- File-based backups:
  `USING CATALOG PATH ('<path>')`
- Third-party backup tools:
  `USING CATALOG BACKINT`

If no location is specified, the location specified in the .ini files is used.

For more information, see *RECOVER DATABASE Statement (Backup and Recovery)* and *RECOVER DATA Statement (Backup and Recovery)* in the *SAP HANA SQL and System Views Reference*.

Considerations for System Copy

When you recover SAP HANA to create a system copy, it is not necessary to move old log backups and old backups of the backup catalog. The location of the backup catalog is specified explicitly.

Considerations for an SAP HANA Upgrade

During an upgrade from SAP HANA 1.0 to SAP HANA 2.0, customer-specific configuration settings are automatically retained.

> → Tip
>
> Before an upgrade, if the destination for log backups was changed, you should check that the backup catalog is being backed up to the desired destination. Check the parameters `catalog_backup_using_backint`, `catalog_backup_parameter_file`, and `basepath_catalogbackup`.

Consideration for a New SAP HANA Installation

When you install a new SAP HANA system, ensure that the log backups and the backup catalog are BOTH backed up to the desired destination.

Related Information

Database Configuration in SAP HANA Cockpit
RECOVER DATABASE Statement (Backup and Recovery)
RECOVER DATA Statement (Backup and Recovery)

Consolidated Log Backups [page 1002]
Copying a Database Using Backup and Recovery [page 1065]
Change the Log Backup Destination Type [page 1015]
Timeout for Log Backups (Backint) [page 978]

12.2.6.1.3.3 Accumulated Backups of the Backup Catalog

Each time a backup is created, the operation is recorded in the backup catalog, which is itself then backed up. If many data and log backups are created within a short time, the backup catalog would need to be backed up just as frequently.

If many backups of the backup catalog are queued to run, the most recently completed backup of the backup catalog will not reflect the most recent database backups. If many backups are waiting to be processed, this can cause increased backup times, as a backup is only completed when it has been recorded in the backup catalog and the backup catalog has been backed up.

To address this issue, SAP HANA can accumulate changes to the backup catalog, and back them up together in one operation.

A new backup of the backup catalog would then include all the changes to the backup catalog that were made since the last backup of the backup catalog. Accumulating multiple backups of the backup catalog in this way has the advantage that fewer backups of the backup catalog are created.

Accumulated backups of the backup catalog are supported for both file system backups and third-party backup tools.

12.2.6.1.3.3.1 Disable Writing Accumulated Backups of the Backup Catalog

By default, writing accumulated backups of the backup catalog is enabled. You can disable writing accumulated backups of the backup catalog.

Procedure

1. Locate the parameter `enable_accumulated_catalog_backup` in the `backup` section of `global.ini`.

   For more information, see *Database Configuration in SAP HANA Cockpit*.
2. Open the change dialog.
3. Disable (or enable) writing accumulated backups of the backup catalog.

   To back up the backup catalog after each log backup, set the parameter to **`false`**.

   The default setting is **`true`** (multiple log backups are accumulated to one backup of the backup catalog).

   To reset to the default behavior, choose *Restore Default*.

4. Save.

   Changes take effect immediately.

Related Information

Database Configuration in SAP HANA Cockpit

12.2.6.2 Estimate the Space Needed in the File System for a Data Backup

If there is not enough space in the file system for backups, a backup will fail. To ensure that sufficient free space is available, before you back up the database, you should estimate the amount of space that will be needed.

Context

When you back up an SAP HANA database, the estimated backup size is displayed in the backup dialog in SAP HANA cockpit.

This information is read from table `M_BACKUP_SIZE_ESTIMATIONS`.

For more information, see *M_BACKUP_SIZE_ESTIMATIONS System View* in *SAP HANA SQL and System Views Reference*.

To estimate the space required for a backup, make a note of the space requirement from table `M_BACKUP_SIZE_ESTIMATIONS`, and use the SQL statement `BACKUP CHECK` to check that this amount of space is available in the backup destination.

For more information, see *BACKUP CHECK Statement (Backup and Recovery)* in *SAP HANA SQL and System Views Reference*.

> **Note**
>
> The actual size of a data backup can be larger or smaller than the estimated size.
>
> For example, if data is changed in the database after the size has been estimated and before the backup is created, the actual backup size may be different from the estimated size.
>
> If existing backups are overwritten by backups with the same names, at least **twice the space** in the backup location is needed, because the old backup and the new backup exist for a time in parallel.
>
> It is therefore recommended to keep some additional free space in reserve.

Related Information

M_BACKUP_SIZE_ESTIMATIONS System View
BACKUP CHECK Statement (Backup and Recovery)
Create Data Backups (SAP HANA Administration with SAP HANA Cockpit)

12.2.6.3 Backup Audit Actions for Security

You can audit the creation and cancelation of a backup.

When an action occurs, the audit policy is triggered and an audit event is written to the audit trail. Audit policies are database-specific.

For more information, see the SAP HANA Security Guide.

Related Information

Audit Policies (SAP HANA Security Guide for SAP HANA Platform)
Create an Audit Policy (SAP HANA Administration with SAP HANA Cockpit)
CREATE AUDIT POLICY Statement (Access Control)
CREATE AUDIT POLICY Statement (Access Control) (SAP HANA SQL Reference Guide for SAP HANA Platform)

12.2.6.4 Create a Data Snapshot (Native SQL)

You can create a data snapshot of an SAP HANA database system with one or more tenant databases. You create a data snapshot using SQL.

Prerequisites

- You need the `BACKUP ADMIN` or `BACKUP OPERATOR` system privilege.
  For more information about authorizations, see *Authorizations Needed for Backup and Recovery* in the *SAP HANA Administration Guide for SAP HANA Platform (SAP HANA Database Backup and Recovery)*.
- A data snapshot can only be created through the system database.
  It is not possible to create a data snapshot for a tenant database separately.
- The SAP HANA database (the system database and all the tenant databases) is online, and all the configured services are running.
  The system status for the system database and the tenant database is `System Running`. This is shown on the *Services* card in the *Database Overview*.

Context

A data snapshot is created in three steps that are performed in the SAP HANA database and at storage system level.

Step to Create a Data Snapshot	Description
Prepare the database for the data snapshot.	An **internal database snapshot** is created that reflects a consistent database state at the point in time it is created in the file system. **Note** If an internal database snapshot exists, no new data backups or new data snapshots can be created. Conversely, while a data backup is running, you cannot create a data snapshot.
Create the data snapshot.	The data snapshot is created based on the previously created internal database snapshot. **Note** To ensure its consistent state, the **data snapshot** relies on the previously created **internal database snapshot**. If the database or a database service is restarted, the **internal database snapshot** is lost. At this stage, the data snapshot is in the SAP HANA data area. To be able to create further data snapshots or data backups, you need to manually make all the content of the data area available in a separate storage location. and then confirm the data snapshot. → Remember Data snapshots only offer increased data safety if they are moved or replicated to a separate storage medium. The files and directories under the mountpoint of the data area must all be stored together. The data volumes themselves must not be moved.

Step to Create a Data Snapshot	Description
Confirm or *Abandon* the data snapshot.	When the data snapshot was successfully created, you must confirm or abandon it to be able to create further data snapshots or data backups. When a data snapshot is confirmed, it is recorded in the backup catalog as successful. When a data snapshot is abandoned, it is recorded in the backup catalog as unsuccessful. The data snapshot is always **recorded in the backup catalog** - even if the internal database snapshot is lost before the data snapshot is confirmed. When the data snapshot is confirmed, you are notified whether the data snapshot can be used for a recovery. **Note** If the confirm fails, the database snapshot is marked as unsuccessful. You should physically delete the data snapshot because it may not be possible to use it for recovery.

Procedure

To execute SQL statements, you can use the SQL console in SAP HANA cockpit. (Choose *Open SQL Console* from the *Database Overview*.)

1. Create a new internal database snapshot.

   From the system database, execute the following SQL statement:

   ```
   BACKUP DATA FOR FULL SYSTEM CREATE SNAPSHOT [COMMENT <STRING>];
   ```

   Optionally, add a comment. This comment can help to identify the data snapshot in the backup catalog.

   Note

   `FOR FULL SYSTEM` is mandatory to create a snapshot.

   Sample Code

   ```
   BACKUP DATA FOR FULL SYSTEM CREATE SNAPSHOT COMMENT 'SNAPSHOT-2019-10-22';
   ```

   For more information, see *BACKUP DATA CREATE SNAPSHOT Statement (Backup and Recovery)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

   An **internal database snapshot** is now created ('prepared').

> **Note**
>
> The snapshot will be created for the entire SAP HANA database, that is, the system database and all the tenant databases. It is not possible to create snapshots for an individual tenant database or only for the system database.

2. Find out the backup ID of the **internal database snapshot** in the state `PREPARED`.

   > **Note**
   >
   > SAP HANA cannot ensure that the backup ID of the data snapshot of the system database and the backup ID of the tenant database are the same. As data snapshots are administered by the system database, you must use the backup ID of the **system database** to create the data snapshot.

   Use the following SQL statement:

   ```
   SELECT * FROM M_BACKUP_CATALOG WHERE ENTRY_TYPE_NAME = 'data snapshot';
   ```

   > **Sample Code**
   >
   > ```
   > SELECT BACKUP_ID, COMMENT FROM M_BACKUP_CATALOG WHERE ENTRY_TYPE_NAME
   > = 'data snapshot' AND STATE_NAME = 'prepared' AND COMMENT =
   > 'SNAPSHOT-2019-10-22';
   > ```

   Make a note of the backup ID.

   > **Note**
   >
   > Older internal database snapshots may exist in the state `successful` or `unsuccessful`.

   The database is now prepared for the data snapshot.

   An **internal database snapshot** is created, reflecting a consistent database state at the point in time it is created.

   > **Note**
   >
   > If an internal database snapshot exists, no new data backups or new data snapshots can be created.
   >
   > Conversely, while a data backup is running, you cannot create a data snapshot.

   At this stage, all the snapshot-relevant data is only stored in the data area. To be able to use the data snapshot for a recovery later on, this data needs to be stored in a separate location.

3. In the storage system, make all the content of the data area available together in a separate storage location.

   To create the data snapshot, you can use the tool provided by your storage vendor. For more information, consult the tool documentation.

   > **Note**
   >
   > A data snapshot contains all the persisted data in the data area. For this reason, the files and directories under the mountpoint of the data area must all be stored together.

→ Tip

For a recovery using a data snapshot, only the data area must be restored from the storage tool. You still can use the log area for the recovery.

ⓘ Note

The directory name of the data area is defined by configuration parameter `basepath_datavolumes` in the `global.ini` configuration file, in the `persistence` section.

After the data snapshot has been created in a separate storage location, it needs to be confirmed.

4. *Confirm* or *Abandon* the data snapshot.

   Use the following SQL statement:

Option	Description

Confirm	`BACKUP DATA FOR FULL SYSTEM CLOSE SNAPSHOT BACKUP_ID <BACKUP_ID> SUCCESSFUL <STRING>;`
	
	Confirm that the data snapshot has been successfully saved to a new storage location.
	
	You can specify an external ID to identify the data snapshot later in the storage system.
	
	{·} Sample Code
	
	`BACKUP DATA FOR FULL SYSTEM CLOSE SNAPSHOT BACKUP_ID 1489592445498 SUCCESSFUL 'SNAPSHOT-2019-10-22';`

Abandon	`BACKUP DATA FOR FULL SYSTEM CLOSE SNAPSHOT BACKUP_ID <BACKUP_ID> UNSUCCESSFUL [<STRING>];`
	
	If the data snapshot cannot be created, or if confirmation fails, choose *Abandon*.
	
	Optionally, you can add a comment to explain why the data snapshot was not successful.
	
	{·} Sample Code
	
	`BACKUP DATA FOR FULL SYSTEM CLOSE SNAPSHOT BACKUP_ID 1489592445498 UNSUCCESSFUL 'SNAPSHOT-2019-10-22 FAILED';`

For more information, see *BACKUP DATA CLOSE SNAPSHOT Statement (Backup and Recovery)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

→ Tip

It is strongly recommended to confirm or abandon a data snapshot **as soon as possible after it has been created**.

While the data snapshot is being prepared or created, the snapshot-relevant data is frozen. While the snapshot-relevant data remains frozen, changes can still be made in the database. Such changes will not cause the frozen snapshot-relevant data to be changed. Instead, the changes are written to

positions in the data area that are separate from the data snapshot. Changes are also written to the log.

However, the longer the snapshot-relevant data is kept frozen, the more the data volume can grow.

> ⓘ Note
>
> If the database or an individual database service is restarted, the **internal database snapshot** is lost. If the database snapshot is lost before the data snapshot is confirmed, the data snapshot is still written. During confirmation, the database notifies you that the data snapshot cannot be used.

After you have confirmed or abandoned a data snapshot, it is recorded in the backup catalog as either successful or unsuccessful.

> ⓘ Note
>
> A data snapshot now exists for both the system database and the tenant database.

The internal database snapshot that was used to create the data snapshot is discarded.

It is now possible to create further data snapshots or data backups.

Related Information

Authorizations Needed for Backup and Recovery [page 938]
Data Snapshots [page 945]
BACKUP DATA CREATE SNAPSHOT Statement (Backup and Recovery)
BACKUP DATA CLOSE SNAPSHOT Statement (Backup and Recovery)

12.2.7 SAP HANA Recovery

It may be necessary to recover an SAP HANA database due to a number of different reasons.

- Data is unusable (disaster recovery)
  - The data area is unusable.
    For more information, see *Data Area is Unusable (Disaster Recovery)*.
  - The log area is unusable.
    For more information, see *Log Area is Unusable (Disaster Recovery)*.
- A logical error has been detected (fault recovery)
  If a logical error occurs, the database needs to be recovered to its state at a particular point in time.
  For more information, see *Logical Error - Point-in-Time Recovery (Fault Recovery)*.
- You want to create a copy of the database.
  For more information, see *Copying a Database Using Backup and Recovery* in *SAP HANA Administration with SAP HANA Cockpit (Backup and Recovery)*.

> **Note**
>
> If using the local secure store (LSS), you need the root keys backup password for a successful SAP HANA database recovery.

Related Information

Data Area is Unusable (Disaster Recovery) [page 1059]
Log Area is Unusable (Disaster Recovery) [page 1060]
Logical Error - Point-in-Time Recovery (Fault Recovery) [page 1061]
Copying a Database Using Backup and Recovery [page 1065]
Prerequisites for Database Recovery

12.2.7.1 Checking Whether a Recovery is Possible

The success of a database recovery can only be ensured if the required backups are available and have not been changed since they were created. For this reason, it is recommended that you check backups periodically, or if you suspect that they have been changed in some way since they were created.

Automatic Checks for Backups

When SAP HANA data backups, delta backups, or log backups are created, the integrity of the data to be backed up is automatically checked while the backups are being written. These integrity checks are performed on block level. However, the actual content of the data blocks is not analyzed. The backup is written only if the integrity check was successful.

When a recovery is started, the block-level integrity of the backups to be used is automatically checked. If an error is detected, the recovery is stopped, and will need to be repeated.

Manual Checks for Recovery

In addition to the automatic backup checks performed by SAP HANA, you can manually check data backups and log backups **without performing a recovery**.

You can check the following:

Check...	Using...
Can a database can be recovered to a specific point in time?	Before you start a recovery, you can use the *Check Recoverability* option in SAP HANA cockpit. For more information, see *Check Recoverability*. Alternatively, you can use `hdbrecovercheck`. For more information, see *Check Recoverability (hdbrecovercheck)*.
Which backups are required?	`hdbbackupdiag` determines: • Which data backups and log backups are required to recover the database to a specified point in time • Whether these backups are available and can be accessed For more information, see *Check the Backups Required for a Recovery (hdbbackupdiag)*. `hdbbackupdiag` can also be used to rebuild the backup catalog. For more information, see *Rebuilding the Backup Catalog*.
Were individual data backups or log backups changed since they were created?	Use `hdbbackupcheck` For more information, see *Checking Individual Backups (hdbbackupcheck)*. For more information, see SAP Note 1869119 *(Checking backups with "hdbbackupcheck")* **Restriction** `hdbbackupcheck` does not support SAP HANA Dynamic Tiering.

Note

`hdbrecovercheck`, `hdbbackupdiag`, and `hdbbackupcheck` can all be used with file system backups and third-party backup tools.

`hdbbackupdiag` and `hdbbackupcheck` cannot be used with data snapshots.

With third-party backup tools, `hdbbackupdiag` and `hdbbackupcheck` must run in a system with the same SID to which the backups were written.

Isolation Level High for Backups and Third-Party Tools

To grant the system administrator access to the tenant database backup files and directories, you need to add the `<sid>adm` user to the operating system group of each tenant. For more information, see *File and Directory Permissions with High Isolation* in the *SAP HANA Administration Guide*.

Related Information

Check Recoverability
Check Recoverability (hdbrecovercheck) [page 1032]
Check the Backups Required for a Recovery (hdbbackupdiag) [page 1036]
Checking Individual Backups (hdbbackupcheck) [page 1042]
SAP Note 1869119
File and Directory Permissions with High Isolation [page 83]
Isolation Level High for Backups and Third-Party Backup Tools [page 979]
Rebuilding the Backup Catalog [page 962]

12.2.7.1.1 Check Recoverability (hdbrecovercheck)

Before you start a recovery, you can use the Python script `hdbrecovercheck.py` to verify that a database can be recovered to a specific point in time.

Prerequisites

- Python support
  In a standard SAP HANA installation, `hdbrecovercheck.py` is installed in `$DIR_EXECUTABLE/python_support`.
- For file-based backups: If the directories that you want to use are different from the directories specified in the backup catalog, you need to manually specify the directories that contain the backup catalog, the data backups, and the log backups.
  You can opt to ignore the log area (log segments that have not been backed up) or delta backups.
  If no backup paths are specified, `hdbrecovercheck.py` uses the paths specified in the backup catalog.

> ⚠ Restriction
>
> `hdbrecovercheck.py` does not recognize data snapshots, and does not support system copy or system replication.

Context

`hdbrecovercheck.py` checks all the resources required (backups and log area) for a recovery to a specific point in time to verify whether they:

- Exist
- Can be accessed

`hdbrecovercheck.py` also gives you the option to check the consistency of the backups on:

- Volume level
- Page level
- All hosts for all services

You can check the consistency of the backups up to a point in time without actually verifying that a recovery to that point in time is possible.

Procedure

1. Use the following command to call `hdbrecovercheck.py`:

Option	Description
Backup Catalog is in the File System:	`python python_support/hdbrecovercheck <option> -c <backup catalog path> --database <name>`
Backup Catalog is in Backint:	`python python_support/hdbrecovercheck <option> -b --database <name>`

   To display all the options, use the command: `python python_support/hdbrecovercheck.py -h`

2. (Optional) To check the consistency of the backups, use the following command: `hdbrecovercheck.py checkBackupConsistency [-h]`.

   Select the following options:

Option	Description
`-D DATABASE`	Specify the name of the database
`-c BACKUPCATALOGPATH`	Specify the path to backup catalog in the file system
`-b`	Use the backup catalog from the third-party backup tool (Backint)
`-t TIMESTAMP`	Timestamp to be reached, in UTC Default: now

Option	Description
`-l LOGBACKUPPATHS`	Specify the log backup paths as a comma-separated list
`-d DATABACKUPPATHS`	Specify the data backup paths as a comma-separated list
`-v, --verbose`	Verbose output
`--json`	json output

Example

```
python python_support/hdbrecovercheck.py checkBackupConsistency
-c /usr/sap/HD3/HDB03/backup/log/DB_HD3 --database HD3
```

For the database named HD3, checks the consistency of the log backups specified in the file-based backup catalog needed for a recovery until now.

3. To check whether the backups can be used to recover the database to the desired point in time, use the following command: `hdbrecovercheck.py checkPointInTimeReachable [-h]`.

   Select the following options:

Option	Description
`-D DATABASE`	Specify the name of the database
`-c BACKUPCATALOGPATH`	Path to backup catalog in the file system
`-b`	Use the backup catalog from the third-party backup tool (Backint)
`-t TIMESTAMP`	Specify the timestamp to be reached, in UTC Default: now
`-i`	Ignore the log area
`-I`	Ignore delta backups
`-l LOGBACKUPPATHS`	Specify the log backup paths as a comma-separated list
`-d DATABACKUPPATHS`	Specify the data backup paths as a comma-separated list
`-v`	Verbose output
`--json`	json output

Example

```
python python_support/hdbrecovercheck.py checkPointInTimeReachable
-c /usr/sap/HD3/HDB03/backup/log/DB_HD3 --database HD3 -t "2020-09-02
10:10:30" -v
```

For the database named HD3, checks the backups specified in the file-based backup catalog to determine whether the backups can be used to recover to the desired point in time (2020-09-02 at 10:10:30 UTC), with verbose output.

Results

The backups and the log area are checked.

Trace data (`hdbrecovercheck.trc`) is written to the SAP HANA trace directory.

For more information about viewing trace files, see *View Diagnostic Files in the SAP HANA Database Explorer*.

Options for a Recovery

You are notified as to what options are available for the recovery:

Result	Option
The specified point in time is in the current backup history or log area.	You can proceed with the recovery.
The specified point in time is missing from the current backup history, but it is recorded in the backup log.	Action: Check the backup log. You may be able to recover to the point in time using the relevant log positions.
The point in time cannot be reached.	This could mean that a specified path is incorrect or that a backup is corrupt at the specified path, but backups at a different location may be still usable. Action: • Specify the correct backup paths. or • Recover to a different point in time.
The backup history is lost. If log mode overwrite was used, log segments may have been overwritten.	The database can be recovered without log replay.
(If there is no log backup) The log area and the data backup are incompatible.	1. Specify the path to a different backup catalog containing the correct data backup(s). 2. You can recover the database without log replay. (Initialize the log area.)

Result	Option
The log backups and the log area are incompatible.	1. Specify the path to a different backup catalog containing the correct log backups. 2. You can recover the database without log replay. (Initialize the log area.)

Related Information

View Diagnostic Files in the SAP HANA Database Explorer

12.2.7.1.2 Check the Backups Required for a Recovery (hdbbackupdiag)

The `hdbbackupdiag` tool determines which backups are required to complete a recovery to a specified point in time. It also checks whether these backups are available and whether they can be accessed.

Context

With `hdbbackupdiag`, you can verify the following:

For file-based backups:

- The backup is available in the file system, either at the location to which it was written or at a location specified by a search path.
  The backups to be used can be in any directory in the file system.
- The current operating system user has read authorization for the file.
- The actual size of the backup file is the same as the size recorded in the backup file header.
- The backup ID is identical to the backup ID specified in the backup catalog.

For backups created using a third-party backup tool:

- The backup is available in the third-party backup tool.

> → Tip
>
> To maintain good recovery performance, and to allow the check to be completed quickly, `hdbbackupdiag` checks only the metadata of a backup. It does not check the integrity of the backup content on block level.
>
> It is recommended that you use this tool periodically to check the consistency of the metadata of a backup.

Procedure

1. Ensure that `hdbbackupdiag` can locate the backup catalog.

   To do this, you can either execute `hdbbackupdiag` in the directory where the backup catalog is located (file-based only) or execute `hdbbackupdiag` with options that specify which directories to search for the latest backup catalog. By default, this is the directory where the last log backups were written before the recovery was started.

   The default directory is `$DIR_INSTANCE/backup/log`.

   For more information, see *Change the Log Backup Settings*.

2. Start `hdbbackupdiag`.

   Use the following command:

   ```
   hdbbackupdiag [options] [-d <directory>]
   ```

   The options for `hdbbackupdiag` are described below.

Option	Description
`-h | --help`	Display the available options.
`--check`	Check: • Whether all the backups (including log backups and delta backups) required for a recovery exist and can be accessed • The metadata is consistent The `hdbbackupdiag` tool first computes the recovery strategy in accordance with the input parameters and builds a list of backups that will be needed for the recovery. `hdbbackupdiag` then checks whether all the backups needed are available and can be accessed. To verify that all the backup metadata is consistent and has not changed since the backup was made, `hdbbackupdiag` then also checks that the backup headers are correct. **Note** The `hdbbackupdiag` tool (and the SQL statement BACKUP CHECK ACCESS) does **not** check the persisted content of the backup. To check the backup content, you can use the `hdbbackupcheck` tool for each backup listed by `hdbbackupdiag`. For more information about `hdbbackupcheck`, see *Manually Checking Whether a Recovery is Possible*.
`-f`	Display the names of the backups required for recovery as a simple list. From this list, the backup names can be easily included in shell scripts.
`-B`	Display the names of backups with Backint information.
`-v`	Display all available information.

Option	Description
	For example, the SAP HANA version that was used to create a backup.
`-d <directory>`	Specify the directory to search for the backup catalog. If you do not specify a directory, the current directory is searched for the latest version of the backup catalog. If specified, the directories indicated with `--logDirs` (see below) and the third-party backup tool are also searched. **Note** All directories must be specified as absolute paths.
`-c <catalog>`	Specify the name of the backup catalog.
`-i <BackupID>`	Specify a backup ID. If you do not specify a backup ID, the most recent usable data backup is used.
`-u <"YYYY-MM-DD hh:mm:ss>"`	Specify a target time for the recovery (UTC time). If you do not specify a time, the most recent possible point in time is used.
`--dataDir <directory>`	Specify a directory to search for data backups or delta backups. If you do not specify a directory, only the paths specified in the backup catalog are searched.
`--logDirs <directories>`	Specify a comma-separated list of directories to search for log backup files. If you do not specify this option, only the paths specified in the backup catalog are searched.
`--useBackintForCatalog`	With this option, the third-party backup tool is searched for the most recent version of the backup catalog.

`--databaseName <database>`	Used only with SAP HANA and third-party backup tools.
	This option is used with `--useBackintForCatalog` to specify a tenant database or the system database:
	`--databaseName <name_of_tenant_database>`
	`--databaseName SYSTEMDB`

`--backintDataParamFile <paramFileName>`	Specify a parameter file to access data backups and delta backups through a third-party backup tool.
`--backintLogParamFile <paramFileName>`	Specify a parameter file to access log backups through a third-party backup tool. If you do not specify a parameter file, the parameter file used to access the data backups is used.

Option	Description
`--pickCatalog`	If you want to recover SAP HANA to a point in time (UNTIL timestamp) that is not available in the current timeline, a suitable catalog is selected for the recovery time. More information: See SAP Note: 2050606 (Recover database from not current backup history)
`--generate`	Generate a new backup catalog **⚠ Caution** The `--generate` option is intended for exceptional situations and **file-based backups only**. This option does not support data snapshots or backups created using third-party tools. If you use the `--generate` option, any information about data backups or log backups created using third-party tools will be lost from the newly generated backup catalog. It will then no longer be possible to use this information for a recovery.
`--ignoreDeltaDataBackups`	Exclude delta backups.

Results

The output of `hdbbackupdiag` contains the names of all the files required to recover the SAP HANA database.

Related Information

Change the Log Backup Destination Type [page 1015]
Checking Whether a Recovery is Possible [page 1030]
Verify That Log Backups are Being Consolidated (hdbbackupdiag) [page 1004]

12.2.7.1.2.1 Examples of Output From hdbbackupdiag

The output of `hdbbackupdiag` contains the names of all the files required to recover a SAP HANA database.

If you specify the `--check` option, the results of the metadata checks are also displayed.

Example

```
hdbbackupdiag --check --logDirs /usr/sap/HD2/HDB00/backup/log/ --
dataDir /usr/sap/HD2/HDB00/backup/data/
```

With this example command, the following steps are performed:

- `hdbbackupdiag` recursively searches the directory `/usr/sap/HD2/HDB00/backup/data/` for data backups and delta backups.
  If a data backup is not found, `hdbbackupdiag` attempts to check the original backup directory specified in the backup catalog.
- `hdbbackupdiag` recursively searches the directory `/usr/sap/HD2/HDB00/backup/log` for log backups.
- `hdbbackupdiag` checks whether a recovery is possible from the perspective of the backup component.
  The metadata of the backup files is checked to determine whether all the required backups are available and consistent.
  In addition, `hdbbackupdiag` checks whether a recovery to the desired time is possible In this example, this is most recent point in time.

  > **Note**
  >
  > A recovery can still fail due to a persistent page corruption. This would be detected by the `hdbbackupcheck` tool. For more information, see *Manually Checking Whether a Recovery is Possible*.

- SAP HANA decides which data backup in the specified directory to use for the recovery.

This command returns the following output, including some errors:

Sample Code

```
found backup catalog 1426152872410 from backint /usr/sap/HD2/SYS/global/hdb/
backint/log_backup_0_0_0_0
found backup catalog 1426152780165 from file /usr/sap/HD2/HDB00/backup/log/
log_backup_0_0_0_0.1426152780165
using backup catalog 1426152872410 from backint /usr/sap/HD2/SYS/global/hdb/
backint/log_backup_0_0_0_0
Backup '/usr/sap/HD2/HDB00/backup/data/COMPLETE_DATA_BACKUP_databackup_0_1'
successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/data/COMPLETE_DATA_BACKUP_databackup_1_1'
successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/data/COMPLETE_DATA_BACKUP_databackup_2_1'
successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/data/COMPLETE_DATA_BACKUP_databackup_3_1'
successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_1_0_380224_384576.1426152365477' successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_1_0_384576_385216.1426152395479' successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_1_0_385216_385984.1426152425481' successfully checked.
ERROR: [111119] file '/usr/sap/HD2/HDB00/backup/log/
log_backup_1_0_385984_386816.1426152455484' not found
ERROR: Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_1_0_385984_386816.1426152455484' check failed.
(...)
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_265408_271040.1426152365475' successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_271040_271104.1426152396044' successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_271104_271680.1426152440145' successfully checked.
ERROR: Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_271680_272576.1426152500624' has size 45131 bytes, but is
expected to be at least 61440 bytes
```

```
ERROR: [110059] The backup /usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_271680_272576.1426152500624 is corrupt, size is 45131 bytes
ERROR: Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_271680_272576.1426152500624' check failed.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_272576_273152.1426152560136' successfully checked.
Backup '/usr/sap/HD2/HDB00/backup/log/
log_backup_2_0_273152_273792.1426152620143' successfully checked.
(...)
```

The first error occurs because a log backup is not available. The second error is because a log backup does not have the expected size.

Example

Display all the backups required to recover the database until May 11, 2015, 01:05:00 p.m. The metadata of the backups is not checked.

From this list, the backup names can be easily included in shell scripts.

The time specified is UTC time, not local time.

```
hdbbackupdiag -f -d /usr/sap/MBY/HDB01/backup/log -u "2015-05-11 13:05:00"
```

```
2015-05-11_13-05_databackup_0_1
2015-05-11_13-05_databackup_1_1
2015-05-11_13-05_databackup_2_1
2015-05-11_13-05_databackup_3_1
2015-05-11_13-05_databackup_4_1
log_backup_1_0_426304_427776.1431332400000
log_backup_1_0_427776_428288.1431333300000
log_backup_1_0_428288_428608.1431334200000
log_backup_1_0_428608_429376.1431335100000
log_backup_1_0_429376_429696.1431336000000
log_backup_1_0_429696_430464.1431336900000
log_backup_1_0_430464_430784.1431337800000
log_backup_1_0_430784_431552.1431338700000
log_backup_1_0_431552_431872.1431339600000
log_backup_2_0_598080_598592.1431340500000
log_backup_2_0_598592_598976.1431341400000
log_backup_2_0_598976_599360.1431342300000
log_backup_2_0_599360_602304.1431343200000
log_backup_2_0_602304_602688.1431344100000
log_backup_3_0_536064_538304.1431345000000
log_backup_4_0_1190656_1191360.1431345900000
log_backup_4_0_1191360_1191680.1431346800000
log_backup_4_0_1191680_1192000.1431347700000
log_backup_4_0_1192000_1192640.1431348600000
log_backup_4_0_1192640_1192960.1431349500000
```

Related Information

Checking Whether a Recovery is Possible [page 1030]

12.2.7.1.3 Checking Individual Backups (hdbbackupcheck)

It is recommended that you check backups to verify that they have not been changed since they were created. For example, if backups have been transferred to a new storage medium.

Context

You can use the `hdbbackupcheck` tool to check the integrity of individual data backups and log backups for both file-based backups and backups created using third-party backup tools.

`hdbbackupcheck` can be used either from inside an SAP HANA installation, or from outside an SAP HANA installation to check backups that are not accessed by an SAP HANA node.

> → Tip
>
> Even if the metadata of a backup is correct, the backup may still have internal errors. For this reason, we recommend that you use `hdbbackupcheck` to check for corruption in individual data or log backups.

Related Information

Check Individual Backups Inside an SAP HANA Installation (hdbbackupdiag, hdbbackupcheck) [page 1042]
Check Individual Backups Outside an SAP HANA Installation [page 1048]
SAP Note 1869119

12.2.7.1.3.1 Check Individual Backups Inside an SAP HANA Installation (hdbbackupdiag, hdbbackupcheck)

Manually check the integrity of individual data backups and log backups.

Context

To find out which backup files are needed for a recovery, you can use `hdbbackupdiag`.

For more information, see *SAP Note 1821207 (Determining required recovery files)*.

A backup of an SAP HANA instance consists of multiple parts, each with the same prefix. A part of a backup is a backup file in the system storage or a backup object that has been transferred to an external backup storage system.

To check a backup, execute `hdbbackupcheck` for each backup individually.

> **Note**
>
> If you are working with third-party backup tools, consult the tool documentation to learn more about the backup checks that these tools offer.

Procedure

1. Identify the backups that you want to check.

   Use the `hdbbackupdiag` tool.
2. To check the backups, you need the following information:

   - File name (*Location*)
     For file-based backups, the location is the file system path to the backup. If the backup is below the current directory, the relative path can be used.
     For backups managed using a third-party backup tool, the location is the complete path and name, beginning with `/usr/sap/<SID>/SYS/global/hdb/backint/`.
   - External backup ID
     You need the external ID if you are using a third-party backup tool.
   - Optionally, the backup ID assigned by the SAP HANA database when the backup was created.
3. From the command line, call `hdbbackupcheck` for each backup file.

   ```
   hdbbackupcheck [options] <backup>
   ```

   `hdbbackupcheck` offers the following options:

   Options for hdbbackupcheck

Option	Description
`-h`	Display available options.
`-e <ebid>`	You need to specify the external backup ID (`ebid`) if part of the backup is in a third-party backup tool. **Note** If a backup is in a third-party backup tool, you need to specify **both** the external backup ID and the database name, using `--databaseName <database name>`. **Note** `hdbbackupcheck` does not support encrypted and compressed backups for external backup IDs.

Option	Description
`-i <backupid>`	Check whether the backup has the backup ID specified. The backup ID is assigned to the backup when it is created.
`-o <filename>`	Specify the output file to use (together with `--exportLssBackup o <filename>`).
`-p <directory>`	By default, the log files `backupcheck.log` and `backintcheck.log` are created in the trace directory. To create the log files in a different directory, call `hdbbackupcheck` with option `-p <directory>` and specify the directory.
`-t`	Enable tracing. A trace file is created: • If run within the SAP HANA environment: In the SAP HANA trace directory • If run outside the SAP HANA environment: In `./trace`
`-v`	Display the header data of the backup.
`--backintParamFile <filename>`	Specify the parameter file for the third-party backup tool (Backint). You can also specify a parameter file for encrypted log backups and encrypted and unencrypted data backups, even if SAP HANA is not currently running. If the working directory is not the directory where the file is located, specify the absolute path. To find out this path, consult the documentation provided by the tool vendor. To specify a Backint parameter file, you must also specify the database name using the option `--databaseName <database name>`.
`--backintVersion <backint_version>`	Display the version of the SAP HANA Backint protocol version used to communicate with the third-party backup tool. Default: Version 1.0

Option	Description
`--databaseName <database name>`	You need to specify the name of the database if: • Part of the backup is in a third-party backup tool • `-e <ebid>` is specified with a relative `<backup>` path • You want to dump an encrypted log backup or data backup using the option --dump The name of the database is the name used in the CREATE DATABASE statement. **Note** If a backup is in a third-party backup tool, you need to specify **both** the database name (using the option --databaseName <database_name>) and the external backup ID (`ebid`).
`--dump <backup file>`	List the contents of the backup, if possible. For example, with a metadata backup, the `--dump` option displays the topology information and the `.ini` files, but does not display the content of the metadata. To list the contents of encrypted log backups or encrypted data backups, you need to also specify the database name using the option --databaseName <database_name>. You can do this both if the SAP HANA database is running or not running.
`--dump -c <.ini file>`	List the contents of a specified configuration file (`.ini` file) that is included in the metadata of a data backup. For example: `hdbbackupcheck --dump -c indexserver.ini demo_databackup_0_1` For more information about including user configuration files in data backups, see *BACKUP DATA Statement (Backup and Recovery)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.
`--exportLssBackup o <filename>`	Write the content of the LSS backup in the specified backup to the file specified.

Option	Description
`--usingSource <source spec>`	Specify the source system in the format: **`<database name>@<SID>`** **Note** The **`<backup>`** specified must be the backup of the source system.
`<backup>`	Name of the backup file.

Results

`hdbbackupcheck` notifies you if any errors were detected in the checked part of the backup.

Related Information

SAP Note 1821207
BACKUP DATA Statement (Backup and Recovery) (SAP HANA SQL Reference Guide for SAP HANA Platform)

12.2.7.1.3.1.1 Examples of Output From hdbbackupcheck

`hdbbackupcheck` notifies you if any errors were detected in the checked part of the backup.

If no errors were detected, `hdbbackupcheck` returns 0.

If an error was detected, `hdbbackupcheck` returns 1.

Below are some examples of output from `hdbbackupcheck`:

Example

```
hdbbackupcheck backup/data/BackupTestMaster_databackup_1_1
```

Output Code

```
Backup '/hana/shared/BNR/HDB00/backup/data/BackupTestMaster_databackup_1_1'
successfully checked.
```

Example

```
hdbbackupcheck -v backup/data/BackupTestMaster_databackup_1_1
```

Output Code

```
Check backup '/hana/shared/BNR/HDB00/backup/data/
BackupTestMaster_databackup_1_1'
Check backup '/hana/shared/BNR/HDB00/backup/data/
BackupTestMaster_databackup_1_1'.
Destination header information:
DestVersion: 5
DatabaseID: 51a3a622-1627-46c8-e100-00000a1d0eab
InternalStartTime: 1370415876795
CurrDestInformation: [FILE][/usr/sap/BNR/HDB00/backup/data/
BackupTestMaster_databackup_1_1]
backupID: 1370415876776
ServiceName: nameserver
NumberOfVolumeFiles: 4
HostName: berl30052174a
VolumeID: 1
DestID: 1
MaxDestID: 1
SrcPoolInformation[0]: [DATABASE_SNAPSHOT]@node[1]
DstPoolInformation[0]: [FILE][/usr/sap/BNR/HDB00/backup/data/
BackupTestMaster_databackup_1_1]
Source header information:
SrcType: 1
SourceInformation: [DATABASE_SNAPSHOT]@node[1]
srcVersion: 5
sourceSize: 70455296
```

Example

Check a data backup that was written to the file system and at some stage was corrupted:

```
hdbbackupcheck backup/data/Hallo_databackup_2_1
```

Output Code

```
ERROR: [110088] Error reading backup from 'FILE' '/hana/shared/BNR/HDB00/
backup/data/Hallo_databackup_2_1'
ERROR: [110059] The backup /hana/shared/BNR/HDB00/backup/data/
Hallo_databackup_2_1 is corrupt, size is 14807859
ERROR: Backup '/hana/shared/BNR/HDB00/backup/data/Hallo_databackup_2_1' not
successfully checked!
```

Example

Check a log backup that was saved to a third-party backup tool using the configuration file `/myBackupTool/backint.cfg`:

```
hdbbackupcheck --backintParamFile /myBackupTool/backint.cfg /usr/sap/BNR/SYS/
global/hdb/backint/log_backup_1_0_2177088_2177344 -e BCKINTk168Gc
```

Output Code

```
Backup '/usr/sap/TG2/SYS/global/hdb/backint/log_backup_1_0_2177088_2177344'
successfully checked.
```

12.2.7.1.3.2 Check Individual Backups Outside an SAP HANA Installation

You can use the `hdbbackupcheck` tool to manually check the integrity of backups **outside** an SAP HANA installation.

Context

You can use `hdbbackupcheck` to check backups that are not accessed by an SAP HANA node, without generating additional load on the SAP HANA node.

> ### Note
>
> Using `hdbbackupcheck` **outside** an SAP HANA installation is recommended only for file-based backups.

Procedure

1. In the SAP HANA installation, create an archive with the required files: `hdbbackupcheckpack <archive>`

   > ### Note
   >
   > The archive created here contains only the test software and not the data backups to be tested.

2. Move the archive `<archive>` to the target system and unpack it:
   a. Create a directory `<targetdir>` in the target system.
   b. Copy the archive `<archive>` and the program `$DIR_INSTANCE/exe/SAPCAR` to the directory `<targetdir>` in the target system.
3. Unpack the archive, add the directory of `hdbbackupcheck` to the environment variable `<LD_LIBRARY_PATH>`, and call the program as described above:
   a. `cd <targetdir>`
   b. `./SAPCAR -xvf <archive>`
   c. `export LD_LIBRARY_PATH=<targetdir>:$LD_LIBRARY_PATH`
   d. `./hdbbackupcheck -v <backup>`

   > ### Note
   >
   > By default, the files `backupcheck.log` and `backintcheck.log` are created in the current directory. To create these files in a different directory, start `hdbbackupcheck` with option `-p <directory>`.

12.2.7.1.4 Prerequisites: Recovery Using Multistreamed Backups

For a recovery using multistreamed backups, there needs to be the same number of channels that were used for the backup.

When you prepare a recovery from multistreamed backups, you do not need to take any special steps to adjust the number of channels used. During a recovery, SAP HANA is able to recognize how many channels were used for a backup, and automatically uses this number of channels for the recovery.

SAP HANA does **not** check the value of parameter `parallel_data_backup_backint_channels`, which defines the number of channels used for multistreaming for both data backups and delta backups.

The backup catalog shows all the parts of a multistreamed backup. For a recovery, the order of the backup parts is not important. SAP HANA can recover the parts of a multistreamed backup in any order.

Related Information

Configure Multistreaming with Third-Party Backup Tools [page 976]
Configure SAP HANA Parameters for a Third-Party Backup Tool [page 972]

12.2.7.2 Recover SAP HANA From a Data Snapshot

Using SAP HANA cockpit and storage system tools, you can recover a complete SAP HANA system (the system database and all its tenant databases) from a data snapshot.

Prerequisites

> ⚠ Caution
>
> Unlike a database recovery from data backups and log backups, the process of recovering SAP HANA from a data snapshot is not entirely under the control of SAP HANA. Manual steps are necessary, for example, to make available the data snapshot in the correct location before the recovery can begin.
>
> To ensure that it succeeds, recovering SAP HANA from a data snapshot requires expert knowledge.
>
> If you are in doubt as to how to ensure that the prerequisites for a recovery from a data snapshot are met, we recommend that you instead consider recovering SAP HANA using a standard recovery from data backups and log backups.

The corresponding SQL statements to recover data from a snapshot are:

- RECOVER DATA - recovers the SAP HANA database using a specific data backup
- RECOVER DATABASE - recovers an SAP HANA database to a specific point in time or to a specific log position

These are described in detail in the *SAP HANA SQL Reference Guide* - see Related Links below.

Before you Start

- A data snapshot cannot be used to create a copy of a lower SAP HANA version.
- While a data snapshot is being created, no data integrity checks are performed by SAP HANA (checksum calculation) on page or block level.
  This means that, if SAP HANA database blocks were corrupted when the data snapshot was created, SAP HANA will only detect this when the database is started. It may then not be possible to complete the recovery.

During the Recovery

- To recover SAP HANA from a data snapshot, you need to shut down the target SAP HANA database, then make available the data snapshot in the data area of the target database.
- Recover the system database first. When the system database has been recovered successfully, recover each tenant database separately.
  The tenant databases cannot be recovered together in one single operation.
  These steps are described in *Recover all Tenant Databases From a Data Snapshot*.
- To recover a database from a data snapshot, you can optionally also use delta backups and log backups.

Procedure

First, recover the **system database** from the data snapshot.

1. From the *Database Directory*, open the system database and shut down the database.
2. From the *Database Overview*, then choose *Stop System*.
3. Follow the on-screen instructions to shut down the database.

   The database is shut down.

   In *Manage Services*, the system status is shown as *Stopped*.
4. Go to the *Database Overview*.
5. Outside of SAP HANA cockpit, make the data snapshot (the content of the data area) available in the data area of the storage system.

   The data snapshot must be accessible by the `<SID>adm` user.
6. From SAP HANA cockpit, go to the *Database Overview*, and choose *Recover database* from the *Database Backups* card.
7. Specify the recovery target.

   You can recover the database to its most recent state or specify a point in time to which to recover the database.
8. Specify the location of the most recent backup catalog.

Option	Description
Backint location only	If a third-party backup tool is selected, Backint is searched.
Default location	For file system backups, the location for the backups of the backup catalog for the system database is defined using the parameter `basepath_catalogbackup`. The default setting for `basepath_catalogbackup` is: `$DIR_INSTANCE/backup/log` By default, log backups for tenant databases are written to a tenant-specific subdirectory. By default, backups of the backup catalog are written to the same tenant-specific subdirectory as the log backups.
Alternative file system location	If the backup catalog is not in the default location, specify its location. For more information, see *Destination for Backups of the Backup Catalog* in the *SAP HANA Administration Guide (SAP HANA Database Backup and Recovery)*.

An overview of available backups is displayed.

9. Select the data snapshot.
10. Specify whether to use delta backups.

> **Note**
>
> To recover SAP HANA using delta backups (differential or incremental backups), you must also use log backups. If log backups are not available, you can only recover using a full data backup.

11. Specify the location of the log backups, if available.
12. To ensure that a backup exists at the specified location, check the availability of the backups.
13. If necessary, initialize the log area.

> **Caution**
>
> Initializing the log area may cause significant loss of data.

If you initialize the log area, the content of the log area is lost. No records from the log area can then be replayed. The records from the **log backups** are replayed if they are needed.

In the following situations, you **must** initialize the log area:

- The log area is unusable
- You are recovering the database to a different system

14. To recover the system database, choose *Start Recovery*.

The progress of the recovery is displayed.

When the recovery is completed, a message confirms this, and shows the point in time to which the database was recovered.

The system database is now online, but cannot yet be used by applications.

You should now first check that the system database was recovered successfully, then restart the tenant databases manually.

For more information, see *Recover all Tenant Databases From a Data Snapshot*.

Related Information

Recover all Tenant Databases From a Data Snapshot [page 1052]
Data Snapshots [page 945]
Create a Data Snapshot (Native SQL) [page 1024]
Recovering an SAP HANA Database
Authorizations Needed for Backup and Recovery [page 938]
Destination for Backups of the Backup Catalog [page 1020]
RECOVER DATABASE Statement (Backup and Recovery)
RECOVER DATA Statement (Backup and Recovery)
SAP Note 2340161

12.2.7.2.1 Recover all Tenant Databases From a Data Snapshot

When the system database has been recovered successfully from a data snapshot, you need to recover all the tenant databases from the same data snapshot.

Prerequisites

- The system database has been successfully recovered.
- The data snapshot that was used to recover the system database remains available in the data area of the storage system.

Context

When the system database has been recovered successfully from a data snapshot, only the system database is started automatically. When you have checked that the system database was recovered correctly, you need to then recover each tenant database separately.

Procedure

1. From the *Database Directory*, select the recovered system database and go to the *Database Overview*.
2. Choose *Database Management*.

   An overview of the databases in the SAP HANA system is displayed.
3. Select a tenant database.
4. Choose *Recover Tenant*.

   Follow the on-screen instructions.
5. To recover the tenant database, choose *Start Recovery*.

   The progress of the recovery for each SAP HANA service is displayed.

   When the recovery is completed, a message confirms this, and shows the point in time to which the database was recovered.

   The SAP HANA tenant database is now online and can be used by applications.

   To recover the remaining tenant databases, repeat the steps for each tenant database.

Related Information

Recover SAP HANA From a Data Snapshot [page 1049]

12.2.7.3 Recover a System Database to a Point in Time (recoverSys.py)

To recover an SAP HANA database, it is strongly recommended that you use SAP HANA cockpit. Using SAP HANA cockpit, it is only possible to recover a system database to its most recent state. To recover a system database to a point in time, you can use the `recoverSys.py` tool.

Prerequisites

> **Note**
>
> The `recoverSys.py` tool can only be used to recover a system database. `recoverSys.py` cannot be used to recover a tenant database. To recover a tenant database, use SAP HANA cockpit or native SQL.

> **Caution**
>
> If you recover a system database to a point in time, tenant databases that were registered after that point in time will be lost.

- The system database is offline.
- You are logged with the OS user `<sid>adm`.

Context

To call `recoverSys.py`, enter the statement in the following format: `HDBSettings.sh recoverSys.py [<parameters>]`

If you run `HDBSettings.sh recoverSys.py` without any parameters, `recoverSys.py` performs a recovery to the most recent point in time.

> **Note**
>
> Starting `recoverSys.py` on its own does not do anything.

Procedure

To recover a system database:

1. Set the environment using `HDBSettings.sh`.
2. Execute the `recoverSys.py` tool: `HDBSettings.sh recoverSys.py [<parameters>]`.

   `recoverSys.py` shuts down the database.

Results

Once the system database has started, `recoverSys.py` terminates.

To check that the recovery was successful, see the backup.log.

For more information, see *Diagnosis Files for Backup and Recovery*.

> **Note**
>
> If `recoverSys.py` returns an exit code '0', this is not confirmation that the recovery was successful.
>
> The recovery is not complete yet.
>
> If you use the parameter `--wait`, the script waits until the recovery has completed.
>
> If you do not use the `--wait` parameter, you need to manually check whether the recovery has completed by looking at the instance status or the logs.

Related Information

Options for Recovery with recoverSys.py [page 1055]
Diagnosis Files for Backup and Recovery [page 1070]
RECOVER DATABASE Statement (Backup and Recovery)
RECOVER DATA Statement (Backup and Recovery)
BACKUP CHECK ACCESS Statement (Backup and Recovery)
BACKUP LIST DATA Statement (Backup and Recovery)

12.2.7.3.1 Options for Recovery with recoverSys.py

The default behavior of the `recoverSys.py` tool can be overridden using the options described below.

Options for recoverSys.py

recoverSys.py Options	Description
`--help`	Get help for the `recoverSys.py` script.

recoverSys.py Options	Description
`--command="<SQL command>"`	Use this option to specify an SQL recovery command. **Sample Code** This statement performs a recovery to the database state of `'2023-10-22 15:00:00'`. ``` HDBSettings.sh recoverSys.py -- command="RECOVER DATABASE UNTIL TIMESTAMP '2023-10-22 15:00:00'" ``` **Note** If you specify a point in time in the future, the effect will be the same as recovering the database to the most recent state. **Sample Code** This statement performs a recovery to the database state of `'2018-10-22 15:00:00'` based on the data backup identified by `BACKUP ID '1380740407446'`, using the backup catalog located in `'/remote/backup/CHH/catalog'`. ``` HDBSettings.sh recoverSys.py --command="RECOVER DATABASE UNTIL TIMESTAMP '2018-10-22 15:00:00' USING CATALOG PATH ('/ remote/backup/CHH/catalog') USING BACKUP_ID 1380740407446 CHECK ACCESS USING FILE" ``` The statement checks the availability of the backup files before performing the recovery. **Sample Code** To perform a recovery on a remote host, pass the recovery command to a remote shell command. ``` ssh <sid>adm@<remoteHost> "HDBSettings.sh recoverSys.py -- command=\"RECOVER DATABASE UNTIL TIMESTAMP '2018-10-22 15:00:00'\"" ``` **Remember** The times specified are UTC times.

recoverSys.py Options	Description
`--wait`	Causes the script to wait until the recovery has completed (either successfully or unsuccessfully). Default: The script **does not wait** for the recovery to complete. The recovery is started and runs in the background. If the script is terminated manually, the database recovery will not stop. For more information, see *Starting and Stopping Distributed SAP HANA Systems Using sapcontrol*. **Note** If `recoverSys.py` is called automatically, you should use the option `--wait` to wait for the recovery to complete before you send further commands to the database.
`--password=<password>`	If authentication is necessary, you can supply a password for `<sid>adm`. If you do not specify the password, `recoverSys.py` prompts you to enter a password. **Note** If you use the `--password` option, the password can be displayed in the process list of the operating system.
`--timeout=<time>`	Specify a timeout for database shutdown and start. Default: 120s
`--licenseFile=<file name>`	Specify a license key file to append to the recovery command as a `SET LICENSE` clause. If you specify a command using the `--command` option, `SET LICENSE` is automatically appended to the command. **Note** An SAP HANA license key becomes invalid if the `<SID>` or landscape ID is changed. The recovered system is assigned a temporary license that is valid for 90 days. You can apply to SAP to have the license from the source system transferred to a new license key for the recovered system.
`--semaphoreOnly`	For use by SAP HANA cockpit only.

recoverSys.py Options	Description
`--masterOnly`	For use by SAP HANA cockpit only.
`--forceMaster <host>`	`recoverSys.py` attempts to use the current host as the master host for the recovery. If this host cannot be used as a master, `recoverSys.py` fails. To use a different host, use `--forceMaster` to specify the master host for the recovery. **Note** At most, three hosts can be used as the master host. The roles of the hosts are defined through the nameserver.ini file. For this reason, it is not possible to use any host as the master host.
`--feature`	For use by SAP HANA studio only.
`--silent`	Use this option to reduce diagnostics output.
`--cancel`	Use this option to cancel a recovery after it has started. **Note** Canceling a recovery makes the database state inconsistent. SAP HANA prevents an inconsistent database from being started. To be able to work with the database again after canceling a recovery, you would need to perform the recovery again. For more information, see *Cancel a Recovery*.

Related Information

RECOVER DATABASE Statement (Backup and Recovery)
RECOVER DATA Statement (Backup and Recovery)
BACKUP CHECK ACCESS Statement (Backup and Recovery)
BACKUP LIST DATA Statement (Backup and Recovery)
Starting and Stopping Distributed SAP HANA Systems Using SAPControl [page 1124]
Cancel a Recovery

12.2.7.4 Recovery Scenarios

Depending on the cause of the database failure, a different recovery strategy and procedure may be appropriate.

The following sections outline the recommended steps to recover the database in different recovery scenarios.

Related Information

Data Area is Unusable (Disaster Recovery) [page 1059]
Log Area is Unusable (Disaster Recovery) [page 1060]
Logical Error - Point-in-Time Recovery (Fault Recovery) [page 1061]
Recovery with System Replication [page 1062]

12.2.7.4.1 Data Area is Unusable (Disaster Recovery)

If the data area becomes unusable, you can recover an SAP HANA database.

If the data area is unusable, and all data changes since the last complete data backup are still available in the log backups and log area, you can still recover the data from committed transactions that was in the memory at the time of the failure. No committed data is lost.

> ⚠ Caution
>
> If you reinstall the SAP HANA software to recover the database, **do not first create a data backup**.
>
> The first data backup in a newly installed SAP HANA system creates a new backup catalog. As a result, the backup catalog from the old database is hidden and cannot be used for the recovery without manual intervention. However, in this scenario, the old backup catalog is still needed to recover the old database.

Once the database has been recovered successfully from a data backup or a data snapshot, the log entries from the log backups and the log area are replayed.

It is also possible to recover the database using an older data backup or data snapshot in combination with delta backups and log backups. The log backups needed for the recovery include those created **after** the data backup or data snapshot.

For more information, see *SAP Note 1821207 (Determining required recovery files)*.

> ⓘ Note
>
> In the recovery dialog, ensure that the paths to the data and log backup files are correct.

Used for Recovery

- Data backup
  Alternatively, data snapshot
- Delta backups
- Log backups
- Log area

Steps for Recovery

Recover the database to its most recent state.

For more information, see *Recovering an SAP HANA Database*.

Related Information

SAP Note 1821207
Recovering an SAP HANA Database
Recover SAP HANA From a Data Snapshot

12.2.7.4.2 Log Area is Unusable (Disaster Recovery)

If a log area becomes unusable, it is still possible to recover an SAP HANA database.

If the **log area** becomes unusable, it cannot be used for a recovery. It is only possible to recover the entries from the **log backups**. As a consequence, any changes that were made after the most recent log backup will be lost after a recovery. In addition, all the transactions that were open during the log backup will be rolled back.

It is still possible to recover the database to a point in time covered by the existing log backups.

> ⚠ Caution
>
> If you reinstall the SAP HANA software to recover the database, **do not first create a data backup**.
>
> The first data backup in a newly installed SAP HANA system creates a new backup catalog. As a result, the backup catalog from the old database is hidden and cannot be used for the recovery without manual intervention. However, in this scenario, the old backup catalog is still needed to recover the old database.

To prevent entries from the unusable log area from being replayed, in the recovery dialog, you must select the *Initialize log area* option. This option initializes the log area, and the old (unusable) content of the log area is lost.

Used for Recovery

- Data backup
  Alternatively, data snapshot
- Delta backups
- Log backups

Steps for Recovery

1. Recover the database to the most recent state.
   When the database has been successfully recovered from the data backup or data snapshot, the log entries from the log backups are replayed.
2. Select the *Initialize log area* option.

Related Information

Recovering an SAP HANA Database
Delta Backups [page 943]
Recover SAP HANA From a Data Snapshot

12.2.7.4.3 Logical Error - Point-in-Time Recovery (Fault Recovery)

If a logical database error occurs, you can recover an SAP HANA database to a point in time before the error occurred.

> ⚠ Caution
>
> All changes made after the point in time of the recovery will be lost in the recovered database.
>
> For this reason, a point-in-time recovery is not recommended for production systems.
>
> If you need to perform a point-in-time recovery of your production system, consider recovering the database to a different system and importing the missing data back into your production system. For example, if a specific table was lost, import that table from the recovered system to the new system.

Used for Recovery

- Data backup from before the point in time to recover to.

Alternatively, data snapshot

- Delta backups
- Log backups made after the data backup
- Log area

Steps for Recovery

Recover the database to a point in time before the logical error occurred.

You need to specify a point in time to which to recover the database.

> **Note**
>
> If you specify a point in time in the future, the effect will be the same as recovering the database to the most recent state.

Related Information

Recovering an SAP HANA Database
Recover SAP HANA From a Data Snapshot

12.2.7.4.4 Recovery with System Replication

If you are using a disaster-tolerant solution with system replication, some specific recovery scenarios apply.

Related Information

Point-In-Time Recovery of a Primary System [page 1063]
Recovery of a New Primary System After a Takeover [page 1064]
Points to Note: System Replication [page 936]

12.2.7.4.4.1 Point-In-Time Recovery of a Primary System

A primary system in a system replication scenario can be recovered to a specific point in time.

Used for Recovery

- Data backup
  Alternatively, data snapshot
- Delta backups

  > **Note**
  >
  > Delta backups must be based on the complete data backup of the current primary. After a takeover, if you wish to create delta backups, you must first have created a complete data backup of the current primary.

- Log backups
  The log backups that are associated with the data backup and cover the desired point-in-time (including log backups that were created **after** the desired point in time).

Steps for Recovery

1. Keep the secondary system online or unregister the secondary system.
   If the secondary system continues to run while the primary system is being recovered:

Primary System Recovered	Secondary System...
Tenant database:	is re-initialized through full data shipping immediately after the primary system is online again.
System database:	is unregistered.

2. Recover the primary system.
3. If the system database was recovered, or if the secondary system was unregistered (Step 1), re-register the secondary system to the primary system.
   For more information, see *Configure SAP HANA System Replication with hdbnsutil*.
4. Start the secondary system.

Related Information

Configure SAP HANA System Replication with hdbnsutil
Recovering an SAP HANA Database
Recovery with System Replication [page 1062]

SAP Note 3195590

12.2.7.4.4.2 Recovery of a New Primary System After a Takeover

With system replication, during a takeover, you switch your active system from the current primary system to the secondary system. After a takeover, it may at some stage be necessary to recover the active system (the former secondary system).

Used for Recovery

- Data backups
  The data backup can be created either from the original primary or the now active system. Alternatively, you can use a data snapshot.
- Log backups
  The log backups that belong to the data backup or the data snapshot. That is, if the data backup was made on the new primary system after takeover, only the log backups from the new primary system can be used.
- Log area of the new primary system
- Delta backups

  > ### Note
  >
  > Delta backups must be based on the complete data backup of the current primary. After a takeover, if you wish to create delta backups, you must first have created a complete data backup of the current primary.

> ### Caution
>
> If SAP HANA is recovered from backups that were created with different UIDs, some third-party backup tools may prevent the recovery from being started.
>
> For more information, contact your backup tool vendor or ensure that the same UID is used for all the backups used for a recovery.

Steps for Recovery

1. Ensure that the original primary system is stopped and is not writing complete data backups and log backups.
2. Ensure that the required data backup (or data snapshot), the delta backups, and the log backups can be accessed by the now active system.
3. Recover the now active system.
   For more information, see *Recovering an SAP HANA Database*.

When the database has been successfully recovered from the data backup or data snapshot, the log entries from the log backups are replayed.

> ⚠ Caution
>
> After a takeover, ensure that the original primary system does not continue to write log backups to the same location as the now active system.
>
> For more information, see *Recovery with System Replication*.

> ⓘ Note
>
> After a takeover, it is not necessary to create a new full data backup of the now active system. Backups of the former primary system can be used to recover the database.
>
> Delta backups must be based on the complete data backup of the current primary. After a takeover, if you wish to create delta backups, you must first have created a complete data backup of the current primary.

Related Information

Recovering an SAP HANA Database
Recovery with System Replication [page 1062]
Data Snapshots [page 945]
Delta Backups [page 943]

12.2.8 Copying a Database Using Backup and Recovery

A database copy is a quick way to set up a cloned database, for example, for training, testing, or development.

You can use backup and recovery to copy a system database or a tenant database within the same system or to a different system. You can create a copy of a database using a complete data backup or a data snapshot. Additionally, using delta backups (differential or incremental backups) and log backups allows you to copy the database to a specific point in time.

> ⓘ Note
>
> To create a database copy using differential or incremental backups, you must also use log backups. If log backups are not available, you can only create a database copy using a full data backup.

You can create a database copy with the following combinations of source database and target database:

Source Database	Target Database
System database	The system database of a different system

Source Database	Target Database
Tenant database	A different tenant database in the same system A tenant database in a different system **Note** A tenant database cannot be copied to a single-container system.
Single-container system	Tenant database **Note** An SAP HANA backup created with SAP HANA 1.0 SPS10 (single-container system) or newer can be used to recover a tenant database.

Database Copy and System Replication

If you have system replication configured, and require near-zero downtime, consider using system replication to copy a tenant database.

For more information, see *Copying and Moving Tenant Databases Between Systems* in the *SAP HANA Administration Guide*.

Related Information

Copying and Moving Tenant Databases [page 671]
Copying a Database Using Backup and Recovery
Prerequisites for Copying a Database Using Backup and Recovery

12.2.8.1 Database Copy: Scenarios

The following examples outline situations in which a database copy is created.

Related Information

Database Copy: Target Database has Fewer Hosts Than the Source Database [page 1067]
Database Copy: Target Database has More Hosts Than the Source Database [page 1068]

Database Copy: System Database [page 1069]

12.2.8.1.1 Database Copy: Target Database has Fewer Hosts Than the Source Database

In this example scenario, we create a database copy where the target database has fewer hosts than the source database.

Prerequisites

The same prerequisites apply as for an SAP HANA database recovery.

For more information, see *Prerequisites for Database Recovery* in *SAP HANA Administration with SAP HANA Cockpit (Backup and Recovery)*.

Context

In the following example, the source database and the target database are both tenant databases.

The source database has two hosts, each with one index server.

The target database has only one host.

File-based backups are used.

Procedure

1. Create a target database with one host.

   Follow the steps described in *Create a Tenant Database*.
2. Copy the database.

   Follow the steps described in *Copy a Database Using File-Based Data Backup or Data Snapshot Only*.

   All content of the data backup is recovered to the target host.

   The target database now holds the data of the two hosts from the source database, and there are two index servers on the single-host target database.
3. Remove the index server.

   For more information, see *Add or Remove Services in a Tenant Database*.

Results

The source database with two hosts has been copied to the target database with one host.

Related Information

Prerequisites for Database Recovery
Create a Tenant Database
Steps After Copying a Database
Add Services in a Tenant Database

12.2.8.1.2 Database Copy: Target Database has More Hosts Than the Source Database

In this example scenario, a third-party backup tool is used to copy a SAP HANA database. The target database has more hosts than the source database.

Prerequisites

The same prerequisites apply as for an SAP HANA database recovery.

For more information, see *Prerequisites for Copying a Database Using Backup and Recovery* in *SAP HANA Administration with SAP HANA Cockpit (Backup and Recovery)*.

Context

In the following example, the source database and the target database are both tenant databases.

The source database has **two** hosts, each with one index server.

The target database has **three** hosts.

You are using a third-party backup tool.

Procedure

1. Create a target database.

   Follow the steps described in *Create a Tenant Database*.

You can use CREATE DATABASE to create the target database. SAP HANA automatically creates an additional index server on a second host. There is no need to remove or add a host.

2. Copy the database.

   Follow the steps described in *Copy a Database*.

   All content of the backup is recovered to the two existing hosts.

   At this stage, the third host in the target system remains unused.

   For more information, see *Adding and Removing Hosts*.

3. Distribute all the data in the target database from the two hosts to the three hosts that are now available.

Results

A database with three hosts has been created, containing the data from the previous two-host database.

Related Information

Prerequisites for Copying a Database Using Backup and Recovery
Create a Tenant Database
Copy a Database
Steps After Copying a Database
Adding and Removing Hosts [page 1081]

12.2.8.1.3 Database Copy: System Database

In these example scenarios, a system database is copied to an SAP HANA system with different host roles.

Scenario 1

System 1	System 2
Host A: host_role = worker	Host C: host_role = worker, xs_worker
Host B: host_role = standby	Host D: host_role = standby, xs_standby

It is possible to copy (recover) the system database from a backup created in System 1 to System 2, because the host roles of the System 2 hosts contain same the roles as System 1.

The services from host A will be moved to host C; the services from host B will be moved to host D.

However, it will not be possible to recover the system database from a backup from System 2 to System 1, because the hosts in System 1 do not have the xs_worker and xs_standby roles.

Scenario 2

System 1	System 2
Host A: host_role = worker	Host C: host_role = worker
Host B: host_role = xs_worker	

It is possible to copy (recover) the system database from System 1 to System 2. This is a scale-down scenario. Host B on System 1 with only the xs_worker role role is removed from the target system.

It is possible to copy (recover) the system database from System 2 to System 1. This is a scale-up scenario. Services from host C will be moved to host A; host B will be used for XSA-related services.

Related Information

Prerequisites for Copying a Database Using Backup and Recovery
Copy a Database
Steps After Copying a Database

12.2.9 Diagnosis Files for Backup and Recovery

The `backup.log` and `backint.log` files record information about backups and recoveries. This information can be used to diagnose errors.

Each SAP HANA database writes its own `backup.log` and `backint.log` file.

If log rotation is enabled, sequentially named files can also exist. For example `backup.001.log` or `backint.001.log`

Related Information

backup.log [page 1071]
backint.log [page 1072]
Housekeeping for backup.log and backint.log [page 1072]
View Diagnostic Files in the SAP HANA Database Explorer
Configuring SAP HANA System Properties (INI Files)

SAP Note 2797078

12.2.9.1 backup.log

The `backup.log` file records information about data backups, log backups, and the backup catalog.

`backup.log` also records information about recovery operations.

> **Note**
>
> The SQL statement used for a recovery is recorded in `backup.log`. The point in time of a recovery is specified in the SQL statement as **UTC**.
>
> The time at which the recovery was started and completed is recorded in `backup.log` as **local server time**, not UTC.

> **Note**
>
> For a point-in-time recovery, the point in time specified in the SQL statement may be different from the point in time that was actually reached in the recovery. This is because the point in time that was actually reached in the recovery is that of the most recent global COMMIT to the database that was recovered.
>
> For example, a point in time of **`13:15`** was specified for a recovery. SAP HANA interprets this time as UTC. The point in time reached could be **`13:28:56+01:00`** (local server time), which would be **`12:28:56`** as UTC.
>
> The point in time recorded for a point-in-time recovery is the same regardless of whether the backups are from the file system or from a third-party backup tool.

Display the Content of backup.log

Using SAP HANA cockpit, you can display the content of `backup.log` as follows:

Choose *View trace and diagnostic files* > *<system database>* > *Database Diagnostic Files* > *<host>* > *other*.

Related Information

Automatic File Rotation for backup.log and backint.log [page 1073]
Deleting backup.log and backint.log Using SQL Statements [page 1075]

12.2.9.2 backint.log

`backint.log` contains information about the activities of the `Backint` agent.

The `Backint` agent is part of a third-party backup tool.

`backint.log` records all the parameters used to call the `Backint` agent, and the values returned. Each time the `Backint` agent is called, the command parameters and the return code are appended to `backint.log`.

`backint.log` includes the content of the following files:

- `Backint` input file
  This file is created when the `Backint` agent is started.
- `Backint` output file
  The `Backint` agent writes its output to this file.

Display the Content of backint.log

You can display the content of `backint.log` as follows using SAP HANA cockpit:

Choose ▶ *View trace and diagnostic files* ▶ `<system database>` ▶ *Database Diagnostic Files* ▶ `<host>` ▶ *other* ▶.

Related Information

Automatic File Rotation for backup.log and backint.log [page 1073]
Deleting backup.log and backint.log Using SQL Statements [page 1075]

12.2.9.3 Housekeeping for backup.log and backint.log

As more data is written to `backup.log` and `backint.log`, the files grow. Their increased size will not impact database performance, but may possibly occupy a large amount of space on disk.

To prevent the files from growing excessively, you can:

- Enable automatic file rotation for `backup.log` and `backint.log`
- Use SQL statements to delete `backup.log` and `backint.log`

> ⚠ Caution
>
> `backup.log` and `backint.log` are managed by SAP HANA. For this reason, you **must not manually delete** these files on operating system level.
>
> If it is necessary to manually delete log files, only delete them using SQL statements. For more information, see *Deleting backup.log and backint.log Using SQL Statements*.

Related Information

Automatic File Rotation for backup.log and backint.log [page 1073]
Deleting backup.log and backint.log Using SQL Statements [page 1075]

12.2.9.3.1 Automatic File Rotation for backup.log and backint.log

Automatic file rotation can be configured individually for each database. For each database, the configuration settings always apply to both `backup.log` and `backint.log`.

> **⚠ Restriction**
>
> Rotation is not supported by SAP HANA studio.
>
> For more information, see *SAP Note 2797078 (How to configure the size of the HANA Backup files backup.log and backint.log)*.

By default, automatic file rotation is disabled.

To enable rotation, configure the following parameters for each database in `global.ini`, section `backup`:

Parameter	Description
`max_trace_file_size`	Specify the maximum size of the `backup.log` and `backint.log` files. Default: **`-1`** (Rotation is disabled, and the files can grow indefinitely.) Specify the desired size as an integer in MB (for example, enter **`10`** for 10MB, **`15`** for 15MB). When you specify a file size, automatic file rotation is then enabled. To disable automatic file rotation, reset the parameter to its default value of **`-1`**. When automatic file rotation is enabled, a sequential 3-digit number from 000 to 999 is appended to the file name. For example: `backup.001.log`, `backint.010.log` When a log file reaches the specified maximum size, it is rotated. When a log file is rotated, a new empty file is created and assigned the next sequential number. If there is no restriction on the maximum number of log files, or if `max_trace_files` >= **`999`**, when `backup.999.log` reaches its maximum size, the file numbering starts from `backup.000.log` again. If, at this

Parameter	Description
	time, a log file named `backup.000.log` already exists, it is overwritten. **Note** Diagnosis messages are never split across multiple log files. Consequently, the specified maximum file size may be exceeded slightly.
`max_trace_files`	Specify the maximum number of rotation files to be retained for `backup.log` and `backint.log`. Default: **`10`** files (The minimum number of log files is **`2`**.) When the specified number of rotation files is exceeded, the oldest rotation file is automatically deleted. **Note** When automatic file rotation is enabled or disabled, existing log files are **not** automatically deleted.

Note

If automatic file rotation for backup diagnosis files is enabled, multiple backup.log or backint.log files can exist. The file names are numbered sequentially, in the same way as SAP HANA trace files. For example, `backup.001.log`.

For more information about how to configure parameters, see *Configuring System Properties in SAP HANA Cockpit* in *SAP HANA Administration with SAP HANA Cockpit*.

Related Information

backup.log [page 1071]
backint.log [page 1072]
SAP HANA Configuration Parameter Reference

12.2.9.3.2 Deleting backup.log and backint.log Using SQL Statements

If it is necessary to manually delete log files, only delete them by using SQL statements.

> ⚠ Caution
>
> `backup.log` and `backint.log` are managed by SAP HANA. For this reason, you **must not delete** these files on operating system level.

Use the following SQL statements:

- To delete `backup.log` files:
  `ALTER SYSTEM CLEAR TRACES ('BACKUP')`
- To delete `backint.log` files:
  `ALTER SYSTEM CLEAR TRACES ('BACKINT')`

With both SQL statements, you can optionally specify `WITH BACKUP` to compress a log file instead of deleting it.

> → Tip
>
> You can use these SQL statements regardless of whether automatic file rotation is enabled or disabled.

For more information, see *ALTER SYSTEM CLEAR TRACES Statement (System Management)* in the *SAP HANA SQL Reference Guide for SAP HANA Platform*.

Related Information

ALTER SYSTEM CLEAR TRACES Statement (System Management) (SAP HANA SQL Reference Guide for SAP HANA Platform)

Automatic File Rotation for backup.log and backint.log [page 1073]

12.2.10 Reference

Reference and legacy information about SAP HANA backup and recovery.

Related Information

Backup Alerts [page 1076]

12.2.10.1 Backup Alerts

SAP HANA raises alerts that warn you of errors related to backups.

Alert:	**Check availability of volumes for backup**
Alert ID:	34
Description:	This check warns you if a backup cannot be created because a volume or a service is unavailable. This alert can be triggered in combination with the alerts `NOT_ASSIGNED_VOLUMES` and `CHECK_INACTIVE_SERVICES`.
Alert Text:	`<no>` (`<service>`) is not available. A backup cannot be created.
User Action:	Find out why the volume or service is not available.
Default Interval:	1 hour

Alert:	**Check whether a data backup exists**
Alert ID:	35
Description:	Checks whether at least one data backup exists, and warns you if no successful data backup is available for the instance. You are warned before any actual data loss occurs.
Alert Text:	No data backup exists.
User Action:	To ensure that your database can be recovered, create a data backup as soon as possible.
Default Interval:	6 hours This check is also performed when the database is started.

Alert:	**Check last data backup**
Alert ID:	36
Description:	Checks whether the last data backup was successful, and warns you if the last data backup failed. If a scheduled backup fails, this check can help you prevent a situation from arising where no current backups are available.
Alert Text:	The last data backup was not successful.

Alert:	Check last data backup
User Action:	Find out why the last data backup was not successful, resolve the problem, and create a new data backup as soon as possible.
Default Interval:	1 hour

Alert:	Check the age of the last data backup
Alert ID:	37
Description:	Checks the age of the last successful backup. This includes complete data backups, data snapshots, and delta backups (differential data backups and incremental data backups). If the last successful backup is too old, the following alert levels are generated: • 20 days: High • 7 days: Medium • 5 days: Low
Alert Text:	The last data backup is `<days>` days old.
User Action:	To reduce your downtime in a recovery situation, create a data backup as soon as possible.
Default Interval:	24 hours

Alert:	Check last log backups
Alert ID:	38
Description:	Checks whether the last log backups were successful, and provides information about a failed log backup for a service or volume. As log backups are created automatically, this is the only way to notify users. This check should therefore be performed frequently and be accorded high priority. **Note** Log backups are created in separate backup destinations for each volume. For this reason, log backups need to be checked for each volume.
Alert Text:	The last log backup was not successful for volume `<no>` (`<service>` at `<host>`:`<port>`).

Alert:	Check last log backups
User Action:	Find out why a log backup was not successful and resolve the problem.
Default Interval:	15 minutes

Alert:	Runtime of the currently running log backups
Alert ID:	65
Description:	Determines whether the most recent log backup terminates in the specified time.
Alert Text:	A log backup with ID `<id>` has been running for longer than `<value>` seconds.
User Action:	Investigate why the log backup runs for too long, and resolve the issue.
Default Interval:	60 seconds

Alert:	Storage snapshot (data snapshot) is prepared
Alert ID:	66
Description:	Determines whether the period, during which the database is prepared for a data snapshot, exceeds a given threshold.
Alert Text:	The database was prepared for a data snapshot for longer than `<value>` seconds.
User Action:	Investigate why the data snapshot was not confirmed or abandoned, and resolve the issue.
Default Interval:	5 minutes

Alert:	Enable automatic log backup
Alert ID:	69
Description:	Determines whether automatic log backup is enabled.
Alert Text:	Automatic log backup is disabled.
User Action:	Enable automatic log backup.
Default Interval:	15 minutes

Related Information

Change the Backup Configuration Settings
Estimate the Space Needed in the File System for a Data Backup [page 1023]
Data Snapshots [page 945]
Monitoring Alerts

12.3 Scaling SAP HANA

There are two general approaches you can take to scale your SAP HANA system: scale up and scale out.

Scale up means increasing the size of one physical machine by increasing the amount of RAM available for processing.

Scale out means combining multiple independent computers into one system. The main reason for distributing a system across multiple hosts (that is, scaling out) is to overcome the hardware limitations of a single physical server. This allows an SAP HANA system to distribute the load between multiple servers. In a distributed system, each index server is usually assigned to its own host to achieve maximum performance. It is possible to assign different tables to different hosts (partitioning the database), as well as to split a single table between hosts (partitioning of tables).

Related Information

Aspects of Scalability [page 1080]
Multiple-Host System Concepts [page 1082]
Host Addition Concepts [page 1086]
Adding Hosts to an SAP HANA System [page 1089]
Removing Hosts from an SAP HANA System [page 1098]
Configuring Host Roles [page 1104]
Configuring the Network for Multiple Hosts [page 1115]
Mapping Host Names for Database Client Access [page 752]
Scaling SAP HANA Extended Application Services, Classic Model [page 1123]
Starting and Stopping Distributed SAP HANA Systems Using SAPControl [page 1124]

12.3.1 Aspects of Scalability

Before you decide how to scale your SAP HANA implementation, there are a number of aspects that need to be considered, such as scaling data, performance, applications, and hardware.

Scaling the Data

One technique you can use to deal with planned data growth is to purchase more physical RAM than is initially required to set the allocation limit according to your needs, and then to increase it over time to adapt to your data. Once you have reached the physical limits of a single server, you can scale out over multiple machines to create a distributed SAP HANA system. You can do this by distributing different schemas and tables to different servers (complete data and user separation). However, this is not always possible, for example, when a single fact table is larger than the server's RAM size.

The most important strategy for scaling your data is **data partitioning**. Partitioning supports the creation of very large tables (billions of rows) by breaking them into smaller chunks that can be placed on different machines. Partitioning is transparent for most SQL queries and other data manipulations.

For more information, see the section on managing tables.

Scaling Performance

SAP HANA's performance is derived from its efficient, parallelized approach. The more computation cores your SAP HANA server has, the better the overall system performance is.

Scaling performance requires a more detailed understanding of your workload and performance expectations. Using simulations and estimations of your typical query workloads, you can determine the expected load that a typical SAP HANA installation may comfortably manage. At the workload level, a rough prediction of scalability can be established by measuring the average CPU utilization while the workload is running. For example, an average CPU utilization of 45% may indicate that the system can be loaded 2X before showing a significant reduction in individual query response time.

For more information, see the sections on workload management and performance analysis.

Scaling the Application

Partitioning can be used to scale the application as it supports an increasing number of concurrent sessions and complex analytical queries by spreading the calculations across multiple hosts. Particular care must be taken in distributing the data so that the majority of queries match partitioning pruning rules. This accomplishes two goals: directing different users to different hosts (load balancing) and avoiding the network overhead related to frequent data joins across hosts.

Scaling Hardware

SAP HANA is offered in a number of ways – in the form of an on-premise appliance, delivered in a number of different configurations and "sizes" by certified hardware partners or by using the tailored data center integration model, and as part of a cloud-based service. This creates different system design options with respect to scale-up and scale-out variations. To maximize performance and throughput, SAP recommends that you scale up as far as possible (acquire the configuration with the highest processor and memory specification for the application workload), before scaling out (for deployments with even greater data volume requirements).

> **Note**
>
> The SAP HANA hardware partners have different building blocks for their scale-out implementations. Therefore, you should always consult with your hardware partner when planning your scale-out strategy.

Related Information

Table Partitioning [page 317]
Workload Management [page 415]
Managing and Monitoring SAP HANA Performance [page 148]

12.3.2 Adding and Removing Hosts

After installation, it is possible to add hosts to a single-host or multiple-host SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM).

Before you add or remove hosts, it is important to understand multiple-host system concepts, as well as the SAP HANA database lifecycle manager. Also, be aware that changing the number of hosts may affect how you back up and recover SAP HANA.

For more information about installing SAP HANA, see the *SAP HANA Server Installation and Update Guide*.

Related Information

SAP HANA Server Installation and Update Guide

12.3.2.1 Multiple-Host System Concepts

It is important to review multiple-host system concepts like host grouping and storage options before installing a multiple-host system.

Host Types

When configuring a multiple-host system, the additional hosts must be defined as **worker** hosts or **standby** hosts (worker is default). Worker machines process data; standby machines do not handle any processing and instead wait to take over processes in the case of worker machine failure.

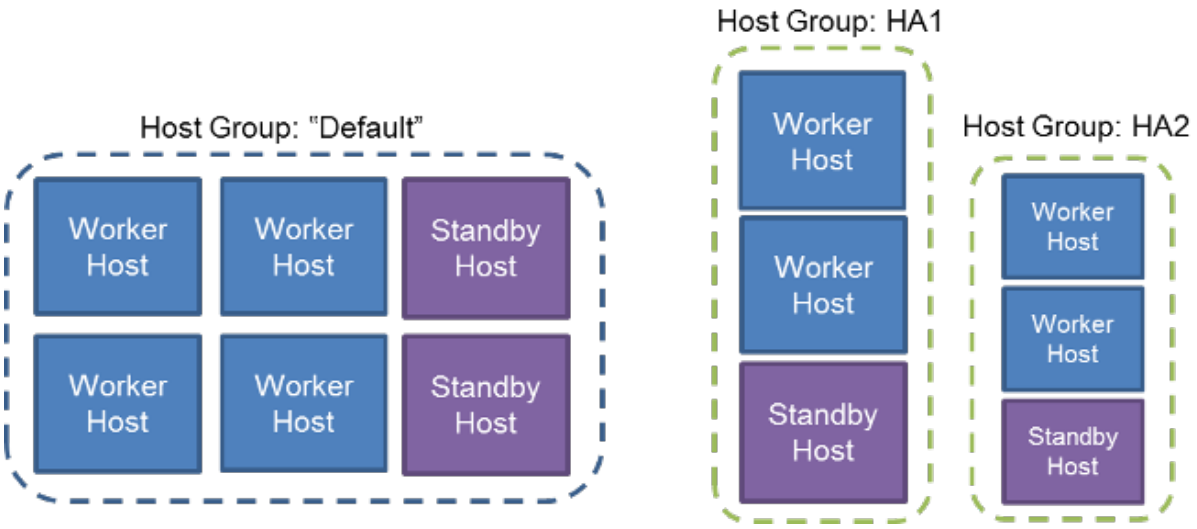
Auto-Failover for High Availability

As an in-memory database, SAP HANA is not only concerned with maintaining the reliability of its data in the event of failures, but also with resuming operations with most of that data loaded back in memory as quickly as possible. Host auto-failover is a local fault recovery solution that can be used as a supplemental or alternative measure to system replication. One (or more) standby hosts are added to a SAP HANA system, and configured to work in standby mode.

Before installing a multiple-host system, it is important to consider whether high availability is necessary and how hosts should be grouped to ensure preferred host auto-failover. For host auto-failover to be successful, if the active (worker) host fails, the standby host takes over its role by starting its database instance using the persisted data and log files of the failed host. The name server of one of the SAP HANA instances acts as the cluster manager that pings all hosts regularly. If a failing host is detected, the cluster manager ensures that the standby host takes over the role and the failing host is no longer allowed write access to the files (called fencing) so that they do not become corrupted. The crash of a single service does not trigger failover since services are normally restarted by `hdbdaemon`. For more information, see *Setting Up Host Auto-Failover* in the *SAP HANA Administration Guide*.

Host Grouping

Host grouping does not affect the load distribution among worker hosts - the load is distributed among all workers in an SAP HANA system. If there are multiple standby hosts in a system, host grouping should be considered, because host grouping decides the allocation of standby resources if a worker machine fails. If no host group is specified, all hosts belong to one host group called "default". The more standby hosts in one host group, the more failover security.



If the standby hosts are each in a different host group, the standby host in the same group as the failing worker host is preferred. Only if no standby host is available in the same host group, the system will try to fail over to a standby host, which is part of another host group. The advantage of this configuration is that in an SAP HANA system with mixed machine resources, similar sized machines can be grouped together. If a small worker host fails, and a small standby in the same group takes over, the processes are moved to a machine with similar resources, which allows processing to continue as usual with optimal resource allocation.

Worker Host Grouping for Warm Data

If you use SAP Business Warehouse to apply a temperature-based data strategy you can significantly optimize the usage of memory and hardware resources by reserving one node of the scaled-out HANA landscape exclusively for warm data. Due to information lifecycle management, multi-temperature strategies are often applied, whereby data is classified by access frequency as either hot, warm, or cold. Depending on this classification and data usage, this data is stored in different memory areas.

A multi-temperature memory strategy may be required for different reasons, for example:

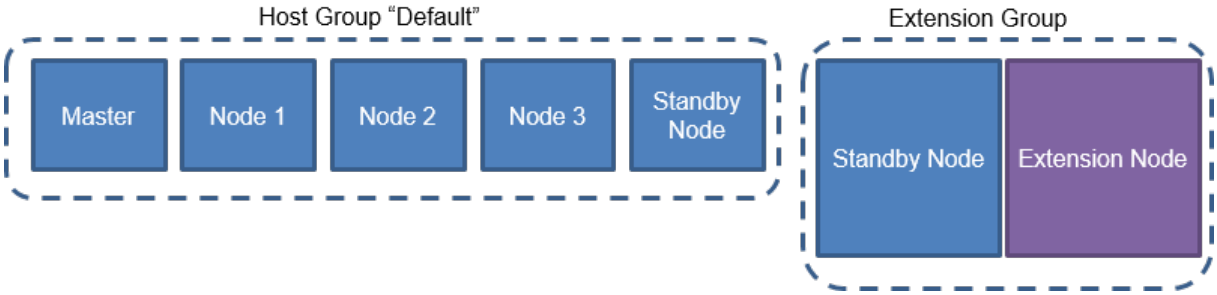
- Storage of historical data
- Clickstream logs for multiple years of Web data and detailed machine logs
- Guidelines for saving company data, such as the need to save all data for at least seven years due to legal reasons.

The standard SAP HANA sizing guidelines allow for a data footprint of 50% of the available RAM. This ensures that all data can be kept in RAM at all times and there is sufficient space for intermediate result sets. These sizing guidelines can be significantly relaxed on the extension group, since 'warm' data is accessed:

- less frequently
- with reduced performance SLAs
- with less CPU-intensive processes
- only partially at the same time.

To implement a multi-temperature memory strategy, you can assign hosts to worker groups. Hot and warm data is then distributed across hosts. To increase performance and memory usage, a slave node is assigned

to a separate *Extension Node*. Unlike the standard nodes (master and slave), the extension node is intended exclusively for data that is not accessed as frequently (warm) as other data (hot).



For more information, see *Data Temperature: Extension Node for Business Warehouse* in the *SAP HANA Administration Guide* and SAP Note 2453736.

Storage and File System Options

In single-host SAP HANA systems, it is possible to use local file systems residing on direct-attached internal or external storage devices, such as SCSI hard drives, SSDs, SAN storage, or NAS. However, in order to build a multiple-host system with failover capabilities this is not sufficient. Either the chosen file system type or the SAN Infrastructure along with a SAP HANA functionality capable of disc fencing must ensure the following:

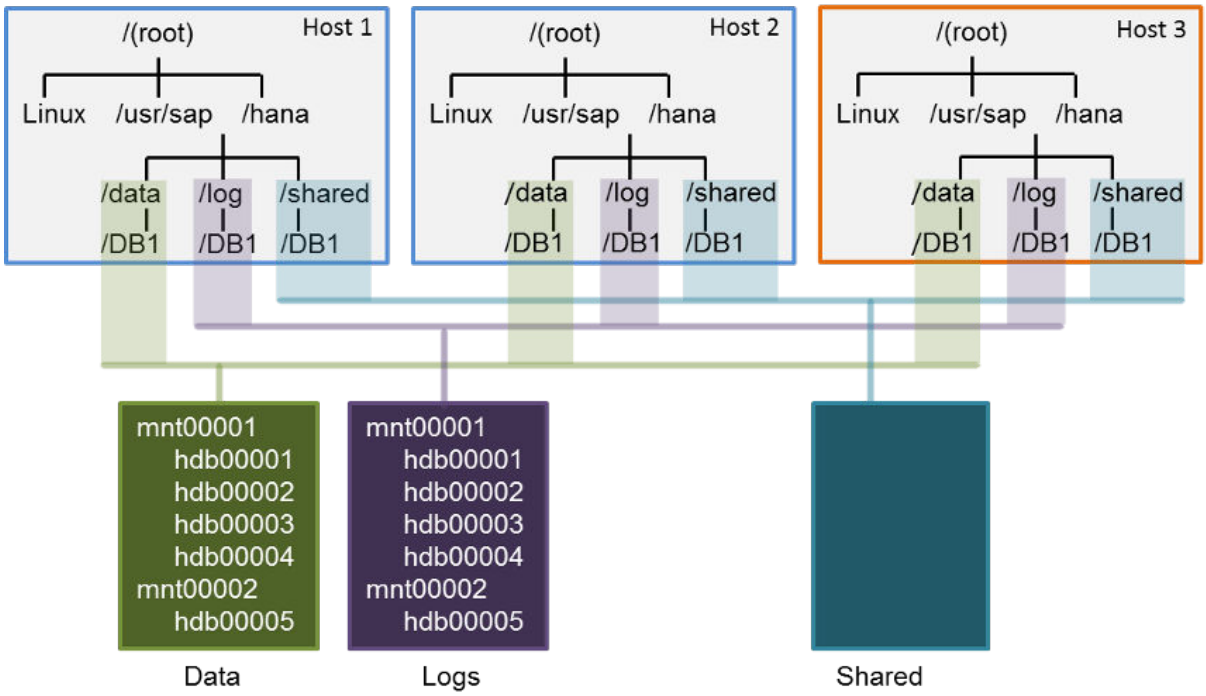
- The standby host has file access to data and log volumes of the failed host.
- The failed worker host no longer has access to write to files - called fencing.

There are two fundamentally different storage configurations which meet the two conditions above: **shared storage devices** or **separate storage devices with failover reassignment**. Do not confuse "shared storage" with the installation directory `/hana/shared` that must be shared across all hosts.

Shared File Systems

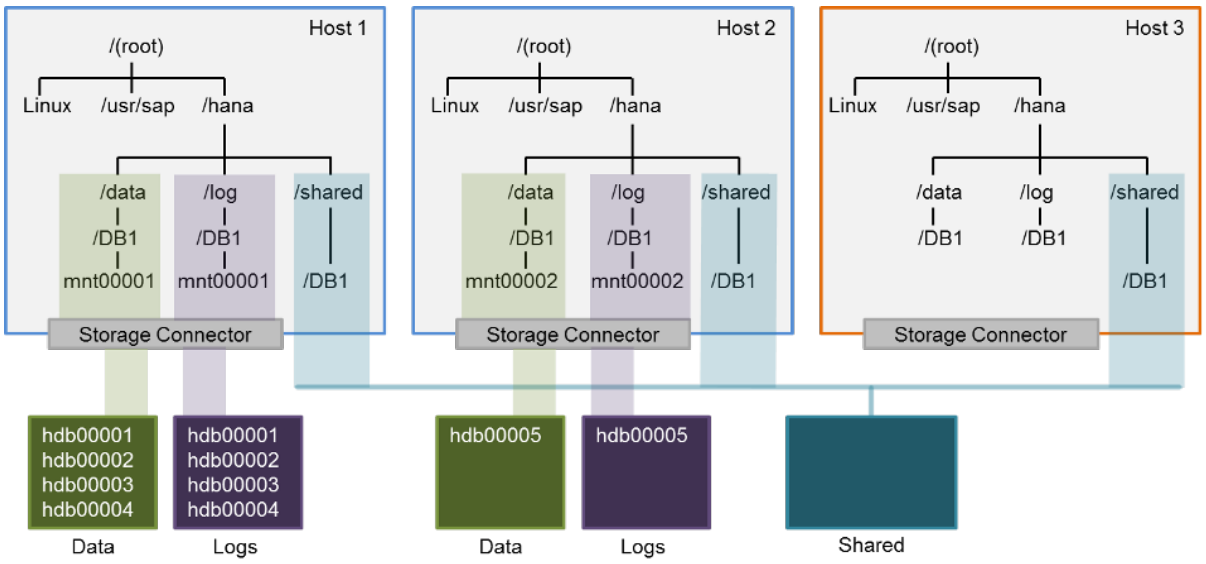
A shared storage subsystem, which is accessed using file systems such as NFS or IBM's GPFS, makes it easy to ensure that the standby host has access to all active host files in the system. In a shared storage solution, the externally attached storage subsystem devices are capable of providing dynamic mount points for hosts. Since shared storage subsystems vary in their handling of fencing, it is the responsibility of the hardware partner and their storage partners to develop a corruption-safe failover solution which is specific for the file system used to access that storage subsystem. An NFSv3 storage solution must be used in combination with the storage connector supplied by the hardware partner. NFSv4 and GPFS storage solutions can optionally be used with a storage connector.

A shared storage system could be configured as in the diagram below, however mounts may differ among hardware partners and their configurations. For more information, see the *SAP HANA Storage Whitepaper* available in SAP Note 1900823 in Related Information.



Non-shared Storage

It is also possible to assign every SAP HANA host a separate storage, which has nothing mounted except the shared area. A SAN storage must be used in combination with the SAP Fiber Channel Storage Connector, which SAP HANA offers storage technology vendors. During failover, SAP HANA uses the storage connector API to tell the storage device driver to remount the required data and logs volumes to the standby host and fence off the same volumes from the failed host.



In a non-shared environment, separate storage is used in combination with the storage connector API. For more information about the storage connector API, see the *SAP Fiber Channel Storage Connector Admin Guide* available in SAP Note 1900823 in Related Information.

Related Information

Recommended File System Layout
SAP Note 405827
Setting Up Host Auto-Failover [page 896]
SAP Note 1900823
Extension Node [page 243]
SAP Note 2453736
More Details – HANA Extension Nodes for BW-on-HANA
workergroup
ROUTE_TO Hint

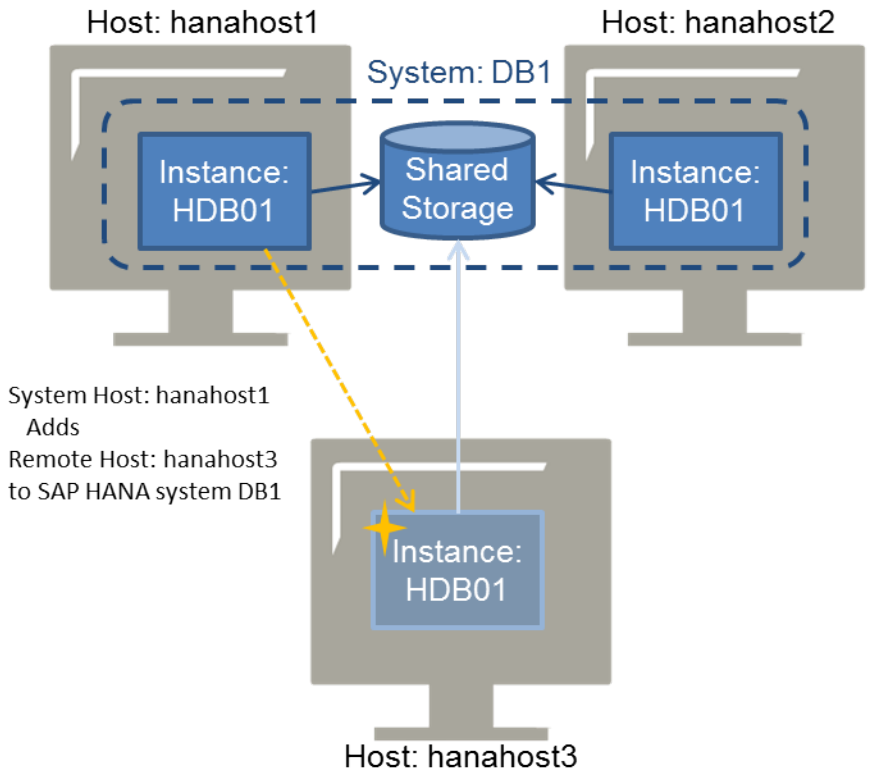
12.3.2.2 Host Addition Concepts

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM).

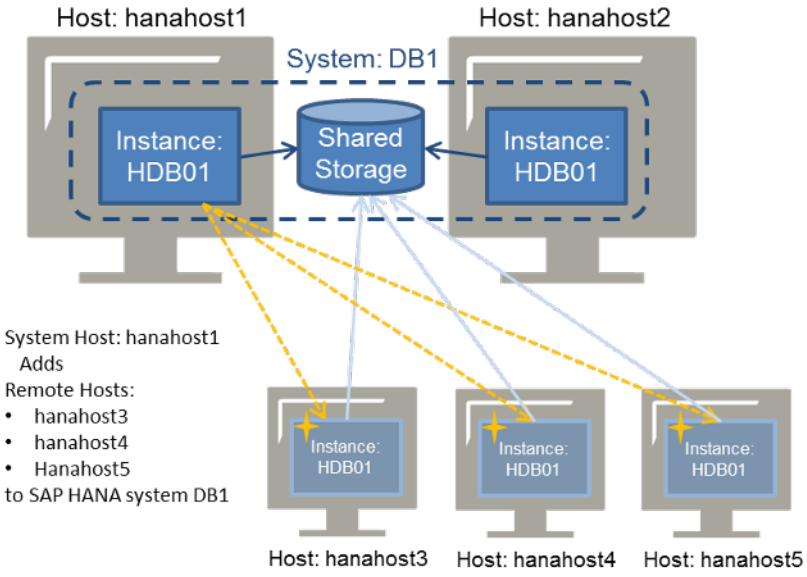
Using either the SAP HANA database lifecycle manager graphical user or command-line interface, one or multiple hosts can be added to an SAP HANA system in a variety of ways. The configuration options change depending on how the host is added.

Adding Hosts from an Integrated Host

The first consideration is whether the host you are logged on to is integrated in the system. If you are logged on to a configured system host, then you are on an integrated host and adding a non-integrated host to the system. In the diagram below, the hosts in the dotted line (hanahost1 and hanahost2) are integrated hosts because they both belong to the SAP HANA system DB1. Consider being logged on to hanahost1, and adding non-integrated host, hanahost3, to the SAP HANA system. The SAP HANA database lifecycle manager is started on the integrated host, hanahost1, and the addhost configuration task is carried out. The host information for hanahost3 is entered, and hanahost3 is configured as either a worker host or standby host. As soon as the addhost configuration task is finished, hanahost3 has access to the shared storage of the DB1 system.

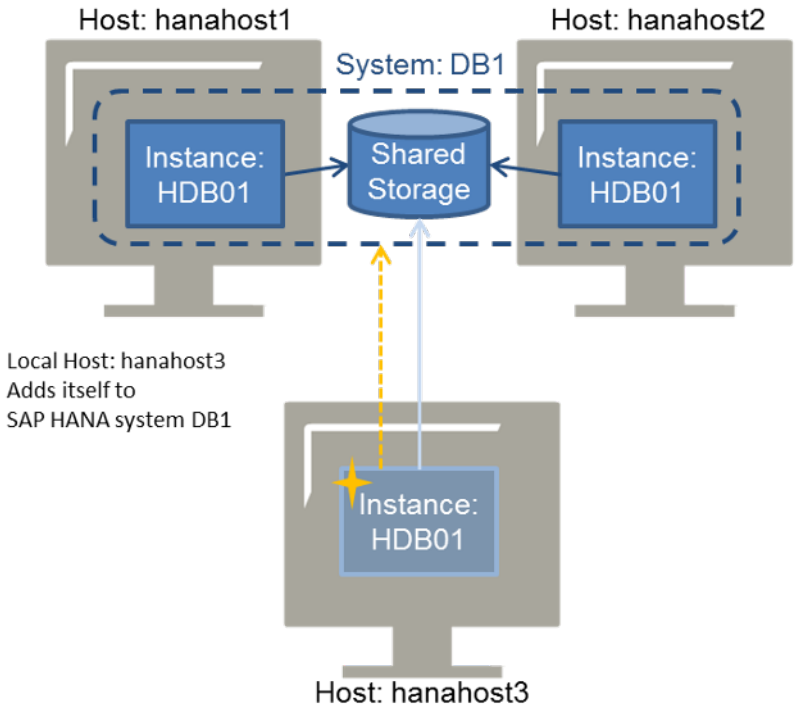


It is also possible to add multiple non-integrated hosts to the same system at one time. In the diagram below, three remote hosts (hanahost3, hanahost4, hanahost5) are added to the SAP HANA system (DB1) from a system host (hanahost1).

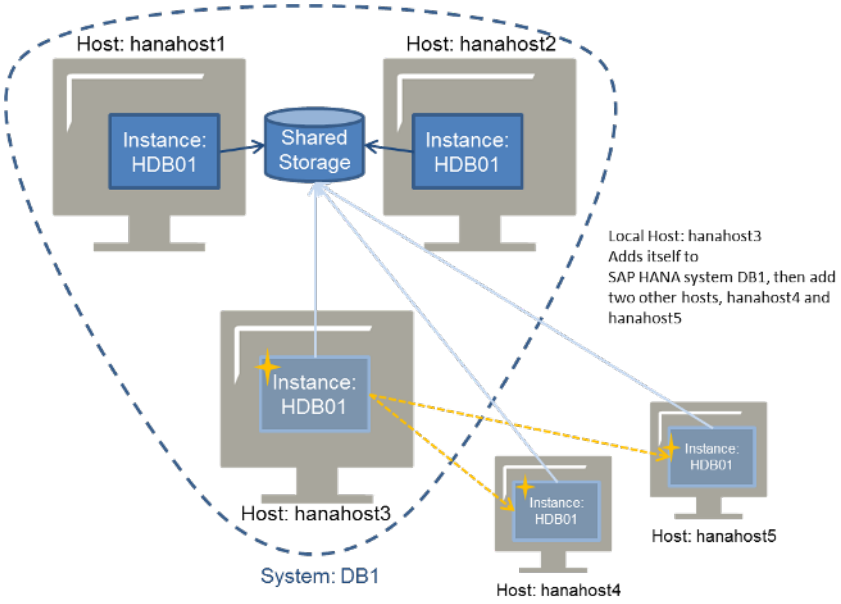


Adding Hosts from a Non-Integrated Host

Alternatively, a non-integrated host can add itself to an SAP HANA system. This is referred to as adding a host from a non-integrated host, because you are logged on to a host which you want to add to the system.



To add multiple hosts to an SAP HANA system from a non-integrated host, first the non-integrated host must be added (and, therefore, become integrated), and then it can add more hosts. The SAP HANA database lifecycle manager interface is designed so that the non-integrated host and the additional hosts can be added in the same procedure. In the diagram below, the non-integrated host has already been newly added to the system (become integrated), and is now adding the other hosts.



Related Information

Add Hosts Using the Command-Line Interface [page 1092]

12.3.2.3 Adding Hosts to an SAP HANA System

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program or the SAP HANA database lifecycle manager (HDBLCM) Web user interface.

If you want to configure a new multiple-host (distributed) system during installation, see the multiple-host system installation information in the *SAP HANA Server Installation and Update Guide*.

Before adding a host to an SAP HANA system, you need to consider the following:

- If you are adding hosts from a host that is already integrated in the SAP HANA system
- If the system is a single-host or multiple-host system
- How many hosts you want to add to the system at one time

For more information about how these conditions affect the addition of hosts to an SAP HANA system see the host addition concepts in Related Information.

If you are adding a host to a single-host system, the listen interface is automatically configured to global during the host addition. After the host is added to the system, the internal network address can be defined and the inter-service communication can be reconfigured to a different setting, if required. For more information about configuring inter-service communication, see Related Information.

Related Information

Multiple-Host System Concepts [page 1082]
Host Addition Concepts [page 1086]
Add Hosts Using the Graphical User Interface [page 1090]
Add Hosts Using the Command-Line Interface [page 1092]
Add Hosts Using the Web User Interface [page 1095]

12.3.2.3.1 Add Hosts Using the Graphical User Interface

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the graphical user interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

   ```
   ./hdblcmgui
   ```

   The SAP HANA database lifecycle manager graphical user interface appears.

   > ⓘ Note
   >
   > To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.
3. Select *Add Hosts to SAP HANA System* from the activity options. Then select *Next*.
4. Select *Add Host...* to define the required parameters. Then select *Next*.

Field Name	Description
Host Name	Specifies the host name of the machine.

Field Name	Description
Role	Specifies the purpose of the SAP HANA host. Although multiple host roles may be assigned, check the corresponding documentation for the SAP HANA option for what configurations are supported in production environments. • Database Worker (worker) - A worker host (default) is used for database processing. • Database Standby (standby) - A standby host is idle and available for failover in a high-availability environment. • Database Elastic Compute Server (compute) - Database elastic compute server • Dynamic Tiering Worker (extended_storage_worker) - Worker host for SAP HANA dynamic tiering • Dynamic Tiering Standby (extended_storage_standby) - Standby host for SAP HANA dynamic tiering • Accelerator for SAP ASE Worker (ets_worker) - Worker host for SAP HANA accelerator for SAP ASE • Accelerator for SAP ASE Standby (ets_standby) - Standby host for SAP HANA accelerator for SAP ASE • Streaming Analytics (streaming) - Host for SAP HANA streaming analytics • XS advanced runtime worker (xs_worker) - Host for SAP HANA XS advanced runtime • XS advanced runtime standby (xs_standby) - Standby host for SAP HANA XS advanced runtime
Worker Group	Specifies the worker group of the host. If undefined, the worker group is named "default". If you are using extension node for Business Warehouse, you must name the worker group "worker_dt".
High-Availability Group	Specifies the host group ID for failover scenarios. If undefined, the host group is named "default".
Storage Partition	Specifies the storage partition number, which is a logical role number assigned to non-shared storage devices in a storage connector API. Standby hosts do not have a storage partition.

5. Define additional system properties.

Field Name	Description
Inter-Service Communication	Specifies the listen interface for the internal network communication. `global` - Binds the processes to all interfaces. This option does not require an internal network address entry. `internal` - Binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry.
Internal Network Address	Specifies the internal subset address in CIDR notation. If you define a value other than `local`, the local interfaces will always be open.
Certificate Host Name	Specifies the hostname used for generation of self-signed SSL certificates for the SAP Host Agent.

6. Review the summary, and select *Add Hosts* to finalize the configuration.

Results

You have added one or more new hosts to an SAP HANA system. The SAP HANA system you have configured is a multiple-host system.

The new hosts have been added to the SAP HANA landscape information. The new hosts have been added to the landscape information of the system database.

12.3.2.3.2 Add Hosts Using the Command-Line Interface

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --addhosts=<host>[,<host2>]
   ```

where the syntax for the `addhosts` call option is as follows:

```
<host name>:role=<role name>:group=<group ID>:storage_partition=<partition number>
```

Field Name	Description
`<host name>`	Specifies the host name of the machine.
`role`	Specifies the purpose of the SAP HANA host. Although multiple host roles may be assigned, check the corresponding documentation for the SAP HANA option for what configurations are supported in production environments. • `worker` - A worker host (default) is used for database processing. • `standby` - A standby host is idle and available for failover in a high-availability environment. • `compute` - Database elastic compute server • `extended_storage_worker` - Worker host for SAP HANA dynamic tiering • `extended_storage_standby` - Standby host for SAP HANA dynamic tiering • `ets_worker` - Worker host for SAP HANA accelerator for SAP ASE • `ets_standby` - Standby host for SAP HANA accelerator for SAP ASE • `streaming` - Host for SAP HANA streaming analytics • `xs_worker` - Host for SAP HANA XS advanced runtime • `xs_standby` - Standby host for SAP HANA XS advanced runtime
`workergroup`	Specifies the worker group of the host. If undefined, the worker group is named "default". If you are using extension node for Business Warehouse, you must name the worker group "worker_dt".
`group`	Specifies the host group ID for failover scenarios. If undefined, the host group is named "default".
`storage_partition`	Specifies the storage partition number, which is a logical role number assigned to non-shared storage devices in a storage connector API. Standby hosts do not have a storage partition.

The required parameters depend on the type of host addition you are performing: host addition from an integrated host to a multiple-host system, host addition from an integrated host to a single-host system, or host addition from a non-integrated host. For more information about host addition types, see Related Information.

> **Note**
>
> When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the hdblcm directory of the SAP HANA system, execute the following command:
>
> ```
> ./hdblcm --action=add_hosts --help
> ```

3. Select the index for the *add_hosts* action.
4. Define additional system properties.

Field Name	Description
Inter-Service Communication	Specifies the listen interface for the internal network communication. `global` - Binds the processes to all interfaces. This option does not require an internal network address entry. `internal` - Binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry.
Internal Network Address	Specifies the internal subset address in CIDR notation. If you define a value other than `local`, the local interfaces will always be open.
Certificate Host Name	Specifies the hostname used for generation of self-signed SSL certificates for the SAP Host Agent.

5. Review the summary, and select *y* to finalize the configuration.

Results

You have added one or more new hosts to an SAP HANA system. The SAP HANA system you have configured is a multiple-host system.

The new hosts have been added to the SAP HANA landscape information. The new hosts have been added to the landscape information of the system database.

This configuration task can also be performed in batch mode and using a configuration file. For more information about the available configuration methods, see *Using the SAP HANA Platform LCM Tools*.

Example

The following example adds two hosts, *Host1* and *Host2* to a single-host SAP HANA system. The role of the two hosts is *worker*, by default. No SSH keys are installed. A trusted connection between the hosts is configured and therefore, root user password is not required. The listen interface of the SAP HANA system is changed to *global*.

```
./hdblcm --action=add_hosts --addhosts=host1,host2  --root_user=lmroot --
listen_interface=global
```

Related Information

Host Addition Concepts [page 1086]
Configure SAP HANA Inter-Service Communication Using the Command-Line Interface [page 1119]
Using the SAP HANA Platform LCM Tools [page 585]
addhosts

internal_network
listen_interface
nostart [page 75]
remote_execution
timeouts

12.3.2.3.3 Add Hosts Using the Web User Interface

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) Web user interface.

Prerequisites

- On the host, which is to be added, SAP Host Agent is installed with SSL configured. The SAP Host Agent will create the`<sapsys>` group, if it does not exist prior to installation. Make sure that the group ID of the `<sapsys>` group is the same on all hosts.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<SID>`adm) user may exist on the additional host. Make sure that you have the password of the existing `<SID>`adm user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) will not modify the properties of any existing user or group.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release

    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

> **Note**
>
> For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.

- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://`<hostname>`:1129/lmsl/HDBLCM/`<SID>`/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

2. Select the *Add Hosts* tile.
3. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
import_xs_content	Imports SAP HANA XS advanced runtime content.
Install or Update SAP Host Agent	Installs or updates SAP Host Agent.
Do Not Start Added Hosts	Does not start hosts after addition.
Do Not Modify 'sudoers' File	Prevents the file `/etc/sudoers.d/sap_hdb_sudoers` from being modified.
Timeouts	Sets customized timeouts (`start_instance`, `start_service`)

4. Provide the necessary credentials, then select *Add Host*.
5. Define the required host parameters. Then select *OK*.

Field Name	Description
Host Name	Specifies the host name of the machine.
Role	Specifies the purpose of the SAP HANA host. Although multiple host roles may be assigned, check the corresponding documentation for the SAP HANA option for what configurations are supported in production environments. • Database Worker (worker) - A worker host (default) is used for database processing. • Database Standby (standby) - A standby host is idle and available for failover in a high-availability environment. • Database Elastic Compute Server (compute) - Database elastic compute server • Dynamic Tiering Worker (extended_storage_worker) - Worker host for SAP HANA dynamic tiering • Dynamic Tiering Standby (extended_storage_standby) - Standby host for SAP HANA dynamic tiering • Accelerator for SAP ASE Worker (ets_worker) - Worker host for SAP HANA accelerator for SAP ASE • Accelerator for SAP ASE Standby (ets_standby) - Standby host for SAP HANA accelerator for SAP ASE • Streaming Analytics (streaming) - Host for SAP HANA streaming analytics • XS advanced runtime worker (xs_worker) - Host for SAP HANA XS advanced runtime • XS advanced runtime standby (xs_standby) - Standby host for SAP HANA XS advanced runtime
Worker Group	Specifies the worker group of the host. If undefined, the worker group is named "default". If you are using extension node for Business Warehouse, you must name the worker group "worker_dt".
High-Availability Group	Specifies the host group ID for failover scenarios. If undefined, the host group is named "default".
Storage Partition	Specifies the storage partition number, which is a logical role number assigned to non-shared storage devices in a storage connector API. Standby hosts do not have a storage partition.

6. Define additional system properties.

Field Name	Description
Inter-Service Communication	Specifies the listen interface for the internal network communication. `global` - Binds the processes to all interfaces. This option does not require an internal network address entry. `internal` - Binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry.

Field Name	Description
Internal Network Address	Specifies the internal subset address in CIDR notation. If you define a value other than `local`, the local interfaces will always be open.

7. Review the summary, and select *Run* to finalize the configuration.

Results

You have added one or more new hosts to an SAP HANA system. The SAP HANA system you have configured is a multiple-host system.

The new hosts have been added to the SAP HANA landscape information. The new hosts have been added to the landscape information of the system database.

Related Information

SAPUI5 Developer Guide

12.3.2.4 Removing Hosts from an SAP HANA System

A number of options are available to remove hosts from an SAP HANA system.

You can remove hosts from an SAP HANA system using any of the following methods:

- SAP HANA database lifecycle manager (HDBLCM) (graphical user interface or command line)
- SAP HANA database lifecycle manager (HDBLCM) Web user interface.

Refer also to SAP Note 2272550 – *Remove Host from SAP HANA System*, which includes details of the steps required to remove a host using the SQL procedure UPDATE_LANDSCAPE_CONFIGURATION().

Related Information

Remove Hosts Using the Graphical User Interface [page 1099]
Remove Hosts Using the Command-Line Interface [page 1100]
Remove Hosts Using the Web User Interface [page 1102]
Add Services in a Tenant Database
SAP Note 2272550

12.3.2.4.1 Remove Hosts Using the Graphical User Interface

You can remove hosts from an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) in the graphical user interface.

Prerequisites

- You are logged in as root user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- If you want to remove a host that runs the coordinator name server, another host that will take over the role of the coordinator name server must be up and running.
- You have removed all tenant-specific services from the host.
- You are logged on as root user or as the system administrator user `<sid>adm`.

> ⚠ Caution
>
> Removing a host breaks the backup history of the database. To ensure that the database is fully recoverable, perform a full backup (data backup or storage snapshot) immediately after adding a service.

Procedure

1. Remove tenant-specific services. For more information, see *Remove a Service from a Tenant Database* in the *SAP HANA Tenant Databases Operations Guide*.
2. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
3. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

   ```
   ./hdblcmgui
   ```

   The SAP HANA database lifecycle manager graphical user interface appears.

   > ⓘ Note
   >
   > To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.
4. Select *Remove Hosts from the SAP HANA System* from the activity options. Then select *Next*.
5. Select the host you would like to remove from the system. Then select *Next*.

   You also have a choice to enable the following:

Field Name	Description
Keep System Administrator User	Keeps the system administrator user (`<sid>adm`) from the source system to be used in the target system.
Keep Home Directory of System Administrator	Prevents the home directory of the source system administrator user (`<sid>adm`) from being removed.

6. Enter the required credentials. Then select *Next*.
7. Review the summary, and select *Remove Hosts* to finalize the configuration.

Results

You have removed one or more new hosts from an SAP HANA system. This configuration task can also be performed using a configuration file. For more information about the available configuration methods, see *Using the SAP HANA Platform LCM Tools*.

Related Information

Using the SAP HANA Platform LCM Tools [page 585]
Host Addition Concepts [page 1086]
Add Services in a Tenant Database
SAP Note 2272550

12.3.2.4.2 Remove Hosts Using the Command-Line Interface

You can remove hosts from an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) in the command-line interface.

Prerequisites

- You are logged in as root user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- If you want to remove a host that runs the coordinator name server, another host that will take over the role of the coordinator name server must be up and running.
- You have removed all tenant-specific services from the host.
- You are logged on as root user or as the system administrator user `<sid>adm`.

> ⚠ Caution
>
> Removing a host breaks the backup history of the database. To ensure that the database is fully recoverable, perform a full backup (data backup or storage snapshot) immediately after adding a service.

Procedure

1. Remove tenant-specific services. For more information, see *Remove a Service from a Tenant Database* in the *SAP HANA Tenant Databases Operations Guide*.
2. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
3. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm
   ```
4. Select the index for the *remove_hosts* action.
5. Select the hosts to be removed as a comma-separated list of indexes, and specify the following system properties:

Field Name	Description
Keep System Administrator User	Keeps the system administrator user (`<sid>adm`) from the source system to be used in the target system.
Keep Home Directory of System Administrator	Prevents the home directory of the source system administrator user (`<sid>adm`) from being removed.

6. Review the summary, and select *y* to finalize the configuration.

Results

You have removed one or more new hosts from an SAP HANA system.

This configuration task can also be performed in batch mode and using a configuration file. For more information about the available configuration methods, see *Using the SAP HANA Platform LCM Tools* and SAP Note 2272550 *Remove Host from SAP HANA System*.

Related Information

Using the SAP HANA Platform LCM Tools [page 585]
Host Addition Concepts [page 1086]

Add Services in a Tenant Database

SAP Note 2272550

12.3.2.4.3 Remove Hosts Using the Web User Interface

You can remove hosts from an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) Web user interface.

Prerequisites

- You are logged in as root user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- If you want to remove a host that runs the coordinator name server, another host that will take over the role of the coordinator name server must be up and running.
- You have removed all tenant-specific services from the host.
- You are logged on as root user or as the system administrator user `<sid>adm`.

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

  > **Note**
  >
  > For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.
- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

> ⚠ Caution
>
> Removing a host breaks the backup history of the database. To ensure that the database is fully recoverable, perform a full backup (data backup or storage snapshot) immediately after adding a service.

Procedure

1. Remove tenant-specific services. For more information, see *Remove a Service from a Tenant Database* in the *SAP HANA Tenant Databases Operations Guide*.
2. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://`<hostname>`:1129/lmsl/HDBLCM/`<SID>`/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

3. Select the *Remove Hosts* tile.
4. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
Do Not Remove XS Advanced OS Users	Prevents the XS advanced runtime OS Users from being removed.
Do Not Modify 'sudoers' File	Prevents the file `/etc/sudoers.d/sap_hdb_sudoers` from being modified.
Timeouts	Sets customized timeouts (`start_instance`, `start_service`, `stop_instance`, `stop_service`).

5. Select the host you would like to remove from the system. Then select *Next*.

   You also have a choice to enable the following:

Field Name	Description
Keep System Administrator User	Keeps the system administrator user (`<sid>adm`) from the source system to be used in the target system.
Keep Home Directory of System Administrator	Prevents the home directory of the source system administrator user (`<sid>adm`) from being removed.

6. Enter the relevant credentials. Then select *Next*.
7. Review the summary, and select *Run* to finalize the configuration.

Related Information

Using the SAP HANA Platform LCM Tools [page 585]
Host Addition Concepts [page 1086]
Add Services in a Tenant Database

12.3.3 Configuring Host Roles

After installation, it is possible to add and remove host roles in a single-host or multiple-host SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM).

Before you add or remove a host role, it is important to understand multiple-host system concepts, and also the SAP HANA database lifecycle manager host addition concepts. Also, be aware that changing the number of hosts may affect how you back up and recover SAP HANA.

Only component-specific host roles can be added or removed. The roles `worker` and `standby` cannot be added or removed.

An SAP HANA system can also be configured with multiple host roles on single hosts during installation using the SAP HANA database lifecycle manager. For more information about installing an SAP HANA multiple-host system, see the *SAP HANA Server Installation and Update Guide*.

Related Information

Adding Host Roles [page 1105]
Removing Host Roles [page 1110]

12.3.3.1 Adding Host Roles

You can add host roles to hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program.

Related Information

Add Host Roles Using the Graphical User Interface [page 1105]
Add Host Roles Using the Command-Line Interface [page 1107]
Add Host Roles Using the Web User Interface [page 1108]

12.3.3.1.1 Add Host Roles Using the Graphical User Interface

You can add host roles to hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the graphical user interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.

2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

```
./hdblcmgui
```

The SAP HANA database lifecycle manager graphical user interface appears.

> **Note**
>
> To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.

3. Select *Add Host Roles* from the activity options. Then select *Next*.
4. Select *Assign Roles...* to assign additional host roles to each host. Then select *Next*.

Field Name	Description
Role	Specifies the purpose of the SAP HANA host. Although multiple host roles may be assigned, check the corresponding documentation for the SAP HANA option for what configurations are supported in production environments. • Database Worker (worker) - A worker host (default) is used for database processing. • Database Standby (standby) - A standby host is idle and available for failover in a high-availability environment. • Database Elastic Compute Server (compute) - Database elastic compute server • Dynamic Tiering Worker (extended_storage_worker) - Worker host for SAP HANA dynamic tiering • Dynamic Tiering Standby (extended_storage_standby) - Standby host for SAP HANA dynamic tiering • Accelerator for SAP ASE Worker (ets_worker) - Worker host for SAP HANA accelerator for SAP ASE • Accelerator for SAP ASE Standby (ets_standby) - Standby host for SAP HANA accelerator for SAP ASE • Streaming Analytics (streaming) - Host for SAP HANA streaming analytics • XS advanced runtime worker (xs_worker) - Host for SAP HANA XS advanced runtime • XS advanced runtime standby (xs_standby) - Standby host for SAP HANA XS advanced runtime

5. Review the summary, and select *Run* to finalize the configuration.

12.3.3.1.2 Add Host Roles Using the Command-Line Interface

You can add host roles to hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --action=add_host_roles
   ```
3. Select the hosts to which you would like to assign additional roles.
4. Select the additional host roles that you want to assign for each host.

Field Name	Description
`role`	Specifies the purpose of the SAP HANA host. Although multiple host roles may be assigned, check the corresponding documentation for the SAP HANA option for what configurations are supported in production environments. • `worker` - A worker host (default) is used for database processing. • `standby` - A standby host is idle and available for failover in a high-availability environment. • `compute` - Database elastic compute server • `extended_storage_worker` - Worker host for SAP HANA dynamic tiering • `extended_storage_standby` - Standby host for SAP HANA dynamic tiering • `ets_worker` - Worker host for SAP HANA accelerator for SAP ASE • `ets_standby` - Standby host for SAP HANA accelerator for SAP ASE • `streaming` - Host for SAP HANA streaming analytics • `xs_worker` - Host for SAP HANA XS advanced runtime • `xs_standby` - Standby host for SAP HANA XS advanced runtime

5. Enter the required credentials.
6. Review the summary, and select *y* to finalize the configuration.

12.3.3.1.3 Add Host Roles Using the Web User Interface

You can add host roles to hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the Web user interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher

If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.

  - Microsoft Edge
  - Mozilla Firefox - Latest version and Extended Support Release
  - Google Chrome - Latest version
- SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
- Mac OS - Safari 5.1 or higher

> **Note**
>
> For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.

- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://`<hostname>`:1129/lmsl/HDBLCM/`<SID>`/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

2. Select the *Add Host Roles* tile.
3. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
Do Not Start Hosts After Addition of Roles	Does not start hosts after addition of roles.
Do Not Modify 'sudoers' File	Prevents the file `/etc/sudoers.d/sap_hdb_sudoers` from being modified.
Timeouts	Sets customized timeouts (`start_instance`, `start_service`, `stop_instance`, `stop_service`).

4. Select the hosts to which you would like to assign additional roles.
5. Select the additional host roles that you want to assign for each host.

Field Name	Description
`role`	Specifies the purpose of the SAP HANA host. Although multiple host roles may be assigned, check the corresponding documentation for the SAP HANA option for what configurations are supported in production environments. • `worker` - A worker host (default) is used for database processing. • `standby` - A standby host is idle and available for failover in a high-availability environment. • `compute` - Database elastic compute server • `extended_storage_worker` - Worker host for SAP HANA dynamic tiering • `extended_storage_standby` - Standby host for SAP HANA dynamic tiering • `ets_worker` - Worker host for SAP HANA accelerator for SAP ASE • `ets_standby` - Standby host for SAP HANA accelerator for SAP ASE • `streaming` - Host for SAP HANA streaming analytics • `xs_worker` - Host for SAP HANA XS advanced runtime • `xs_standby` - Standby host for SAP HANA XS advanced runtime

6. Enter the required credentials.
7. Review the summary, and select *Add Roles* to finalize the configuration.

Related Information

SAPUI5 Developer Guide

12.3.3.2 Removing Host Roles

You can remove host roles from hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program.

Related Information

Remove Host Roles Using the Graphical User Interface [page 1111]
Remove Host Roles Using the Command-Line Interface [page 1112]
Remove Host Roles Using the Web User Interface [page 1113]

12.3.3.2.1 Remove Host Roles Using the Graphical User Interface

You can remove host roles from hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the graphical user interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

   ```
   ./hdblcmgui
   ```

   The SAP HANA database lifecycle manager graphical user interface appears.

> **Note**
>
> To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.

3. Select *Remove Host Roles* from the activity options. Then select *Next*.
4. Select *Remove Roles...* to remove host roles from a host. Then select *Next*.
5. Review the summary, and select *Run* to finalize the configuration.

12.3.3.2.2 Remove Host Roles Using the Command-Line Interface

You can remove host roles from hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>adm`.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<sid>adm`) user may exist on the additional host. Make sure that you have the password of the existing `<sid>adm` user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --action=remove_host_roles
   ```
3. Select the hosts for which you would like to remove roles.

4. Select the host roles that you want to remove for each host.
5. Enter the required credentials.
6. Review the summary, and select *y* to finalize the configuration.

12.3.3.2.3 Remove Host Roles Using the Web User Interface

You can remove host roles from hosts in an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program in the Web user interface.

Prerequisites

- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

  > ### Note
  > For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.
- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://`<hostname>`:1129/lmsl/HDBLCM/`<SID>`/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

2. Select the *Remove Host Roles* tile.
3. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
Do Not Remove XS Advanced OS Users	Prevents the XS advanced runtime OS Users from being removed.
Do Not Start Hosts After Removal of Roles	Does not start hosts after removal of roles.
Do Not Modify 'sudoers' File	Prevents the file `/etc/sudoers.d/sap_hdb_sudoers` from being modified.
Timeouts	Sets customized timeouts (`start_instance`, `start_service`, `stop_instance`, `stop_service`).

4. Select the hosts for which you would like to remove roles.
5. Select the host roles that you want to remove for each host. Then select *Next*.
6. Enter the relevant credentials. Then select *Next*.
7. Review the summary, and select *Remove Roles* to finalize the configuration.

Related Information

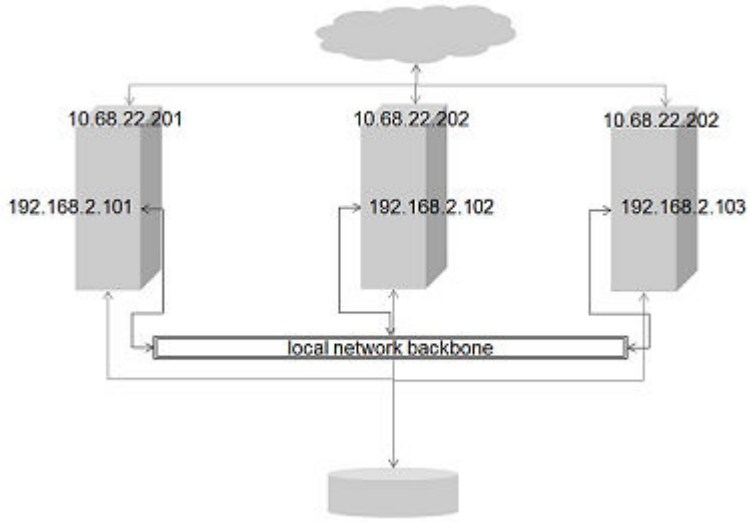
SAPUI5 Developer Guide

12.3.4 Configuring the Network for Multiple Hosts

As part of setting up a distributed system you should configure the network parameters to optimize performance. Make sure you do this before you add additional hosts because one server needs to be available so that you can connect to the SAP HANA studio.

You map host names to IP addresses by editing the section `internal_hostname_resolution` in the `global.ini` file.

General Network Layout



The figure shows a sample cluster with external addresses (10.68.22.*) and internal (192.168.2.*) addresses. To redirect the internal communication over the local network backbone, you could map the internal addresses to the host names of SAP HANA servers as shown in this example:

```
[communication]
listeninterface = .internal
#listeninterface = .global
#listeninterface = .local
#listeninterface = 192.168.2.0/24
[internal_hostname_resolution]
192.168.2.101 = hana01
192.168.2.102 = hana02
192.168.2.103 = hana03
```

For increased security, you can limit the binding of the processes in the `communication` section of the `global.ini` file. The option `listeninterface` can be set in one of the following ways:

- You can set it to one of the predefined keywords:
  - `.global`
  - `.internal`
  - `.local` (default)

  > ⓘ Note
  >
  > You must include the dot at the beginning of the keyword.

- You can set it to a subnet in CIDR notation (classless inter-domain routing).

The `.global` keyword lets the process bind to all interfaces. The `.local` keyword opens the communication ports for internal usage on the local interfaces (which are 127.0.0.1 in IPv4 notation). This configuration is only an option for single host installations as the server is not reachable from the outside. These two options do not require a valid `internal_hostname_resolution` section.

If you specify a keyword other than `.local`, or if you specify a list of networks in CIDR notation, the local interfaces will always be open.

With the `.internal` setup, an `internal_hostname_resolution` section is required. This configuration scans `internal_hostname_resolution` for the local address of the host. The process is bound to this address only (and to all localhost interfaces). So you should add all hosts and their respective addresses to the `global.ini` immediately after installation of the first server. The SAP HANA instance on the first server then needs to be restarted for the changes to take effect. After that, the remaining hosts may be added.

With this configuration the whole landscape uses the internal network immediately after installation. To reduce the possibility of errors, it is also possible to install the whole landscape first without SAP HANA network configuration. This lets you run tests first before you establish the network. Then the configuration options remain the same and the whole SAP HANA landscape needs to be restarted for your changes to take effect.

It is possible to monitor the network using the *Monitor Network* link in the SAP HANA cockpit. The *Measure Network Speed* link on the *Monitor Network* page offers the possibility to measure the network speed between the hosts in a scale-out SAP HANA database. The *Network Speed Check* list offers an overview of all network channels between the involved hosts starting with the slowest network connection.

Further Information

An SAP Community blog *How to configure HANA network communication channels – Part 2: Internal network* gives further details of these features.

Related Information

Internal Host Name Resolution [page 747]
SAP Community Blog: How to configure HANA network communication channels – Part 2: Internal network

12.3.4.1 Configuring SAP HANA Inter-Service Communication

In addition to external network connections, SAP HANA uses separate, dedicated connections exclusively for internal communication. These internal communication channels can be defined using the SAP HANA database lifecycle manager.

In a multiple-host system environment, inter-service communication takes place between the hosts of a multiple-host system on one site. Certified SAP HANA hosts contain a separate network interface card that is configured as part of a private network, using separate IP addresses and ports.

To prevent unauthorized access to the database via the internal communication channels in multiple-host systems, you can isolate internal network ports from client network. To do so, you route communication

between the hosts of a multiple-host environment onto a specified network and bind those internal network services exclusively to the network interface.

In addition, this feature can now be used in the presence of a secondary site (system replication scenario). However, note that additional ports used for communication between primary and secondary sites are opened on the network interface. These ports need to be protected.

> **Note**
>
> In single-host scenarios, the same communication channels are used for communication between the different processes on a single host. The internal IP addresses/ports are by default bound to the local interface. In multi-host scenarios, the specified network prefix must point to a network shared by all hosts. For security reasons, the network should belong to an internal network.

Related Information

Configure SAP HANA Inter-Service Communication Using the Graphical User Interface [page 1117]
Configure SAP HANA Inter-Service Communication Using the Command-Line Interface [page 1119]
Configure SAP HANA Inter-Service Communication Using the Web User Interface [page 1121]

12.3.4.1.1 Configure SAP HANA Inter-Service Communication Using the Graphical User Interface

To prevent unauthorized access to the SAP HANA system via the internal communication channels in multiple-host systems, you can configure inter-service communication using the SAP HANA database lifecycle manager graphical user interface.

Prerequisites

- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on with the required root user or system administrator user `<sid>adm` credentials.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

```
cd <sapmnt>/<SID>/hdblcm
```

By default, `<sapmnt>` is `/hana/shared`.

2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

```
./hdblcmgui
```

The SAP HANA database lifecycle manager graphical user interface appears.

> **Note**
>
> To activate the local secure (LSS) store during installation, run `hdblcmgui` with the parameter `secure_store=localsecurestore`.

3. Select *Configure Inter-Service Communication* from the activity options. Then select *Next*.
4. Define the required parameters. Then select *Next*.

Field Name	Description
Inter-Service Communication	Specifies the listen interface for the internal network communication. `global` - Binds the processes to all interfaces. This option does not require an internal network address entry. `internal` - Binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry. `local` - Opens the communication ports for internal usage on the local interfaces. This configuration is only an option for single installations as the server is not reachable from outside. This option does not require an internal network address entry. If you define a value other than `local`, the local interfaces will always be open.
Internal Network Address	Specifies the internal subset address in CIDR notation.

5. Review the summary, and select *Run* to finalize the configuration.

You can find more information about SAP HANA system internal network and the network security recommendations, in the *SAP HANA Master*, *SAP HANA Security Guide*, and the Network Administration section of this *SAP HANA Administration Guide*.

Results

You have configured the inter-service communication of an SAP HANA system. The parameter values are entered in the `global.ini` configuration file under `[communication]`.

Related Information

Network Administration [page 714]

12.3.4.1.2 Configure SAP HANA Inter-Service Communication Using the Command-Line Interface

To prevent unauthorized access to the SAP HANA system via the internal communication channels in multiple-host systems, you can configure inter-service communication using the SAP HANA database lifecycle manager command-line interface.

Prerequisites

- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on with the required root user or system administrator user `<sid>adm` credentials.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm
   ```
3. Select the index for the *configure_internal_network* action. Then select `Enter`.
4. Define the required parameters.

Field Name	Description
Inter-Service Communication	Specifies the listen interface for the internal network communication. `global` - Binds the processes to all interfaces. This option does not require an internal network address entry. `internal` - Binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry. `local` - Opens the communication ports for internal usage on the local interfaces. This configuration is only an option for single installations as the server is not reachable from outside. This option does not require an internal network address entry. If you define a value other than `local`, the local interfaces will always be open.
Internal Network Address	Specifies the internal subset address in CIDR notation.

For more information about parameters for the `configure_internal_network` action, see Related Information.

5. Review the summary, and select *y* to finalize the configuration.

Results

You have configured the inter-service communication of an SAP HANA system. The parameter values are entered in the `global.ini` configuration file under `[communication]`.

This configuration task can also be performed in batch mode and using a configuration file. For more information about the available configuration methods, see *Using the SAP HANA Platform LCM Tools*.

> **Note**
>
> When using the command line, the options can be set interactively during configuration only if they are marked as interactive in the help description. All other options have to be specified in the command line. To call the help, in the SAP HANA resident HDBLCM directory of the SAP HANA system, execute the following command:
>
> ```
> ./hdblcm --action=configure_internal_network --help
> ```

You can find more information about SAP HANA system internal network and the network security recommendations, in *SAP HANA Master Guide*, *SAP HANA Security Guide*, and in the scaling SAP HANA information in the Network Administration section of this *SAP HANA Administration Guide*.

Example

The following example configures the internal network communication with internal interface:

```
./hdblcm --action=configure_internal_network --listen_interface=internal --
internal_address=10.66.8/21
```

Related Information

Using the SAP HANA Platform LCM Tools [page 585]
Configuring the Network for Multiple Hosts [page 1115]
Network Administration [page 714]
internal_network
nostart [page 75]
timeouts

12.3.4.1.3 Configure SAP HANA Inter-Service Communication Using the Web User Interface

To prevent unauthorized access to the SAP HANA system via the internal communication channels in multiple-host systems, you can configure inter-service communication using the SAP HANA database lifecycle manager Web user interface.

Prerequisites

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

  > **Note**
  >
  > For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide*.
- You are logged on as the system administrator user `<sid>adm`.
- The `<sid>adm` user has read and execute permissions for the directory that contains the installation medium.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser:

Option	Description
	https://<hostname>:1129/lmsl/HDBLCM/<SID>/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA cockpit	1. Enter the URL of the SAP HANA cockpit administration and monitoring console in your browser. `https://<host_FQDN>:<port>` **Note** FQDN = fully qualified domain name 2. Drill down on the name of the system from *My Resources* or from a group. 3. The links in *Platform Lifecycle Management* each launch additional functionality, giving you expanded capabilities for managing the resource.

2. Select the *Configure Inter-Service Communication* tile.
3. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
nostart	Prevents the SAP HANA system from being started.
Timeouts	Sets customized timeouts (`start_instance`, `start_service`, `stop_instance`, `stop_service`).

4. Provide the password of the `<sid>adm` user, then select *Next*.
5. Specify values for the following fields:

Field Name	Description
Inter-Service Communication	Specifies the listen interface for the internal network communication. `global` - Binds the processes to all interfaces. This option does not require an internal network address entry. `internal` - Binds the processes to this address only and to all local host interfaces. This option requires an internal network address entry. `local` - Opens the communication ports for internal usage on the local interfaces. This configuration is only an option for single installations as the server is not reachable from outside. This option does not require an internal network address entry. If you define a value other than `local`, the local interfaces will always be open.
Internal Network Address	Specifies the internal subset address in CIDR notation.

6. Review the summary, and select *Run* to finalize the configuration.

You can find more information about SAP HANA system internal network and the network security recommendations, in the *SAP HANA Master*, *SAP HANA Security Guide*, and the Network Administration section of this *SAP HANA Administration Guide*.

Results

You have configured the inter-service communication of an SAP HANA system. The parameter values are entered in the `global.ini` configuration file under `[communication]`.

Related Information

SAPUI5 Developer Guide
Add an SAP HANA System
Network Administration [page 714]

12.3.5 Scaling SAP HANA Extended Application Services, Classic Model

If you have an application based on SAP HANA XS classic, you can configure multiple SAP HANA XS instances to work in a scale out SAP HANA system.

Context

If you are expecting a high degree of concurrency, you may want to distribute the XS classic server across the various hosts in your system. This is not enabled at the system level by default. You can manually change this setting to the system level in the Administration editor by performing the following steps.

Procedure

1. In the Administration editor, choose the *Configuration* tab.
2. Enter the string "instances" in the Filter box.
   This search string returns a list of instances.
3. Change the instances setting to system level
   a. Set the value of instances to "1" on the system level for `xsengine` and `webdispatcher`.
4. Clear any entries on host level
   a. Right-click on the green circle and choose *Delete* to clear any entries on the host level
   b. In the *Delete Configuration Value* dialog box select the check box beside *HOST* layer and choose *Delete*.

Related Information

Change a System Property in SAP HANA Studio
Configure HTTP Load Balancing for SAP HANA Extended Application Services, Classic Model [page 900]

12.3.6 Starting and Stopping Distributed SAP HANA Systems Using SAPControl

You can use `SAPControl` to start or stop all the hosts in a scaled-out SAP HANA system from the command line.

> **Note**
>
> You must be logged on to the SAP system host as user `<sid>adm` or as a user with root permissions.

Action	Command
Start the system	`/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> -function StartSystem HDB`
Stop the system	`/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> -function StopSystem HDB`
Query current status of all hosts in the system	`/usr/sap/hostctrl/exe/sapcontrol -nr <instance_number> -function GetSystemInstanceList`

> **Note**
>
> HDB start or HDB stop only starts and stops the local host.

13 SAP HANA Deployment Infrastructure (DI) Administration

SAP HDI provides tools that enable the deployment of database artifacts to the run-time environment.

The SAP HANA Deployment Infrastructure, or HDI for short, provides a service that enables you to deploy database development artifacts to so-called containers. This service includes a family of consistent design-time artifacts for all key SAP HANA platform database features, which describe the target (run-time) state of SAP HANA database artifacts, for example: tables, views, or procedures. These artifacts are modeled, staged (uploaded), built, and deployed into SAP HANA.

> ⚠ Restriction
>
> The HDI focuses strictly on deployment; HDI does not include any version-control tools, nor does it provide any tools for life-cycle management.

HDI provides its services using a separate database process named `diserver`. On systems where XS advanced is installed, HDI is already enabled. On systems where HDI is not already enabled, the `diserver` process must usually be enabled by the database administrator before HDI can be used. If required by the usage scenario, other database process may also need to be started as well.

In the context of SAP HANA extended application services (XS advanced), the SAP HANA service broker is used to create and drop HDI containers; each HDI container comprises a design-time container (DTC) and a run-time container (RTC). The HDI deployment tools deploy database artifacts to an HDI container. Design-time database objects are typically located in the `db/` folder of the application design-time hierarchy. The deployment process populates the database run-time with the specified catalog objects. In addition to database artifacts, HDI also enables you to import and export table content such as business configuration data and translatable texts.

> → Tip
>
> For more information about HDI administrator roles, the tasks each HDI administrator is expected to perform, and the scope of each HDI administrator's responsibility, see *SAP HANA Deployment Infrastructure Reference* in *Related Information* below. This HDI Reference also includes details of the SAP SQL API for run-time content development in HANA DI as well as the complete list of database-related, design-time artifacts and the corresponding build plug-in.

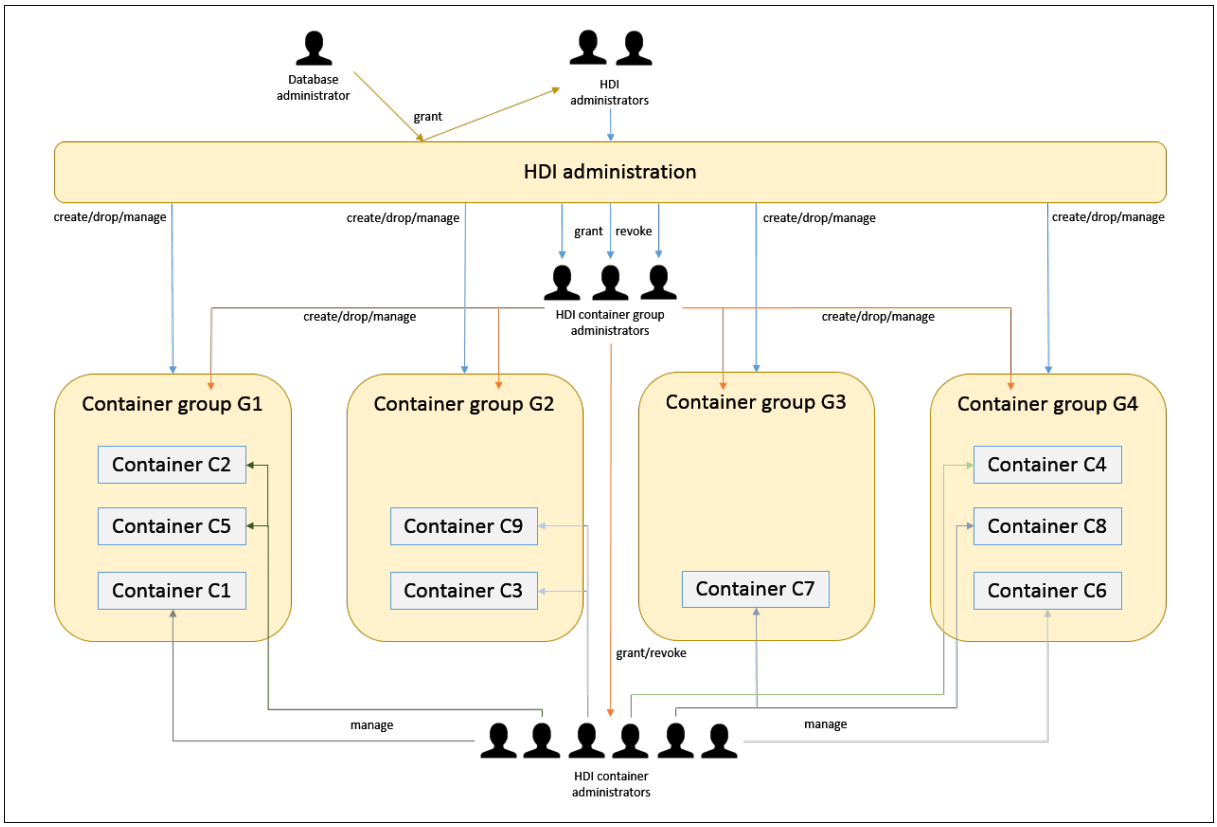
Related Information

The SAP HANA Deployment Infrastructure Reference

13.1 SAP HANA DI Administration at a Glance

An overview of the HDI administration process including the administrator users who set up and maintain HDI and its components.

The relationships between the various roles involved in the maintenance of HDI containers and the corresponding scope of each role is shown in the following diagram:



HDI Administration

HDI provides its services using a separate database process named `diserver`. On systems where XS advanced is installed, HDI is already enabled; on other systems where XS advanced is not installed, the `diserver` process must usually be enabled by the database administrator before HDI can be used. If required by the usage scenario, other database process may also need to be started as well.

The database administrator SYSTEM is needed for the first-time enabling of HDI in SAP HANA and for creating an HDI administrator, who then performs the tasks required to set up and maintain other HDI administrators, if required.

> **Note**
>
> SYSTEM user privileges are required to create the first HDI administrator, who can then create other HDI administrators. After creation of the first HDI administrator, the SYSTEM user can be deactivated.

Created by the database administrator, the HDI administrator is responsible for the setup and overall maintenance of HDI. The role of the HDI administrator includes configuring general HDI parameters, maintaining containers and container groups (for example, by creating and dropping containers and container

groups), and managing container-group administrator privileges, for example, by granting and revoking HDI container-group access permissions.

The HDI container-group administrator manages a set of containers in container groups assigned by the HDI administrator. Container-group management tasks include: granting and revoking container (and container-group) **administrator** access privileges; granting and revoking container **user** access privileges (for example, for temporary support purposes); maintaining containers and container groups.

> → Tip
>
> The APIs of a container group "G" are in the `_SYS_DI#G` schema.

The HDI container administrator manages one or more containers assigned by the container-group administrator. The role of the container-manager focuses primarily on configuring and controlling access to the HDI containers used to store the database objects deployed by the SAP HANA Deployment Infrastructure deploy service and repairing any problems that occur with run-time objects in the assigned HDI containers. An HDI container administrator can manage one or more containers in one HDI container group or multiple containers distributed across multiple container groups.

> → Tip
>
> The APIs of a container "C" are in the `C#DI` schema.

HDI Containers

The SAP HANA Deployment Infrastructure (HDI) provides a service that enables you to deploy database development artifacts to so-called containers. This service includes a family of consistent design-time artifacts for all key HANA platform database features which describe the target (run-time) state of SAP HANA database artifacts, for example: tables, views, or procedures. These artifacts are modeled, staged (uploaded), built, and deployed into SAP HANA.

> ⓘ Note
>
> The HDI focuses strictly on deployment; HDI does not include any version-control tools, nor does it provide any tools for life-cycle management.

The SAP HANA service broker is used to create and destroy HDI containers; each HDI container comprises a design-time container (DTC), which is an isolated environment used to store design-time files, and a run-time container (RTC), which is used to store deployed objects built according to the specification stored in the corresponding design-time artifacts.

The deployment process populates the database run-time with the specified catalog objects. In addition to database artifacts, HDI also enables you to import and export table content such as business configuration data and translatable texts.

> ⚠ Restriction
>
> HDI enables you to deploy database objects only; it is not possible (or necessary) to deploy application-layer artifacts such as JavaScript programs or OData objects.

The configuration of HDI containers also involves the creation and configuration of the following design-time artifacts:

- Container deployment configuration (`.hdiconfig`)

A JSON file containing a list of the bindings between database artifact types (for example, sequence, procedure, table) and the corresponding deployment plug in (and version).

- Run-time container namespace rules (`.hdinamespace`)
  A JSON file containing a list of design-time file suffixes and the naming rules for the corresponding runtime locations.

  > → Tip
  >
  > You can apply the rules defined in the `.hdinamespace` file either exclusively to the folder it is located in or to the folder it is located in **and** its subfolders.

HDI Container Groups

HDI container groups are logical collections of the HDI containers used to store the objects deployed by the SAP HANA Deployment Infrastructure deploy service. After creation, an HDI container group can be assigned to a dedicated container-group administrator, who must be granted the privileges required to perform the typical tasks associated with the administration of container group, for example: granting and revoking container (and container-group) access privileges; and maintaining container groups (and the containers assigned to the groups).

> → Tip
>
> Container groups are intended to make life easier when multiple administrators require access to containers from different contexts, for example, XS advanced, ABAP, or other development groups working in native SAP HANA contexts.

The HDI administrator can create container groups for a target audience whose container-related requirements are unique or where there is an obvious benefit for a logical separation, for example, between ABAP and applications running in the XS advanced model run-time environment.

> → Tip
>
> For more information about HDI administrator roles, the tasks each HDI administrator is expected to perform, and the scope of each HDI administrator's responsibility, see *SAP HANA Deployment Infrastructure Reference* in *Related Information* below. This HDI Reference also included details of the SAP SQL API for run-time content development in HANA DI as well as the complete list of database-related design-time artifacts and the corresponding build plug-in.

Related Information

The SAP HANA Deployment Infrastructure Reference

13.2 SAP HANA DI Administrator Roles

HDI administration involves a number of tasks that must be performed by different administrator roles.

The following table describes the scope of the various HDI administrator roles and lists the most common tasks the administrators are expected to perform.

HDI Admin Scope, Roles, and Tasks

HDI Role	Description	Common Tasks
Database administrator	The database administrator SYSTEM is needed for initially enabling HDI and for creating an HDI administrator. **Note** SYSTEM user privileges are required to create the first HDI administrator, who can then create other HDI administrators. After creation of the first HDI administrator, the SYSTEM user can be deactivated.	Enable HDI Create an HDI administrator Grant and revoke HDI administrator privileges
HDI administrator	Configures general HDI parameters, maintains containers and container groups, and manages container group administrator privileges.	Configure HDI Create and drop containers and container groups Grant and revoke required access privileges Maintain containers and container groups Move containers between container groups
HDI container-group administrator	Manages the container groups assigned by the HDI administrator. The APIs of a container group "G" are in the `_SYS_DI#G` schema.	Grant and revoke container (and container-group) **administrator** access privileges Import and export containers (for support purposes) Grant and revoke container **user** access privileges (for support purposes) Maintain containers and container groups

HDI Role	Description	Common Tasks
HDI container administrator	Configures and controls access to a container and manages run-time objects in the assigned containers. The APIs of a container "C" are in the `C#DI` schema.	Grant and revoke container **administrator** access privileges Configure libraries and parameters Grant and revoke roles from schemas to users Grant and revoke user access to container schemas Cancel an asynchronous make operation

→ Tip

For more information about HDI administrator roles, the tasks each HDI administrator is expected to perform, and the scope of each HDI administrator's responsibility, see *SAP HANA Deployment Infrastructure Reference* in *Related Information* below. This HDI Reference also includes details of the SAP SQL API for run-time content development in HANA SAP DI as well as the complete list of database-related design-time artifacts and the corresponding build plug-in.

Related Information

The SAP HANA Deployment Infrastructure Reference

13.3 SAP HANA DI Administrator Tasks

Learn how responsibility for maintenance tasks is split between SAP HANA Database Infrastructure (HDI) administrators.

HDI provides its services using a separate database process named `diserver`. On systems where XS advanced is installed, HDI is already enabled; on other systems where XS advanced is not installed, the `diserver` process must usually be enabled by the database administrator before HDI can be used.

ⓘ Note

If required by the usage scenario, other database processes may also need to be started as well.

Maintaining the HDI, its containers, and container groups, involves the following high-level administration tasks:

- Enabling the HDI
  A database administrator with SYSTEM privileges starts the HDI for the first time, creates the necessary administrator users, and assigns the new users the access privileges required to administrate the HDI.
- Maintaining the HDI

An HDI administrator configures HDI, creates HDI container groups, and grants and revokes the access privileges required by the HDI container-group administrators.

- Maintaining HDI containers groups
  HDI container-group administrators drop and create HDI containers, grant and revoke container (and container-group) access privileges, and import and export containers (for support purposes).
- Maintaining HDI containers
  HDI container administrators grant and revoke container-based access privileges, configure libraries and parameters, and grant and revoke access to a HDI container's schemas.

> → Tip
>
> For more information about HDI administrator roles, the tasks each HDI administrator is expected to perform, and the scope of each HDI administrator's responsibility, see *SAP HANA Deployment Infrastructure Reference* in *Related Information* below. This HDI Reference also included details of the SAP SQL API for run-time content development in HANA DI as well as the complete list of database-related design-time artifacts and the corresponding build plug-in.

Related Information

The SAP HANA Deployment Infrastructure Reference

13.4 The SQL API for SAP HANA DI

An SQL application programming interface (API) is available to help maintain the SAP HANA Deployment Infrastructure (DI).

SAP HANA Deployment Infrastructure (HDI) can be seen as a layer on top of the SAP HANA database. The HDI includes an SQL-based API that is accessible by means of standard SAP HANA SQL connection data. The following APIs are provided to help you manage and work with HDI:

SQL APIs for HDI

HDI API	Description
HDI administration API	Used mainly for managing container groups and the administrative access to them. Use of the HDI administration API requires the privileges of an HDI administrator. **Note** SYSTEM user privileges are required to create the first HDI administrator, who can then create other HDI administrators, if required. After creation of the first HDI administrator, the SYSTEM user can be deactivated.

HDI API	Description
HDI container group administration API	Used for managing a set of containers inside a container group and the administrative access to them. To use this API, a user needs to be granted the privileges of an HDI container group administrator by an HDI administrator.
HDI container administration API	Used for configuring a container and controlling the access to it. To use this API, a user needs to be granted the privileges of an HDI container administrator by an HDI container group administrator.
HDI container content development API:	Used for enabling applications to deploy or undeploy HDI artifacts within a container and manage access to the containers. To use this API, a user must be granted the privileges of an HDI container content developer by an HDI container administrator.

→ Tip

For more information about HDI administrator roles, the tasks each HDI administrator is expected to perform, and the scope of each HDI administrator's responsibility, see *SAP HANA Deployment Infrastructure Reference* in *Related Information* below. This HDI Reference also includes details of the SAP SQL API for run-time content development in SAP HANA DI as well as the complete list of database-related design-time artifacts and the corresponding build plug-in.

Related Information

The SAP HANA Deployment Infrastructure Reference

13.5 SAP HANA DI Security

An overview of the tools used to configure and ensure security in the SAP HANA Deployment Infrastructure (HDI).

The following list provides an overview of the security aspects that you need to consider when setting up and using the SAP HDI:

ⓘ Note

For more information about an individual security area, see the *SAP HDI Administration* section of the *The SAP HANA Deployment Infrastructure Reference* in *Related information* below.

- Technical System Landscape of SAP HDI
  An overview of the architecture of SAP HDI and which protocols for communication between the various high-level components
- SAP HDI Users
  An overview of the predefined users which SAP HDI relies on and a description of each user's scope

- SAP HDI Database Roles
  An overview of the predefined database roles that are required to operate the SAP HANA Deployment Infrastructure (HDI) along with a description of each role
- Data Storage Security in SAP HDI
  An overview of the mechanisms used to secure and protect critical data managed by the SAP HDI
- Network and Communication Security with SAP HDI
  An overview of the security mechanisms applied to networking and communication in the context of SAP HDI
- Security-Relevant Logging and Tracing in SAP HDI
  An overview of the auditing process, which enables you to trace the different kinds of operations performed by HDI users in the context of SAP HDI
- Data Protection and Privacy in SAP HDI
  An overview of SAP HDI's adherence to (and compliance with) data protection and privacy regulations
- SAP HDI Security in the Context of XS Advanced
  An overview of the security considerations to bear in mind when enabling SAP HDI for use in the context of SAP HANA extended application services, advanced model

> **Note**
>
> SAP HDI is a service layer on top of the SAP HANA platform, so information about other components in other sections of the SAP HANA Security Guide might also apply.

Related Information

SAP HANA Deployment Infrastructure (HDI) Reference for SAP HANA Platform

14 Application Run-Time Services

Maintain the SAP HANA XS run-time environment for XS classic and XS advanced applications.

The SAP HANA administration cockpit provides the tools you need to maintain and manage the various components of the SAP HANA XS run-time environment. Whether you are providing adminstration and support services for applications running in the XS classic run time or you need to set up and maintain an XS advanced run time in SAP HANA, the administration cockpit provides a selection of tools to help you perform your tasks quickly and easily.

- SAP HANA XS classic model
  Maintain and manage the various components of the SAP HANA XS classic Model (XS classic) run-time environment
- SAP HANA XS advanced model
  Maintain and manage the various components of the SAP HANA XS Advanced Model (XS advanced) run-time environment

Related Information

Maintaining the SAP HANA XS Classic Model Run Time [page 1134]
Maintaining the SAP HANA XS Advanced Model Run Time [page 1256]

14.1 Maintaining the SAP HANA XS Classic Model Run Time

Maintain the SAP HANA XS classic model run-time environment.

A number of administration tools are available to enable you to maintain and manage the various components of the SAP HANA XS classic model (XS classic) run-time environment. In the SAP HANA administration cockpit, the *XS Administration* tile catalog contains the *Administration and Monitoring* tile, which contains the following tools:

> **Note**
>
> In the SAP HANA cockpit, tiles and tile catalogs are only visible to users who have been assigned the privileges granted by role `sap.hana.uis.db::SITE_DESIGNER`. In addition, some of the tools listed below are only available to users to whom the suitable role has been assigned. For example, a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator` includes the authorization required for unrestricted access to all the tools used to manage the configuration settings for SAP HANA XS application security and the related user-authentication providers; a role based on the role template `sap.hana.xs.admin.roles::SAMLAdministrator` enables unrestricted access only to the *SAML Identity Providers Configuration* tools.

- *XS Artifact Administration*
  Monitor the system usage of the applications running in the XS Advanced Model run-time
- *SAML Service Provider*
  Configure an SAP HANA system to act as an SAML service provider for SSO authentication.
- *SAML Identity Provider*
  Configure an SAML identity provider for use by the SAML service provider to authenticate the users signing in by means of SSO.
- *SMTP Configuration*
  Maintain and manage details of the SMTP server that is available for use by all applications running on an SAP HANA XS classic model server.
- *Trust Manager*
  Configure SAML Identity providers (IDP) for SAP HANA XS classic model applications that use SAML assertions as the log-on authentication method.
- *XS Job Dashboard*
  Create, schedule, and manage long running operations jobs in the SAP HANA XS classic model run-time environment.

Related Information

SAP HANA XS Classic Administration Tools [page 1135]
SAP HANA XS Classic Administration Roles [page 1137]
SAP HANA XS Classic Configuration Parameters [page 1140]

14.1.1 SAP HANA XS Classic Administration Tools

SAP HANA XS includes a Web-based tool that enables you to maintain important parts of the XS classic application-development environment, for example, security and authentication methods.

The *SAP HANA XS Administration Tool* is a Web-based tool that enables you to configure and maintain the basic administration-related elements of the XS Classic application-development process and environment.

> → Tip
>
> The SAP HANA XS Administration Tool for XS Classic is available on the SAP HANA XS Web server at the following URL:
>
> `http://<WebServerHost>:80<SAPHANAinstanceID>/sap/hana/xs/admin/`

The features included in the Web-based *SAP HANA XS Administration Tool* for XS Classic cover the following areas:

> ⓘ Note
>
> The availability of screens, tabs, and UI controls (for example, *Add*, *Edit*, or *Save* buttons) is based on the privileges granted in the assigned user roles. For example, a user who has a role based on the role template `sap.hana.xs.admin.roles::HTTPDestViewer` can view HTTP destinations; a user assigned

a role based on the role template `sap.hana.xs.admin.roles::SQLCCAdministrator` can not only view but also **edit** SQL connection configurations.

Administration Tools for SAP HANA XS Classic Applications

Tool Name	Description	Scope
XS Artifact Administration	Maintain run-time configurations for individual applications or a complete application hierarchy. The configuration defined for an application is inherited by any application further down the application package hierarchy.	• Application security (public/private) • User-authentication methods (basic, form-based, logon tickets, X509, SAML) • CORS setup for cross-origin resource sharing • Custom headers: enable support for X-Frame-Options HTTP header • HTTP destinations • SQL connection configurations (for SQL connections for users other than the user specified in the HTTP request.
SAML Service Provider	Configure an SAP HANA system to act as an SAML service provider for SSO authentication.	• Management of SAML **service**-providers, including URLs and metadata management
SAML Identity Provider	Configure an SAML identity provider for use by the SAML service provider to authenticate the users signing in by means of SSO.	• Management of SAML **identity**-providers, including IDP metadata, certificates, and destinations
SMTP Configuration	Define the details of the SMTP server that is available for use by all applications running on an SAP HANA XS server.	• SMTP host settings • Authentication type • Transport security
Trust Manager	Maintain the certificates used to establish trust relationships between servers used by SAP HANA XS applications.	• Trust store configuration and management • Certificate management
XS Job Dashboard	Monitor and maintain SAP HANA XS job schedules defined using the XS job syntax	• Enable the job scheduler • Monitor job-schedule status • Display and maintain schedule's run-time configuration • Add schedules to (or delete from) an XS job

Additional Tools in SAP HANA XS Classic

The following table lists some tools that are not strictly part of the SAP HANA XS Administration tool set. The tools are included here primarily for the sake of convenience but also because the tools are installed with the delivery unit which contains the XS Administration tools.

Translation Text Details

Tool Name	Description	Scope
Online Translation Tool	Maintain translations, for example, for UI text elements	• Add, modify, delete translation texts • Export translation text from SAP HANA to an XML-based xlif-format file • Import translation text into SAP HANA.
User Self Service Tools	A set of tools that enable you to maintain user self-service requests and administrate the self-service tools themselves.	• Activate the user self-service tools • Maintain user self-service requests • Maintain user black/white lists • Maintain user self-service e-mail templates

Related Information

Maintaining Application Runtime Configurations [page 1146]
Maintaining SAML Providers [page 1167]
Managing Trust Relationships [page 1160]
Maintaining SMTP Server Configurations [page 1176]
Scheduling XS Jobs [page 1227]
Maintaining User Self Service Tools [page 1202]
Maintaining Translation Text Strings [page 1241]

14.1.2 SAP HANA XS Classic Administration Roles

SAP HANA uses roles to control access to the Web-based tool that enable you to maintain important parts of the application-development environment, for example, security and authentication methods.

When using the Web-based tools provided by SAP HANA XS, the availability of features, screens, tabs, and UI controls (for example, *Add*, *Edit*, or *Save*, or *Delete* buttons) is based on privileges. For the sake of convenience, the specific privileges required to use the features provided with a particular tool have been collected into a selection of predefined roles, which you can use as templates to create your own roles and assign to the user who wants to use a tool. For example, a user assigned a role based on `sap.hana.xs.admin.roles::HTTPDestViewer` can display HTTP destinations but not change them in any way; a user assigned a role based on `sap.hana.xs.admin.roles::SQLCCAdministrator` can view SQL connection configurations and modify them, too.

> → Recommendation
>
> As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your

organization's security policy. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

SAP HANA XS Administration Tools Roles

SAP HANA XS Role	Description
HTTPDestAdministrator	Full access to the details of HTTP destination configurations (display and edit)
HTTPDestViewer	Read-only access to HTTP destination configurations, which are used to specify connection details for outbound connections, for example, using the server-side JavaScript Connectivity API that is included with SAP HANA XS.
RuntimeConfAdministrator	Full access to the configuration settings for SAP HANA XS application security and the related user-authentication providers.
RuntimeConfViewer	Read-only access to the configuration settings for SAP HANA XS application security and the related user-authentication providers, for example, SAML or X509.
JobAdministrator	Full access to the configuration settings for SAP HANA XS job schedules (defined in `.xsjob` files); you can specify start/stop times, the user account to run the job, and the language locale.
JobViewer	Read-only access to the configuration settings for SAP HANA XS job schedules (defined in `.xsjob` files).
JobSchedulerAdministrator	Full access to the *XS Job Dashboard* tool, which you can use to add and delete XS job schedules, maintain individual schedules, and enable the scheduling feature.
oAuthAdmin	Required when setting the client secret during administration of the OAuth client configuration (`.xsoauthclientconfig`) artifact.
SAMLAdministrator	Full access to the details of SAML configurations, including both the service provider and the identity providers. You can add new entries and make changes to existing service or identity providers and parse the resulting metadata.
SAMLViewer	Read-only access to SAML configurations, which are used to provide details of SAML service providers and identity providers.
SMTPDestAdministrator	Full access to the details of SMTP destination configurations, which are used to define details of the SMTP relay server that SAP HANA XS applications use to send e-mails. The administrator role enables you to add new entries and make changes to an existing configuration, for example, the host name and port number, logon credentials and authentication type, and any transport security settings.
SMTPDestViewer	Read-only access to SMTP destination configurations, which are used to define details of the SMTP relay server that SAP HANA XS applications can use to send e-mails.
SQLCCAdministrator	Full access to the details of SQL connection configurations (SQLCC).
SQLCCViewer	Read-only access to SQL connection configurations (SQLCC), which are used to enable the execution of SQL statements from inside your server-side JavaScript application with credentials that are different to the credentials of the requesting user.
TrustStoreAdministrator	Full access to the SAP HANA XS *Trust Manager* tool, which the administrator uses to maintain secure **outbound** communication, for example, the SSL/TLS certificates required by SAP HANA XS applications that connect to an ABAP system.

SAP HANA XS Role	Description
TrustStoreViewer	Read-only access to the trust store, which contains the server's root certificate or the certificate of the certification authority that signed the server's certificate.

Additional SAP HANA XS Roles

The following table lists roles for tools that are not strictly part of the SAP HANA XS Administration toolset. The roles are included here for the sake of convenience and because the roles, and the tools to which they correspond, are (with the exception of the *WebDispatcherAdmin/Viewer*) installed with the delivery unit which contains the XS Administration tools.

Additional Roles for SAP HANA XS Administration Toos

SAP HANA XS Role	Description
translator	The role *sap.hana.xs.translationTool.roles::translator* enables an SAP HANA user to maintain translation text strings with the SAP HANA Online Translation Tool.
USSAdministrator	The role *sap.hana.xs.selfService.admin.roles::USSAdministrator* is assigned to the user responsible for administrating the requests sent by users using self-service tools. For example, it enables the activation of users who request a new user account in the SAP HANA database and allows the user-self-service administrator to manage self-service-specific blacklists for users, e-mail addresses, domains, and IP addresses.
USSExecutor	The role *sap.hana.xs.selfService.user.roles::USSExecutor* is assigned to the technical user that is used to respond to and execute user-self-service requests, for example, to create an new account or request a new password
WebDispatcherAdmin	The role *sap.hana.xs.wdisp.admin::WebDispatcherAdmin* enables full access to the SAP HANA *Web Dispatcher Administration* tool, which the administrator uses to maintain secure **inbound** communication, for example, to enable SSL/TLS connections between an ABAP system and an SAP HANA XS application.
WebDispatcherMonitor	The role *sap.hana.xs.wdisp.admin::WebDispatcherMonitor* enables read-only access to the information displayed in the SAP HANA *Web Dispatcher Administration* tool.
WebDispatcherHTTPTracingViewer	Read-only access to the HTTP setting of SAP HANA XS applications running on the selected SAP HANA instance. This role extends the JobViewer role to enable the user to view details of the xsjob configuration (`httptracing.xsjob`) that starts and stops the HTTP tracing tasks.
WebDispatcherHTTPTracingAdministrator	Full access required to maintain HTTP tracing on the SAP Web Dispatcher for SAP HANA XS applications. This role extends the *JobAdministrator* role to enable the user to maintain the XS job file (`httptracing.xsjob`) used to configure and enable HTTP tracing for XS applications on the SAP Web Dispatcher.

14.1.3 SAP HANA XS Classic Configuration Parameters

An overview of the parameters that the administrator can set to configure how the various components of the XS engine work.

The `xsengine.ini` section of the SAP HANA configuration screen is split into a number of subsections, each of which reflects one of the individual components of the SAP HANA XS engine. Each section contains one or more parameters whose values you can change, where appropriate, to suit the requirements of your system landscape. To display the configuration details of the XS engine in SAP HANA studio, double-click a system in the *Systems* view, choose the *Configuration* tab, and expand the *xsengine.ini* element.

> ### Note
>
> For security reasons, all parameters in the `communication` section of **all** `.ini` configuration files are blacklisted by default; properties included in a blacklist can only be changed by a system administrator. For more information, see *Default Blacklisted System Properties* in the *SAP HANA Administration Guide*.

XS Engine Configuration Parameters (xsengine.ini)

Configuration Section	Description
application_container [page 1141]	Application-related configuration settings, for example, the list of applications that are trusted by the XS engine or the libraries that can be loaded from an `xscfunc` call.
authentication [page 1141]	Options for application-related authentication settings, for example, the location of trust stores.
communication [page 1141]	Options for application-related connection requests and configuration, for example, time-outs, port numbers, and maximum number of data end points allowed by the XS engine
customer_usage [page 1142]	Options for customer-specific usage scenarios in SAP HANA application services, for example, to enable HTTP tracing of XS applications on the SAP Web Dispatcher.
debugger [page 1142]	Settings for the debugging tools, for example, for XS JavaScript.
httpserver [page 1143]	Options for the SAP HANA XS Web server, for example, port numbers, and maximum number of sessions and threads allowed
odata [page 1144]	Configuration settings for OData requests
scheduler [page 1144]	Configuration options for the XS job scheduler, which is used to run an XS Javascript or SQLScript as a task in the background at regular intervals

> ### Note
>
> Some configuration parameters for the SAP HANA XS engine require additional parameters to be set for other SAP HANA components, for example, the SAP Web Dispatcher.

SAP Web Dispatcher Configuration Parameters (webdispatcher.ini)

Configuration Section	Description
webdispatcher.ini/profile [page 1144]	Configuration options for the SAP Web Dispatcher, for example: HTTP tracing of SAP HANA XS applications, logs, allowed connections

application_container

Use the `application_container` section of the `xsengine.ini` file to set configuration options for the application container component of the SAP HANA XS engine, which includes not only the XS application container, but also containers for C++ and JavaScript applications. In this section of hte `xsengine.ini` file, you can modify the list of applications that are trusted by the XS engine or the libraries that can be loaded from an `xscfunc` call.

Parameter	Description	Example Value	Default Value
application_list	Comma-separated list of libraries that can be loaded from an `xscfunc` call	`libxsdxc, InformationAccess`	`libxsdxc, InformationAccess, libtrustmanager, libxsauthenticator, libxsbase`

authentication

Use the `authentication` section of the `xsengine.ini` file to set configuration options for application-related authentication settings, for example, the system ID and hostname of the server providing SAP logon certificates for single sign-on (SSO) purposes.

Parameter	Description	Example Value	Default Value
logonticket_redirect_url	URL that is used to redirect the client to a system that provides SAP logon tickets for SSO authentication	`http://link.to.portal/loginService`	None

communication

Use the `communication` section of the `xsengine.ini` file to set configuration options for application-related connection requests to SAP HANA, for example, timeouts, port numbers, and maximum number of data end points allowed by the XS engine.

Parameter	Description	Example Value	Default Value
default_read_timeout	Time (in milliseconds) before a connection request is closed	-1, 30, 60	-1 (no time set)
default_read_timeout_override	Ignore setting for `default_read_timeout`	No, Yes	Yes
listenport	The port number on which the XS Web server listens for requests	30007	3$(SAPSYSTEM)07

Parameter	Description	Example Value	Default Value
enforced_http_proxy	Override the outgoing proxy settings used for the HTTP/S client, for example, defined in an HTTP/SMTP destination configuration or an `httpclient.request()` method.	myhost.name.com	None
enforced_https_proxy		myhost.name.com	None
enforced_outbound_proxy	Set the proxy not just for HTTP and HTTPS but for **all** outgoing protocols, for example: SMTP, socks, ...	myhost.name.com	None
maxchannels	Maximum number of concurrent channels allowed by the XS Web server	Unsigned integer, for example, 1000	4000
maxendpoints	Maximum number of concurrent data endpoints that the XS Web server can expose	Unsigned integer, for example, 1000	4000

customer_usage

The `customer_usage` section of the `xsengine.ini` file is used by the *SAP Web Dispatcher HTTP Tracing* tool to set configuration for SAP HANA application services, for example, to enable HTTP tracing of XS applications on the SAP Web Dispatcher.

Parameter	Description	Example Value	Default Value
`/path/to/the/XSapp`	The fully qualified path to (and the name of) the application to be traced, for example, `sap.hana.ide` or `sap.hana.xs.admin`. → Tip The parameter value `icm/HTTP/logging_n` is the same as the key defined in the `webdispatcher.ini/profile` section of the configuration parameters, and "n" must be a unique number.	icm/HTTP/logging_n	N/A The parameter `/path/to/XSapp` is set (or removed) automatically when the administrator uses the *XS Admin Tools* to enable (or disable) HTTP tracing on the SAP Web Dispatcher for an application.

debugger

Use the `degugger` section of the `xsengine.ini` file to set configuration options for the SAP HANA XS JavaScript debugging tools.

Parameter	Description	Example Value	Default Value
enabled	Enable debugging functionality	True/False	False

httpserver

Use the `httpserver` section of the `xsengine.ini` file to set configuration options for the SAP HANA XS Web server, for example, port numbers, and maximum number of sessions and threads allowed.

Parameter	Description	Example Value	Default Value
developer_mode	Enable verbose output for HTTP codes/messages	True/False	False
embedded	Enable the SAP HANA XS engine to run in embedded mode (in the index server). See SAP Note 1849775. **Note** Restart the SAP HANA instance if you change the parameter value.	True/False	False
login_screen_background_image	URL to the image displayed as background in the logon screen, with the following prerequisites: • File must be reachable by http(s) • No requirement for authentication or authorization • Recommended minimum resolution of image: 1600*1200 • A technical user has to be assigned to the XSSQLCC artifact `/sap/hana/xs/selfService/user/selfService.xssqlcc`. The technical user must be assigned the role *sap.hana.xs.selfService.user.roles.USSExecutor*. This user will be used to query the details from the server.	/sap/hana/xs/ui/Image.jpg	None
max_message_size_mb	Maximum allowed size (in megabites) of an HTTP request or response	Unsigned integer, for example, 10	100
max_request_runtime	Maximum runtime (in seconds) of an HTTP request targeting an XSJS application. Can be extended in case of long-running database operations.	Unsigned integer, for example, 10	300
maxsessions	Maximum number of registered sessions, not including unauthenticated sessions that are not being debugged	Unsigned integer, for example, 10000	50,000

Parameter	Description	Example Value	Default Value
root_page	Enables requests to the root URI "/" tol be redirected to the URI set wiht this parameter	/sap/xs/path/root.html	None
sessiontimeout	Amount of time (in seconds) before an inactive session is closed	Unsigned integer, for example, 60	900

odata

Use the `odata` section of the `xsengine.ini` file to set configuration options for OData requests.

Parameter	Description	Example Value	Default Value
allow_nullable_keys	Specify if "`key`" entity elements can (`null`) or cannot (`not null`) have the value NULL.	True/False	False

scheduler

Use the `scheduler` section of the `xsengine.ini` file to set configuration options for the XS job scheduler, which is used to run an XS Javascript or SQLScript as a task in the background at regular intervals

Parameter	Description	Example Value	Default Value
enabled	Activate the XS job-scheduler service. Set to `true` on one XS host only; this enables `xsjob` scheduling for the selected instance	True/False	False
sessiontimeout	The amount of time (in seconds) to wait for a job to complete	300 seconds	900 seconds
disable_job_after_restarts	The maximum number of unsuccessful attempts to start a job before the job schedule is automatically disabled	3	5

webdispatcher.ini/profile

Use the `profile` section of the `webdispatcher.ini` file to set configuration options for customer-specific usage scenarios in SAP HANA application services, for example, to enable HTTP tracing of XS applications on the SAP Web Dispatcher.

Parameter	Description	Example Value	Default Value
`icm/HTTP/logging_n`	Defines the application-specific log, where "`_n`" is a unique number. The key's value defines the following: • `PREFIX=` the fully qualified path to (and the name of) the application to be traced, for example, `sap.hana.xs.admin` • `LOGFILE=` the location of the log file used to store the trace information; the location includes a variable for the application's name (`access_log_app-`) and the year, month, and day (`%y-%m-%d`) • MAXSIZEKB= the maximum allowed size and format of the trace file • SWITCHTF=the time of the day when the new log file is created (DAY/NIGHT) • LOGFORMAT= the format of the trace file content, for example: CLF (common log format), CLFMOD (modified CLF), SAP (SAP log file format), SAPSMD, ... • FLUSH=enable or disable the log flush mechanism → Tip The parameter `icm/HTTP/logging_n` is also used as the value for the key defined in the `customer_usage` section of the `xsengine.ini` file.	`PREFIX=/sap/hana/ide/, LOGFILE=$(_LOCAL_HOST_NAME)/trace/access_log_sap.hana.ide-%y-%m-%d, MAXSIZEKB=10000 SWITCHTF=day, LOGFORMAT=SAP, FLUSH=1`	N/A The parameter `icm/HTTP/logging_n` is set (or removed) automatically when the administrator uses the *XS Admin Tools* to enable (or disable) HTTP tracing on the SAP Web Dispatcher for an application.

Related Information

Default Blacklisted System Properties in Tenant Databases [page 101]

SAP Note 1849775

14.1.4 Maintaining Application Runtime Configurations

Application runtime configurations specify the security measures that are implemented for access to applications.

The *SAP HANA XS Administration Tool* includes the *XS APPLICATIONS* tool, which you can use to create and maintain runtime configurations for individual applications or a complete application hierarchy. The configuration defined for an application is inherited by any application further down the application package hierarchy. A runtime configuration takes precedence over any runtime configuration located in an application package above it in the package hierarchy.

> ⓘ Note
>
> SAP HANA uses roles to grant access to the features provided by the *SAP HANA XS Administration Tool*. To access the tools required to configure SAP HANA XS runtime configurations, you must have a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator` assigned.

You can maintain the following aspects of the application runtime configuration:

> → Tip
>
> Runtime configuration settings override any settings in the application's corresponding application-access (`.xsaccess`) configuration file.

- Application security
  Enable/Disable user-authentication checks when starting an application
- User authentication methods (SAML, SPNego, X509, logon tickets, ...)
  Enable one or more authentication methods that applications use to authenticate user requests for content.
- Cross Origin Request Sharing (CORS)
  Enable support for cross-origin requests, for example, by allowing the modification of the request header. Allowing the sharing of cross-origin resources permits Web pages to make HTTP requests to another domain, where normally such requests would automatically be refused by the Web browser's security policy.
- Custom Headers
  Enable support for the X-Frame-Options HTTP header field, which allows the server to instruct the client browser whether or not to display transmitted content in frames that are part of other Web pages. You can also enable this setting in the application's corresponding `.xsaccess` configuration file.
- SQL connection configurations (SQLCC)
  Edit the details of an SQL connection configuration, which you use to enable the execution of SQL statements from inside your server-side JavaScript application with credentials that are different to the credentials of the requesting user
- HTTP destination configurations
  Edit the details of an HTTP destination configuration, which you use to defines connection details for services running on a specific host, whose details you want to define and distribute
- XS Job Schedules

Edit the details of an XS Job, for example to set the user account under which the job runs, define a start or stop time, and browse the job's log files.

Related Information

Create an Application Runtime Configuration [page 1147]
Edit and SQL Connection Configuration [page 1151]
Edit an HTTP Destination [page 1154]
Maintain XS Job Details [page 1228]

14.1.4.1 Create an Application Runtime Configuration

For SAP HANA XS applications, the runtime configuration defines the security and authentication settings to use when granting access to an application of the content it exposes.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. For example, to access the tools required to perform any tasks relating to application runtime configuration, you must have a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator`.

Context

The runtime configuration for an SAP HANA XS application specifies the settings the application uses when it is launched, for example, in response to a user request. If the same settings you define in a runtime configuration are also defined in a design-time file but with a different value, the runtime configuration takes precedence. To create a runtime configuration for an SAP HANA XS application, perform the following steps:

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who

logs on must have the privileges required to perform administration tasks with the *XS Administration Tools*.

2. Start the SAP HANA *XS Artifact Administration* tool.
   In the *Runtime Configuration Details* page you can maintain details of the runtime configurations for the various applications in your package hierarchy.
3. Define the runtime configuration for your application.

   > ⓘ Note
   >
   > The runtime configuration you define is inherited by all sub-packages in the package hierarchy.

   a. Configure the application security:

      Choose the *Edit* button in the *Security & Authentication* tab to can configure the following security options:

      - *Public (no authentication required)*
        Enable/Disable authentication for application requests.

        > → Tip
        >
        > If you **disable** authentication in the *Security and Authentication* panel, the *Authentication Types* options (for example, SAML or logon tickets) are hidden.

      - *Force SSL*
        Enable the force SSL option if you want the application to refuse browser requests that do not use secure HTTP (SSL/HTTPS) for client connections.

        > ⓘ Note
        >
        > The setting for this runtime option overrides the design-time setting for the *force_ssl* keyword in the application's `.xsaccess` file.

      - *Prevent Public Access for Sub-Packages*
        Ensure that public access only applies to the current package; all subpackages are hidden.

        > ⓘ Note
        >
        > This option is not available for packages shipped with SAP HANA.

   b. Configure the methods the applications must use to authenticate users.

      The *Authentication Types* list is only visible if the *Public (no authentication required)* option is disabled.

      > ⓘ Note
      >
      > You can select multiple authentication methods which are used in a specific order of priority, for example: first SAML, then logon tickets, and if the user-logon fails for both methods, then basic logon is offered.

      To ensure that, during the authentication process, the password in transmitted in encrypted form, it is strongly recommended to enable SSL/HTTPS for all application connections to the XS engine.

   c. Enable support for cross-origin request sharing (CORS), if required.

      The *CORS* tab enables you to allow the sharing of cross-origin resources; this permits Web pages to make HTTP requests to another domain, where normally such requests would automatically be refused by the Web browser's security policy.

d. Enable support for custom headers, if required.

   Use the *Custom Headers* tab to configure support for custom headers in the response. This feature enables you to set X-Frame options that allow frames in a Web page to display content from another Web site.

   Check the option *Enable Custom Headers* and choose the one of the entries in the list of *X-Frame Options*, for example:

   - `DENY`
   - `SAMEORIGIN`
   - `ALLOW-FROM <URL>`
     You can only specify one URL with the `ALLOW-FROM` option, for example: `"value":"ALLOW-FROM http://www.site.com"`.

     > **Note**
     >
     > To allow an application to use custom headers, you must enable the *Custom Headers* option.

4. Save the runtime configuration.

   > **Note**
   >
   > Use the *Reset* button to reset the runtime configuration to its previous state; use the *Revert* button to undo changes to the runtime-configuration options in the current tab.

Related Information

Application Runtime Configuration Details [page 1149]
Configure HTTPS (SSL) for Client Application Access [page 1184]

14.1.4.1.1 Application Runtime Configuration Details

In the *XS Artifact Administration* tool, the *Runtime Configuration Details* tab displays information about runtime settings configured for the currently selected application or artifact. You can use the *Runtime Configuration Details* tab to maintain the following details of the runtime configuration:

- Security & Authentication [page 1149]
- CORS [page 1150]
- Custom Headers [page 1151]

Security & Authentication

The *Security & Authentication* tab in the *Runtime Configuration Details* tool enables you to view details of the security settings defined to control access to an application service running on SAP HANA, for example, the

type of access allowed (user/public) and the method used to authenticate users. The following table indicates which information can be defined.

Security and Authentication Details

UI Element	Description	Example
Authentication Type	Enables/Disables requirement for user authentication to access an application service. If you **enable** authentication, you must select the methods that the application applies to authenticate users, for example, SAML or logon tickets.	Public (No Authentication Required)
Connection Security	Allows only secure HTTPS access to an application; insecure standard HTTP requests are refused. To ensure that passwords are transmitted in encrypted form during the authentication process, it is strongly recommended to enable SSL/HTTPS for all application connections to the XS engine. If you set the *force_ssl* option, you must ensure that the SAP Web Dispatcher is configured to accept and manage HTTPS requests.	SSL Enforced
Public Access for Sub-Packages	Enables public access to sub packages in an application package hierarchy. This setting cannot be changed for packages shipped with SAP HANA.	Allowed
Authentication Methods	Defines one of more methods that the application service uses to authenticate users requesting access. If multiple methods are selected, an order of priority applies: from most to least secure, for example, *SAML*, *Form Based*, and then *Basic*.	SAML, X509
SAML Identity Provider	The name of the SAML IDP used to verifty SAML certificates; this setting is only required if SAML is chosen as one of the authentication methods. *Not Applicable* indicates that no SAML IDP is configured.	SAMLIDP1

CORS

The *CORS* tab in the *Runtime Configuration Details* tool enables you to view details of the settings defined to control access to your application resource from other Web browsers. For example, you can specify where requests can originate from or what is allowed in the request and response headers. The following table indicates which information can be defined for Cross Origin Resource Sharing.

CORS Settings

CORS Option	Description
Cross Origin Resource Sharing	Enable/Disable requests from other browser sessions to an application.
ALLOWED ORIGINS	A single host name or a comma-separated list of host names that are allowed by the server, for example: `www.sap.com` or `*.sap.com`. If no host is specified, the default `"*"` (all) applies. Note that matching is case-sensitive.
ALLOWED HEADERS	A single header or a comma-separated list of **request** headers that are allowed by the server. If no request header is specified, no default value is supplied.

CORS Option	Description
EXPOSED HEADERS	A single header or a comma-separated list of **response** headers that are allowed to be exposed. If no response header is specified for exposure, no default value is supplied.
ALLOWED METHODS	A single permitted method or a comma-separated list of methods that are allowed by the server, for example, `“GET”, “POST”`. If no method is specified, the default `“GET”, “POST”, “HEAD”, “OPTIONS”` (all) applies. Note that matching is case-sensitive.
MAX AGE	A single value specifying how long a preflight request should be cached for. If no value is specified, the default time of `“3600”` (seconds) applies.

Custom Headers

The *Custom Headers* tab in the *Runtime Configuration Details* tool enables you to configure support for custom headers in the HTTP response. This feature enables you to set X-Frame options that allow frames in a Web page to display content from another Web site.

Custom Headers Details

UI Element	Description	Example
Custom Headers	Enable/Disable the use of custom headers in HTTP response.	Disabled
X-Frame Options	Allow/Deny requests to display content from the same or another Web site. Note that you can only specify **one** URL with the `ALLOW-FROM` option, for example: `"value":"ALLOW-FROM http://www.site.com"`.	DENY, SAMEORIGIN, ALLOW-FROM

Related Information

Create an Application Runtime Configuration [page 1147]

14.1.4.2 Edit an SQL Connection Configuration

In SAP HANA Extended Application Services (SAP HANA XS), you use the SQL connection configuration to enable the execution of SQL statements from inside your server-side JavaScript application with credentials that are different to the credentials of the requesting user.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. For example, to access the tools required to perform any tasks

relating to SQL connection configuration (SQLCC), you must have a role based on the role template `sap.hana.xs.admin.roles::SQLCCAdministrator`. This role includes the related role `sap.hana.xs.admin.roles::SQLCCViewer`

Context

The SQL connection configuration enables the execution of SQL statements from inside your server-side JavaScript application with credentials that are different to the credentials of the requesting user. You can use the *XS Artifact Administration* tool to change the user name in the XS SQL connection-configuration file.

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *XS Artifact Administration* tool.

2. Start the SAP HANA *XS Artifact Administration* tool.
   In the *XS Artifact Administration* tool you can manage the runtime configurations for the various applications in your package hierarchy.
3. Locate the SQL connection configuration object.

   In the *Application Objects* list, locate and select the object containing the SQL connection configuration that you want to edit; SQL connection configuration objects have the file extension `.xssqlcc`. The details are displayed in the *SQL Connection Details* panel.
4. Maintain the SQL connection details.

   The *SQL Connection Details* allows you to modify details of the database user whose credentials are used to establish the SQL connection defined in the SQLCC object. If a role is specified in the `role_for_auto_user` parameter, SAP HANA assigns the role defined in `role_for_auto_user` to the new auto-generated user.

   > **Note**
   >
   > If you bind the XS SQL connection to a specific existing database user (not the auto user), you must provide the user's password. If do not provide a password for the specified database user, you cannot save the changes to the SQLCC object's runtime configuration.

5. Set the run-time status of the XS SQL connection configuration.

   You must set the rutime status of the XS SQL connection configuration to *Active*; the run-time status can only be changed by an administrator. When the run-time status of the XSSQL connection configuration

is set to *active*, SAP HANA automatically generates a new user (`XSSQLCC_AUTO_USER_[...]`) for the XSSQL connection configuration object and assigns the role defined in `role_for_auto_user` to the new auto-generated user.

6. Save the changes.

Related Information

SQL Connection Details [page 1153]

14.1.4.2.1 SQL Connection Details

The SQL-connection configuration file specifies the details of a connection to the database.

The database connection established by the SQL-connection configuration file enables the execution of SQL statements with credentials that are different to the credentials of the requesting user, for example, from inside a server-side (XS) JavaScript application.

The *SQL Connection Details* tab in the *XS Artifact Administration* tool enables you to view details of the XS SQL connection configurations that you have defined, for example, the package location, and the user bound to the SQL connection. The following table indicates which information can be viewed.

SQL Connection Details

UI Element	Description	Example
Package	The name of the repository package containing the currently selected SQL connection configuration	testApp
Description	A short description of the selected SQL connection configuration	Admin SQL connection
Username	The name of the user to whom you want to bind the SQL connection configuration. If no user is specified, SAP HANA automatically generates the user `XSSQLCC_AUTO_USER_[...]` when the run-time status of the XSSQL connection configuration is set to *Active*. The new auto-user is assigned the role specified in *Role for Auto User*. If you bind the SQL connection manually to a specific SAP HANA user, you must supply the user's password to enable a connection to be established **and** ensure that the user has the necessary privileges (for example, by assigning a role).	XSSQLCC_AUTO_USER_[...]
Password	The password for the user bound to the SQL connection configuration. A password is not required for the automatically generated `XSSQLCC_AUTO_USER_[...]`.	*****
Assigned by	The name of the user who assigned the user defined in *Username* to the currently selected SQL connection configuration	JohnDoe

UI Element	Description	Example
Role for Auto User	The name of (and package path to) the role to be assigned to the new auto-user that is generated when the run-time status of the XSSQL connection configuration is set to *active*	acme.com.xs.roles::JobAdministrator
Status	The current runtime status of the XSSQL connection configuration (active/inactive)	active

Related Information

Edit an SQL Connection Configuration [page 1151]

14.1.4.3 Edit an HTTP Destination Runtime Configuration

An HTTP destination defines connection details for services running on a specific host, whose details you want to define and distribute. The HTTP destination can be referenced by an application.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. For example, to access the tools required to perform any tasks relating to HTTP destination configuration (HTTPDest), you must have a role based on the role template `sap.hana.xs.admin.roles::HTTPDestAdministrator`.

Context

To edit an HTTP destination using the *SAP HANA XS Administration Tool*, perform the following steps:

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user

who logs on must have the privileges required to perform administration tasks with the *XS Artifact Administration* tool.

2. Start the SAP HANA *XS Artifact Administration* tool.
   Use the *XS Artifact Administration* tool to manage the runtime configurations for the various applications in your package hierarchy.
3. Locate the HTTP destination configuration object that you want to edit.

   In the *Application Objects* list, locate and select the object containing the HTTP destination configuration that you want to edit. HTTP destination configuration objects have the suffix `.xshttpdest`. The details are displayed in the *HTTP Destination Details* panel.
4. Edit the details of the HTTP destination configuration.

   To edit the details of an HTTP destination configuration, choose the *Edit* button in the screen displaying the details you want to edit, for example:

   - *General Information*
     Host name and port of the server to connect to, any path prefix (to add to the start of the URL used to connect to the service on the remote host, and a timeout setting for the time allowed to connect to the remote host.

     > **Note**
     >
     > The *Extends* option is only available if the HTTP destination you are modifying is being used to extend the configuration defined in another HTTP destination.

   - *Proxy Details*
     Details of the proxy type (*None*, *HTTP*, or *Socks*), the name of the system hosting the proxy service, the port to connect on and the user credentials required to establish the connection.

     > **Caution**
     >
     > The proxy-server settings you define here are overriden by any SAP HANA system wide setting for a proxy server, for example, defined by the `enforced_outbound_proxy` parameter in the `communication` section of the `xsengine.ini` configuration file.

   - *Authentication Details*
     - *SSL Enabled*
       SSL for outbound connections between SAP HANA XS and the host named in the HTTP destination configuration.
       You must choose an *SSL Authentication Type*. If you choose *Client Certificate* (default), you must specify the *Trust Store* where the certificates are stored. You can choose an existing trust store from a list of stores configured for the SAP HANA instance (in the *Trust Store* drop-down menu), or create a new trust store using the *Trust Manager*.
       SSL Host Check (`true | false`) enables a check which verifies that the certificate used for authentication is valid (matches the host). If the certificate does not match, SSL terminates.
     - *Authentication Type* (for example, *none*, *basic*, *SAP Assertion Ticket*, *SAML*, or *SAML Assertion Propagation*).

       > **Note**
       >
       > The *Authentication Type* you select determines what (and how much) additional information is required.

For example, for the *SAP Assertion Ticket* authentication type, you must provide the SAP SID and client number of the instance providing the service. The value displayed (if any) is the one already defined in the design-time representation of the HTTP destination configuration. Any changes you make to the runtime configuration (here) are synchronized with the design-time configuration artifact.
For *SAML*, the values displayed reflect the parameters set in the corresponding design-time representation of the HTTP destination configuration, for example, `ConfigFileName.xshttpdest`. For more information, see *HTTP Destination Details* in *Related Information*.

- *OAuth Details*
  You cannot enter this information manually; the information is read from the design-time configuration file that describes the OAuth application, for example, `oauthDriveApp.xsoauthappconfig`.
  To display a list of available OAuth application-configuration packages (files with the suffix `*.xsoauthappconfig`) on your SAP HANA system, choose *Browse OAuth App Configs* and select a package from the list. The location of the package containing the OAuth application-configuration you choose is used to populate the *OAuth App Config Package* field; the name of the OAuth application-configuration you choose is used to populate the *OAuth App Config Name* field.

5. Save the changes.

   Saving the changes to the HTTP destination configuration automatically commits the HTTP destination configuration object to the SAP HANA repository and activates it.

> → Tip
>
> Use the *Reset* button to reset the runtime configuration to its previous state; use the *Revert* button to undo changes to the runtime-configuration options in the current tab.

Related Information

HTTP Destination Details [page 1156]
SAP HANA XS Classic Configuration Parameters [page 1140]

14.1.4.3.1 HTTP Destination Details

An HTTP destination defines connection details for services running on a specific host, whose details you want to define and distribute

In the *XS Artifact Administration* tool, the *HTTP Destination Details* tab displays information about the currently selected HTTP destination. You can use the *HTTP Destination Details* tab to maintain the following details of the runtime configuration:

- General Information [page 1157]
- Proxy Details [page 1157]
- Authentication Details [page 1158]
- OAuth Details [page 1159]

General Information

The *General Information* tab in the *HTTP Destination Details* tool enables you to view details of the HTTP destination that you have defined, for example, the name of the destination host, the port to connect on, and a short description. The following table indicates which information can be viewed.

HTTP Destination Details

UI Element	Description	Example
Extends	The name of another HTTP destination configuration which the currently selected configuration is using as a base but also modifying.	gfn.xshttpdest
Description	A short description of the selected HTTP destination	Service @ Destination
Host	The name of the system hosting the services defined in the HTTP destination configuration	download.finance.acme.com
Port	The port to connect to on the remote host	80
Path Prefix	The prefix to add to the start of the URL used to connect to the service on the remote host	/d/quotes.csv?f=a
Timeout	The time allowed to connect to the remote host defined in the HTTP destination	0

Proxy Details

The *Proxy Details* tab in the *HTTP Destination Details* tool enables you to view details of the proxy service used by the HTTP destination that you have defined, for example, the name of the proxy host, the port to connect on, and the user credentials required to establish a connection. The following table indicates which information can be viewed and configured.

Proxy Server Details

UI Element	Description	Example
Proxy Type	The type of proxy service, for example: None, HTTP, or SOCKS.	HTTP
Proxy Host	The name of the system hosting the proxy service used by the HTTP destination	proxy.host.acme.com
Proxy Port	The port to connect to on the system hosting the proxy service	8080
Proxy User	The user credentials required to connect to the proxy service	johndoe

> **Note**
>
> The proxy-server settings you define here are overridden by any SAP HANA system-wide setting for a proxy server, for example, defined by the `enforced_outbound_proxy` parameter in the `communication` section of the `xsengine.ini` configuration file.

Authentication Details

The *Authentication Details* tab in the *HTTP Destination Details* tool enables you to view details of the authentication service used by the HTTP destination that you have defined, for example, the authentication **type** and the trust store used to maintain any SSL client certificates. The following table indicates which information can be viewed and modified.

Authentication Details

UI Element	Description	Example
Authentication Type	The type of service used for authentication, for example: *None*, *Basic*, *SAP Assertion Ticket*, *SAML*, or *SAML Assertion Propagation*	*SAML*
Communication Security	Enable or disable SSL communication. If you enable SSL, you must select an *SSL Authentication Type*.	*SSL Enabled*
SSL Authentication Type	The type of authentication used for SSL, for example, *Client Certificate* (default) or *Anonymous*. If you choose *Client Certificate*, you must specify the trust store where the certificates are located.	*Client Certificate*
SSL Host Check	Enable or disable the SSL host check; the check verifies that the certificate used for authentication is valid (matches the host).	Enabled
Trust Store	The name of the trust store used to maintain security certificates required during the authentication process; select from the drop-down list	SAPLogon

The following table lists the choices available when configuring the authentication type for an HTTP destination.

HTTP Destination: Authentication Type

UI Element	Description	Example
None	No user authentication is performed	-
Basic	The *Name* of the user whose account is used to log on to the HTTP destination using basic authentication	JohnDoe
	The *password* of the user specified in *Name*	*****
SAP Assertion Ticket	System ID (*SAP SID*) of the SAP instance providing the SAP Assertion Ticket service	GFN
	Client number of the SAP instance providing the SAP Assertion Ticket service	007
SAML	The *Entity ID* of the remote SAML party	accounts.acme.com
	User Mapping: a list of name-ID mappings, for example, *Unspecified*, *Email*, *Email, Unspecified*	Email
	Assertion Consumer Service defines the way in which SAML assertions and responses are sent, for example: as an authorization header or POST parameter.	Assertion as POST parameter
	Additional *Attributes* for the SAML Assertion.	Email

UI Element	Description	Example
SAML Assertion Propagation	Allow an SAML token to be forwarded from the server where the token was received to another server.	N/A

For the authentication type *SAML*, the values displayed reflect the parameters set in the corresponding design-time representation of the HTTP destination configuration, as illustrated in the following table:

HTTP Destination SAML Runtime:Design-Time Parameters

SAML Runtime Parameter	SAML Design-Time Parameter	Description
Entity ID	`samlProvider`	The entity ID of the remote SAML party
Assertion Consumer Service	`samlACS`	The way in which SAML assertions or responses are sent
Attributes	`samlAttributes`	Additional attributes for the SAML Assertion.
User Mapping	`samlNameId`	A list of name-ID mappings, for example, *e-mail*.

OAuth Details

The *OAuth Details* tab in the *HTTP Destination Details* tool enables you to view details of the OAuth package used by the HTTP destination that you have defined. An OAuth configuration package is a collection of configuration files that define the details of how an application uses OAuth to enable logon to a resource running on a remote HTTP destination. The following table indicates the information that can be viewed.

HTTP Destination: OAuth Information

UI Element	Description	Example
OAuth App Config Package	The name of the repository package containing the OAuth application-configuration	sap.hana.xs.oAuth.lib.providerconfig
OAuth App Config Name	The name of the OAuth application-configuration (repository artifacts with the suffix `.xsoauthappconfig`)	abap.xsoauthappconfig

> **Note**
>
> You cannot enter this information manually; the information is read from the design-time configuration file that describes the OAuth application, for example, `oauthDriveApp.xsoauthappconfig`.

Related Information

Edit an HTTP Destination Runtime Configuration [page 1154]

14.1.5 Managing Trust Relationships

Trust relationships enable you to establish secure connections between known servers whose identity can be confirmed by a signed certificate. The certificates are stored in a trust store.

The *SAP HANA XS Administration Tool* includes the *Trust Manager*, which is an application that you can use to create and maintain the certificates used to establish trust relationships between servers. You can use the *Trust Manager* to perform the following tasks.

> ⓘ Note
>
> SAP HANA uses roles to grant access to the features provided by the *SAP HANA XS Administration Tool*. To access the tools required to maintain trust relationships between SAP HANA and other systems, you must have a role based on the role template `sap.hana.xs.admin.roles::TrustStoreAdministrator`.

- Add/Delete a trust store
  SAP HANA makes use of multiple trust stores.

  > ⓘ Note
  >
  > The trust stores listed below are located in the **file system**. In some cases, it is possible and recommended to use trust stores that exist in the database as database objects. In-database trust stores (referred to as certificate collections) contain the required client certificates, which are also stored in the database. We recommend using in-database certificate collections where possible. For more information, see *Managing Client Certificates in the SAP HANA Database*.

  - The SAP HANA trust store (`sapsrv.pse`)
    Used for secure SQL and SAML or OAuth scenario, `sapsrv.pse`. This trust store is not created automatically. You must generate it manually (for example using the SAP Web Dispatcher administration tool or the SAPGENPSE tool) and store it in the $SECUDIR directory.

    > → Recommendation
    >
    > For user authentication based on X.509 certificates and SAML assertions, we recommend creating separate certificate collections with the purposes *X.509* and *SAML* instead of using the file system-based trust store `sapsrv.pse`.

  - The SAP Web Dispatcher trust store (`SAPSSLS.pse`)
    Required for SSL connections using the Secure Socket Layer, `SAPSSLS.pse` is installed automatically and is available by default.
  - The SAP Logon Ticket trust store (`saplogon.pse`)
    Optional: `saplogon.pse` is only necessary if an SAP HANA XS application requires an SAP logon ticket from a user at logon

    > → Recommendation
    >
    > For user authentication based on logon tickets, we recommend creating a certificate collection with the purpose *SAP LOGON* instead of using the file system-based trust store `saplogon.pse`.

  - The client authentication trust store (`SAPSSLC.pse`)
    Optional: `SAPSSLC.pse` is only required for client connections, for example, that use the SQL client interface (`hdbsql`).

- Manage your own certificates
  - Import a private key
  - Create a certificate request
  - Have the requested certificate signed by a certificate authority
  - Import the signed certificate into the trust store
- Manage server certificates
  - HTTP destinations (via SSL/HTTPS)
  - Certificate authorities (for example, “Verisign” or “TC TrustCenter Universal”)

The *Trust Manager* tool enables you to configure the out-bound view; that is, trust relationships with remote systems that provide services required by SAP HANA XS applications. If you want to configure the **in-bound** view (for example, incoming requests **to** SAP HANA), use the SAP HANA *Web Dispatcher Administration* tool.

- Out-bound trust
  Secure communication and trust for out-bound communication, for example, between an SAP HANA XS application and and an ABAP system using using SSL/TLS.
- In-bound trust
  Secure communication and trust for in-bound communication, for example, between an SAP HANA XS application and and an ABAP system using using SSL/TLS.

Both the *Trust Manager* and the *Web Dispatcher Administration* tools are available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/wdisp/admin`.

> **Note**
>
> Access to the *Web Dispatcher Administration* tool is enabled by the role *sap.hana.xs.wdisp.admin::WebDispatcherAdmin*.

Related Information

Add/Edit a Trust Store [page 1162]
Import a Server Certificate [page 1165]
Create Your Own Certificate [page 1163]

14.1.5.1 Add/Edit a Trust Store

The trust store enables you to maintain a list of servers that you trust; the trust is based on a certificate you import into the trust store and which can be signed by a certificate authority, for example, Verisign or TCTrustCenter.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS Administration Tool. To access the tools required to add a trust store, you must have a role based on the role template `sap.hana.xs.admin.roles::TrustStoreAdministrator`.

Context

> → Recommendation
>
> This procedure describes how to create a trust store in the file system. We recommend creating trust stores in the database (referred to as certificate stores) where possible. For more information, see the section *Managing Client Certificates in the SAP HANA Database*.

To enter the details of trust store, you can use the *SAP HANA XS Administration Tool*, as described in the following steps.

> ⚠ Caution
>
> To maintain the details of a trust store, you must be familiar with the concepts of trust stores and the certificates they contain.

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must also have the privileges required to perform the administration tasks associated with trust stores.

2. Start the *Trust Manager*.

   The *Trust Manager* is available in the list of SAP HANA XS administration tools.

3. Create the new trust store.

   In the *Create Trust Store* dialog, you must provide a name for the new trust store.

   a. In the *Trust Store* pane, choose *Add* to open the *Create Trust Store* dialog.
   b. Type a name for the new trust store and choose *OK*.
      Choose *OK* to add the trust store to the list of trust stores known to SAP HANA XS.
4. Define the details of the new trust store.

   You can use the *Own Certificate* and *Certificate List* to manage the certificates you import for the servers that are known to and trusted by SAP HANA XS.

Related Information

Import a Server Certificate [page 1165]
Create Your Own Certificate [page 1163]

14.1.5.2 Create Your Own Certificate

The trust store enables you to maintain a list of servers that you trust; the trust is based on a certificate you import into the trust store and which can be signed by a certificate authority, for example, Verisign or TCTrustCenter.

Prerequisites

> **Note**
>
> This feature is available with restricted releases. If you want to use it, refer to SAP Note 1779803. See the Related Information section for the direct link.

SAP HANA uses roles to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. To access the tools required to perform trust manager tasks, you must have a role based on the role template `sap.hana.xs.admin.roles::TrustStoreAdministrator`.

Context

You can use the certificates stored in the trust store to secure the communication between trusted servers, for example, with SSL/HTTPS. However, you must also create a certificate that you can use to authenticate the identity of the SAP HANA server, too.

To create your own certificate and import it into your trust store, perform the following steps:

Procedure

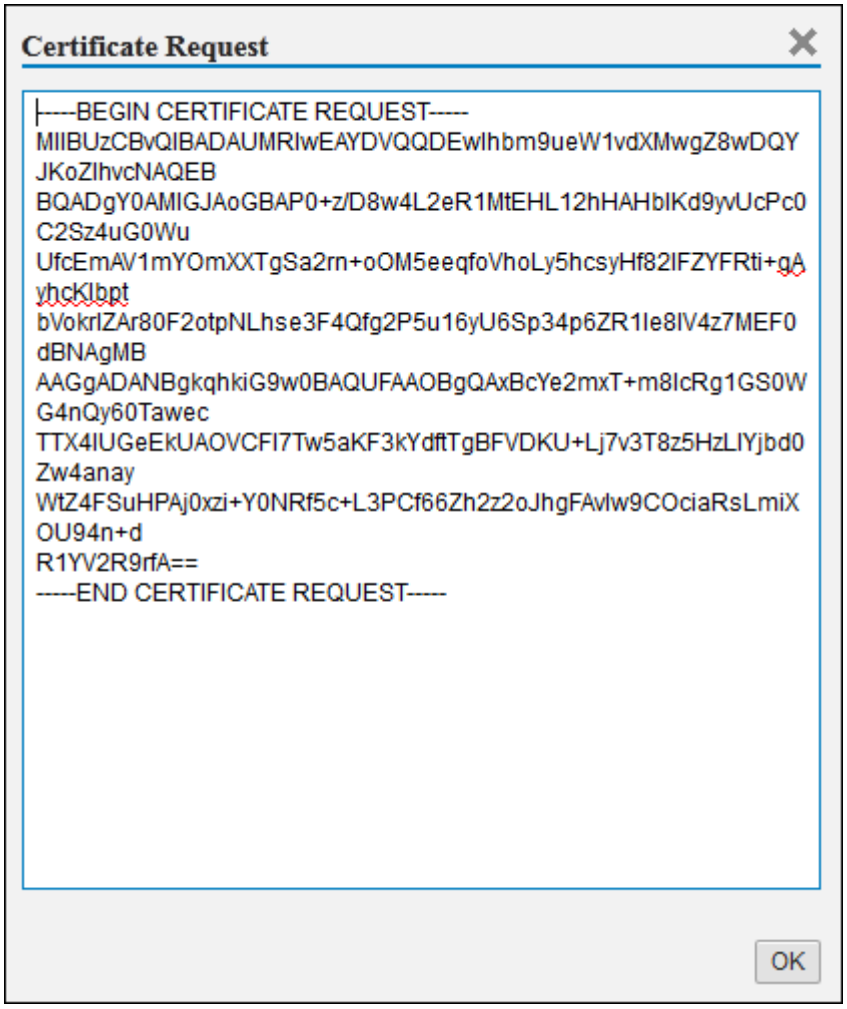
1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *Trust Manager* tool.

2. Start the SAP HANA XS *Trust Manager* tool.
   In the list of tools, choose *Trust Manager* tab to display the screen where you can manage the certificates in your trust store.
3. Create a certificate request.

   In the *Own Certificate* panel, choose *Certificate Actions* > *Create CA Request*.



4. Send the certificate request to a certificate authority for signing.

   You must send the certificate request to a certificate authority (CA) to have it signed; you import the response from the CA into your trust store.

5. Import the signed certificate into the trust store.

   This may be a trust store in the file system (for example, `sapsrv.pse`) or an in-memory certificate collection with the purpose *SAML* (recommended).

Option	Description
Certificate collection with purpose *SAML* (recommended)	Use the SAP HANA cockpit to import the certificate into the certificate store and then add it to the relevant collection. For more information, see the section on managing certificates.
Trust store in the file system	In the *Own Certificate* panel, choose *Certificate Actions* > *Put CA Response*. The imported certificate is displayed in the certificate list.

Related Information

SAP Note 1779803
Add/Edit a Trust Store [page 1162]

14.1.5.3 Import a Server Certificate

A server certificates enables you to establish a trusted relationship between SAP HANA and the server described in the server certificate. You import the certificates into the trust store.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. To access the tools required to perform trust manager tasks, you must have a role based on the role template `sap.hana.xs.admin.roles::TrustStoreAdministrator`.

Context

> → Recommendation
>
> This procedure describes how to import a server certificate into a trust store in the file system. We recommend creating trust stores in the database (referred to as certificate stores) where possible. For more information about how to import certificates into the in-memory certificate store and add them to certificate collections, see the section *Managing Client Certificates in the SAP HANA Database*.

The trust store enables you to maintain a list of servers that you trust; the trust is based on a certificate you import into the trust store and which can be signed by a certificate authority, for example, Verisign or TCTrustCenter. You can use the certificates to secure the communication between the trusted servers, for example, with SSL/HTTPS.

To import a certificate into your trust store, perform the following steps.

Procedure

1. Obtain a copy of the certificate you want to import into your trust store.

   You can export a certificate from a server and save it to a temporary location.
2. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ### Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *Trust Manager* tool.
3. Start the SAP HANA XS *Trust Manager* tool.
   In the list of tools, choose the *Trust Manager* tab to display the screen where you can manage the certificates in your trust store.
4. Locate the copy of the certificate you want to import into the trust store.

   In the *Certificate List* panel, choose *Import Certificate* > *Browse...* and navigate to the folder containing the certificate you want to import.

   > ### Note
   >
   > Trust certificates usually have a recognizable suffix such as `.crt`, for example, `TCTrustCenterUniversalCAIII.crt`.
5. Import the certificate into the trust store.

   In the *Import Certificate* dialog, choose *Import Certificate*.

   > ### Note
   >
   > If you are importing a certificate you created yourself, you must provide a password to complete the import operation.

   The imported certificate is displayed in the certificate list.

Related Information

Add/Edit a Trust Store [page 1162]

14.1.6 Maintaining SAML Providers

You can configure an SAP HANA system to act as a service provider for Single Sign On (SSO) authentication based on Security Assertion Markup Language (SAML) certificates.

The *SAP HANA XS Administration Tool* includes the *SAML CONFIGURATION* application, which you can use to configure SAP HANA system to act as an SAML service provider for SSO authentication. You must perform this step if you want your SAP HANA XS applications to use SAML as the logon authentication method, for example, by enabling the *SAML* option in the *AUTHENTICATION* panel in the *XS APPLICATIONS* tool

> ⓘ Note
>
> SAP HANA uses roles to grant access to the features provided by the *SAP HANA XS Administration Tool*. To access the tools required to configure an SAP HANA system to act as an SAML service provider, you must have a role based on the role template `sap.hana.xs.admin.roles::SAMLAdministrator`.

You can use the *SAML CONFIGURATION* tool to perform the following tasks:

- Configure an SAP HANA system to act as a service provider
- Add a new SAML Identity provider (IDP)
- Modify the details of an existing SAML Identity provider (IDP)

> ⓘ Note
>
> To maintain a SAML identity provider (IDP), you must be logged on to SAP HANA with the credentials of the system user.

> ⓘ Note
>
> If you want your SAP HANA system to validate audience restrictions in SAML assertions, you need to set the `[authentication] saml_service_provider_name` configuration parameter. If the parameter is not set, an SAP HANA system ignores a possible audience restriction tag in a SAML assertion. By default, in SAP HANA tenant databases the parameter is set to the tenant name and the instance ID, for example, 'TENANT01'. For the system database, the parameter is empty by default.

Related Information

Configure an SAP HANA System as an SAML Service Provider [page 1168]
Add a SAML Identity Provider in SAP HANA Studio

14.1.6.1 Configure an SAP HANA System as an SAML Service Provider

SAP HANA supports the use of authentication based on Security Assertion Markup Language (SAML) certificates.

Prerequisites

SAP HANA user roles are used to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. To access the tools required to configure an SAP HANA system to act as an SAML service provider, you must have a role based on the role template `sap.hana.xs.admin.roles::SAMLAdministrator`.

Context

You can configure an SAP HANA system to act as a service provider for authentication based on Security Assertion Markup Language (SAML) certificates. You must perform this step if you want your the SAP HANA XS applications to use SAML as the user authentication method.

> ⚠ Caution
>
> To maintain the details of SAML service providers, you must be familiar with the technical background of SAML SSO mechanisms and requirements.

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform SAML administration tasks.

2. Start the SAP HANA XS *SAML Service Provider* tool.
   In the list of tools, choose *SAML Service Provider* to display the screen where you can enter details of the SAML service provider you want to configure.
3. Enter details of the SAML service provider.

   In the *Service Provider Information* panel choose *Edit*; you must provide the following information:

> **Note**
>
> The information you enter is used to populate the XML document saved as the SAML service-provider metadata.

- *Name*
  This can be any name but, for troubleshooting purposes, is usually the fully qualified name of the system hosting the SAML service.
- *Organization Name*
  According to the oasis SAML standard, the name of the organization responsible for the SAML service described here. The name you enter here is wrapped in the XML tag `<OrganizationName>` used in the SAML certificate. The organization name can (but does not have to) be human readable.
- *Organization Display Name*
  According to the oasis SAML standard, the human-readable form of the name of the organization responsible for the SAML service described here. The name you enter here is wrapped in the XML tag `<OrganizationDislayName>` that is contained in the SAML certificate.
- *Organization URL*
  A URL that specifies a location where a user can find additional information about the organization responsible for the SAML service you describe in this task.

The information you enter in the various configuration tabs and screens is added to the appropriate tags in the XML document displayed in the *Metadata* tab.

4. Save the SAML service-provider configuration.

   Choose *Save*; the XML document describing the SAML service is parsed and, if no errors are found, saved.

Related Information

SAML Service Provider Details [page 1169]

14.1.6.1.1 SAML Service Provider Details

An SAP HANA system can act as an SAML service provider for SSO authentication.

An SAP HANA system can act as a service provider for authentication based on Security Assertion Markup Language (SAML) certificates. The *SAML Service Provider* tool displays the following screens to help you maintain details of the SAML service provider:

- Service Provider Information [page 1170]
- Service Provider Configuration [page 1170]
- Metadata [page 1170]

> **Note**
>
> The information you enter is used to populate the XML document saved as the SAML service-provider metadata.

Service Provider Information

The *Service Provider Information* tab in the *SAML Service Provider* tool enables you to provide details of the SAML service provider. The following table indicates which information is required.

UI Element	Description	Example
Name	The fully qualified name of the system hosting the SAML service	SAMLSP01
Organisation Name	The name of the organisation responsible for the SAML service provider. The name you enter is wrapped in the XML tag `<OrganizationName>` used in the SAML certificate. *Organization Name* can (but does not have to) be human readable	SAP
Organisation Display Name	The human-readable name of the organisation responsible for the SAML service provider. The name you enter here is wrapped in the XML tag `<OrganizationDislayName>` used in the SAML certificate.	SAP
Organisation URL	A location where a user can find additional information about the organization responsible for the SAML service	sap.com

Service Provider Configuration

The *Service Provider Configuration* tab in the *SAML Service Provider* tool enables you to maintain details of the SAML service provider used to handle SAML assertions. The following table indicates which information is required.

UI Element	Description	Example
Hash	The hash algorithm use to encode SAML assertions	SHA256
Add Key Info	If `<Keyinfo>` node should be included in the XML signature; default = yes	"yes" or "no"
Default Application Path	Path to the application requiring logon user credentials provided by the SAML service provider, if the SSO request is initiated by an SAML identity provider	/
Assertion Timeout	Period of time (in seconds) for which SAML assertion requests for SSO initiated by an SAML service provider remain valid; default=10 minutes	1000
Default Role	Default SAP HANA role assigned to new SAML users	JobViewer

Service Provider Metadata

The *Metadata* tab in the *SAML Service Provider* tool enables you to view details of the SAML service provider used to handle SAML assertions. The metadata document includes the information you enter in the *Service Provider Information* and *Service Provider Configuration* tabs.

Field Name	Description
Metadata	An XML file containing details of the SAML service provider used to handle SAML assertions

14.1.6.2 Add an SAML Identity Provider

SAP HANA supports the use of SSO authentication based on Security Assertion Markup Language (SAML) certificates. An identity provider is used by the service provider to authenticate the users signing in by means of SSO.

Prerequisites

- The SAP HANA trust store contains the server certificate that will be used to generate SAML SP metadata and validate SAML assertions (service provider certificate). We recommend that you use an in-memory certificate collection with purpose *SAML*. For more information, see the section on managing client certificates.
- SAP HANA user roles are used to determine the level of access to the features provided by the SAP HANA XS Administration Tool. To access the tools required to add an SAML identity provider (SAML IDP), you must have a role based on the role template `sap.hana.xs.admin.roles::SAMLAdministrator`.
- You need access to the XML document containing the IDP metadata that describes the SAML identity provider (SAML IDP) you want to add.

Context

To enter the details of SAML identity providers, you can use the *SAP HANA XS Administration Tool*, as described in the following steps:

> ⚠ Caution
>
> To maintain the details of an SAML identity provider, you must be familiar with the technical background of SAML SSO mechanisms and requirements.

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

> **Note**
>
> In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform SAML administration tasks.

2. Add an SAML SSO identity provider (IDP).

   The information required to maintain details of an SAML IDP is specified in an XML document containing the IDP metadata. This document should be available as part of the SAML service you want SAP HANA XS to use. The only information you must provide manually is the name of the new IDP; the IDP name must be unique.

   a. In the *SAML Identity Provider List*, choose *[+]* to display the *Add Identity Provider Info* pane.
   b. In the *Add Identity Provider Info* pane, paste the contents of the XML document containing the IDP metadata into the *Metadata* box.
      If the contents of the XML document are valid, the parsing process extracts the information required to insert into the *Subject*, *Entity ID*, and *Issuer* fields in the *General Data* screen area, and the URL fields in the *Destination* screen area, for example, *Base URL* and *SingleSignOn URL (*)*.
   c. In the *Name* box of the *General Data* screen area, enter a name for the new SAML SSO identity provider.

      > **Note**
      >
      > The name of the SAML IDP is mandatory and must be unique; it appears in the list of available SAML IDPs that is displayed, if you select SAML as the authentication method for SAP HANA XS applications to use, for example, in the *Authentication* screen area of the *XS Artifact Administration* tool.

3. Save the details of the new SAML identity provider.

   Choose *Save* to save the details of the SAML identity provider and add the new SAML IDP to the list of known SAML IDPs.

   The new SAML IDP is displayed in the list of known IDPs shown in the *SAML Identity Provider List*.

4. Check the details of the new SAML IDP.

   Select the new SAML IDP in the list of known SAML IDPs to display the IDP's details in the information panel.

Next Steps

Copy the certificate from the SAML IDP metadata document and add it to the SAP HANA trust store for SAML authentication (certificate collection with purpose *SAML*). For more information, see *Configure SSO with SAML Authentication for SAP HANA XS Applications*.

Related Information

Configure an SAP HANA System as an SAML Service Provider [page 1168]
Modify an Existing SAML Identity Provider [page 1174]

SAML Identity Provider Details [page 1173]
Configure SSO with SAML Authentication for SAP HANA XS Applications [page 1195]

14.1.6.2.1 SAML Identity Provider Details

An SAML identity provider is used by the SAML service provider to authenticate users signing in by means of a single sign-on (SSO) mechanism.

SAP HANA supports the use of SSO authentication based on Security Assertion Markup Language (SAML) certificates. An SAML identity provider is used by the SAML service provider to authenticate users who sign in to an application by means of SSO. As part of the SAML IDP configuration, you specify the following options:

- General data [page 1173]
- HTTP Destination [page 1173]

General Data

The *General Data* screen area in the *SAML Identity Provider* tool enables you to maintain details of the SAML identity provider. The following table indicates which information can be maintained.

General SAML IDP Details

UI Element	Description	Example
Name	The name of the SAML identity provider is mandatory and must be unique.	ACCOUNTS_ACME_COM
Subject	SAML IDP is specified in an XML document containing the IDP metadata	CN=CPS Production, OU=WebKm, O=ACME, L=Accra, C=GH
Issuer	SAML IDP is specified in an XML document containing the IDP metadata	CN=CPS Production, OU=WebKm, O=ACME, L=Accra, C=GH
Entity ID	The entity ID of the remote SAML party	accounts.acme.com
Dynamic User Creation	Enable or disable the dynamic creation of new SAML users.	Disabled

Destination

The *Destination* screen area in the *SAML Identity Provider* tool enables you to maintain details of the HTTP destination for the system hosting the SAML identity provider service. You must provide a base URL for the SAML IDP as well as further, more detailed, information about the location of the resources that provide the sign-on and sign-off services. The following table indicates which information can be maintained.

Details of the SAML IDP's HTTP Destination

UI Element	Description	Example
Base URL	The resource location where the SAML identity provider is reachable.	https://accounts.acme.com:443
SingleSignOn URL (RedirectBinding)	URL of the IDP endpoint for SSO requests using SAML redirect binding	/saml2/idp/sso/accounts.acme.com
SingleSignOn URL (PostBinding)	URL of the IDP endpoint for SSO requests using SAML post binding	/saml2/idp/sso/accounts.acme.com
SingleLogout URL (RedirectBinding)	URL of the IDP endpoint for single logout (SLO) requests using SAML redirect binding	/saml2/idp/slo/accounts.sap.com
SingleLogout URL (PostBinding)	URL of the IDP endpoint for single logout (SLO) requests using SAML post binding	/saml2/idp/slo/accounts.sap.com

SAML bindings describe a protocol used to transport SAML messages: both the requests and the responses. The following bindings are relevant for the configuration of the HTTP destination for the SAML identity provider.:

- Redirect binding
  The SAML message is in the URL itself as a query parameter. Redirect bindings enforce limitations on the message and ZLIB compression is required.
- Post binding
  The SAML message is transported inside an HTTP body in the `POST` parameter. There is no limitation on the message and no compression needed.

14.1.6.3 Modify an Existing SAML Identity Provider

SAP HANA supports the use of SSO authentication based on Security Assertion Markup Language (SAML) certificates. An identity provider is used by the service provider to authenticate the users signing in by means of SSO.

Prerequisites

- SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS Administration Tool. To access the tools required to add an SAML identity provider, you must have a role based on the role template `sap.hana.xs.admin.roles::SAMLAdministrator`.
- You have access to the XML document containing the IDP metadata that describes the SAML identity provider (SAML IDP) you want to modify.

Context

To edit the details of an SAML identity provider, you can use the *SAP HANA XS Administration Tool*, as described in the following steps:

> ⚠ Caution
>
> To maintain the details of SAML identity providers, you must be familiar with the technical background of SAML SSO mechanisms and requirements.

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform SAML administration tasks.

2. Start the *SAML Identity Provider* tool.

   Choose *SAML Identity Provider* in the list of tools displayed on the left-hand side of the *SAP HANA XS Administration Tool* window.

3. Select the SAML identity provider, whose details you want to modify.

   The list of available SAML IDPs is displayed in the *SAML Identity Provider List* on the left-hand side of the *SAML Identity Provider* tool.

4. Modify the details of the selected SAML SSO identity provider (IDP).

   The information required to maintain details of an SAML IDP is specified in an XML document containing the IDP metadata. This document should be available as part of the SAML service you want SAP HANA XS to use.

   a. Paste the contents of the XML document containing the IDP metadata into the *Metadata* box in the *Add Identity Provider Info* screen area.
   If the contents of the XML document are valid, the parsing process extracts the information required to insert into the *Subject*, *Entity ID*, and *Issuer* fields in the *General Data* screen area, and the URL fields in the *Destination* screen area.

5. Save the modifications to the SAML identity provider.

   Choose *Save* to save the changes.

Related Information

Add an SAML Identity Provider [page 1171]

14.1.7 Maintaining SMTP Server Configurations

Define details of the SMTP server that SAP HANA XS can use to respond to requests from applications to send e-mails.

The SMTP configuration defines the details of the SMTP server that is available for use by all applications running on an SAP HANA XS server. You can configure one SMTP server per SAP HANA XS server. As part of the configuration, you also specify the following options:

- General SMTP details
- Logon authentication type
- Transport-channel security type
- Other settings

> **Note**
>
> SAP does not provide any e-mail services. You will need to set up your own e-mail servers or cooperate with an external e-mail provider of your choice. You can configure only one SMTP server for each SAP HANA XS instance.

SMTP Host System Details

When defining the details of the SMTP server to be used by the SAP HANA XS applications, you must specify the following elements:

- *Mail Server Host*
  The name or the IP address of the system hosting the SMTP relay server that the XS applications can use to send an e-mail, for example, `<smtp.mail.host>` or "*localhost*" if the system hosting the SMTP server is the same as the one hosting the XS instance.
- *Mail Server Port*
  The port number to use for connections to the SMTP relay server, for example, *25*.

  > **Note**
  >
  > The port number to use can change according to the security type specified for the SMTP transport channel, for example, 465 (SMTPS) or 587 (STARTTLS).

SMTP Logon Authentication Type

You must tell SAP HANA XS which method the SMTP server uses to authenticate the logon credentials of the user that SAP HANA uses to establish the connection. The available choices for the authentication type are: *None*, *Auto*, *Logon*, *Plain*, *CRAM-MD5*, or *Digest-MD5*.

If you choose the option *None*, no logon credentials are required for the connection to the SMTP relay server. If you choose the option *Auto*, SAP HANA XS checks the authentication mechanisms supported by the SMTP relay server and selects one automatically according to the following order of preference: *Digest-MD5*, *CRAM-MD5*, Plain, *Login*, or *None*.

> **Note**
>
> For all authentication-type options except *None*, you must specify the name and password of the user whose credentials SAP HANA XS uses to log on to the SMTP server.

SMTP Transport-Channel Security Type

When you set up the SMTP configuration, you must specify the security type used to encrypt the transport channel between the SAP HANA XS server and the SMTP server; you can choose any of the following values:

- *None*
  This is default value for the transport security type; the channel used to communicate with the SMTP relay server is not encrypted. Note that is it possible that both SAP HANA XS and the specified SMTP relay server are running in the same trusted network or even on the same host.
- *STARTTLS*
  You can specify STARTTLS as the transport security only if it is supported by the SMTP relay server. If it is not supported, the application trying to send an e-mail encounters and error and the requested e-mail message is not sent.
- *SSL/TLS*
  Use an SSL/TLS-wrapped channel to communicate with the SMTP relay server. If SSL/TLS is not supported by the SMTP relay server then the connection cannot be established, the application trying to send an e-mail encounters and error, and the requested e-mail message is not sent. If you choose SSL/TLS as the transport security type, you will very probably have to specify a different port, usually 465, in the SMTP host section. You will also have to specify the name of the trust store holding the certificates and keys required to establish a trusted connection with the SMTP server.

> **Note**
>
> If the SMTP relay server's certificate cannot be verified, then the connection to the specified SMTP server cannot be established, the application trying to sent an e-mail encounters an error, and the requested e-mail message is not sent.

Socket Proxy Settings

If your system uses a proxy service for Socket Secure (SOCKS) routing, you need to enable support using the SOCKS Proxy toggle button (ON) and, in addition, provide connection details for the system where the proxy service is running, for example:

> **Caution**
>
> The proxy-server settings you define here are overriden by any SAP HANA system wide setting for a proxy server, for example, defined by the `enforced_outbound_proxy` parameter in the `communication` section of the `xsengine.ini` configuration file.

- *Proxy Host*
  The name of the system hosting the SOCKS proxy service

- *Proxy Port*
  The port number to use for connections to the SOCKS proxy server running on the host specified in *Proxy Host*
- *Proxy Username*
  The name of the user whose accound is used to log on to the SOCKS proxy server system specified in *Proxy Host*
- *Proxy Password*
  The password of the user whose accound is used to log on to the SOCKS proxy server system specified in *Proxy Host*

Other Settings

You can specify the maximum length of time (in milliseconds) that SAP HANA XS must wait for a response from the SMTP relay server with which it is trying to establish a connection; the default value is 60000 milliseconds (1 minute). It the specified timeout limit is reached, the connection is reset and the application requesting the connection encounters an error.

> **Note**
>
> If a connection is reset due to a timeout problem, the state of any sent e-mail messages is unknown. However, some useful information might be available in the logs of the SMTP relay server.

Related Information

Create an SMTP Configuration [page 1178]
SAP HANA XS Classic Configuration Parameters [page 1140]

14.1.7.1 Create an SMTP Configuration

Define the settings an SAP HANA XS application uses for outbound connections to an SMTP server.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the *SAP HANA XS Administration Tool*. To access the *SMTP Configuration* tools that enable you to set up an SMTP server for SAP HANA XS applications, you must have roles based on the following role templates:

- `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
- `sap.hana.xs.admin.roles::SMTPDestAdministrator`

Context

The SMTP configuration defines the details of the SMTP server that is available for use by all applications running on an SAP HANA XS server. You can configure one SMTP server per SAP HANA XS server. As part of the configuration, you also specify what authentication type to use when establishing the connection as well as the security type used to encrypt the transport channel between the SAP HANA XS server and the SMTP sever, for example, SSL or TLS. To create an SMTP configuration for an SAP HANA XS application, perform the following steps:

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *SMTP Configurations* tool.

2. Start the *SMTP Configurations* tool.

   In the list of XS Administration tools, choose *SMTP Configurations* to display the screen where you can manage the configuration of the SMTP server used by SAP HANA XS applications.

   > ⓘ Note
   >
   > SAP does not provide any e-mail services. You will need to set up your own e-mail servers or cooperate with an external e-mail provider of your choice. You can configure only one SMTP server for each SAP HANA XS instance.

3. Specify details of the system hosting the SMTP server that the SAP HANA XS applications must use.

   Provide the name of the system hosting the SMTP server and the port number required to open a connection.

   > ⓘ Note
   >
   > If SSL or TLS is required to encrypt the transport channel, the port number will probably change, for example, to 465 (SMTPS) or 587 (STARTTLS).

4. Specify the authentication settings required for access to the SMTP host.

   Choose an authentication method from the *Authentication Type* drop-down list, for example, *auto*, *logon*, or *none* and, if necessary, the user credentials required to log on to the SMTP server.

   > → Tip
   >
   > If you choose *auto*, setup checks the authentication mechanisms supported by the specified SMTP server and selects one in the following order of preference: *Digest-MD5*, *CRAM-MD5*, *Plain*, *Login*, and *None*.

5. Specify the security settings for the transport-channel.

   The transport channel is used for the communication between the SAP HANA XS application and the SMTP server. If you choose either the *STARTTLS* or the *SSL/TLS* option, use the *Trust Store* drop-down list to specify the trust store where the certificates and keys for the SMTP sever are located.

   > ### Note
   >
   > If you choose the option *None*, the channel used to communicate with the SMTP relay server is not encrypted.

6. Define the timeout setting for connections to the specified SMTP server.

   You can specify the maximum length of time (in milliseconds) that SAP HANA XS must wait for a response from the SMTP server with which it is trying to establish a connection; the default value is 60000 milliseconds (1 minute).

   > ### Note
   >
   > It the specified timeout limit is reached, the connection is reset and the application requesting the connection encounters an error.

7. Define the socket proxy settings.

   If your system uses a proxy service for Socket Secure (SOCKS) routing, you need to enable support using the *SOCKS Proxy* toggle button (*ON*) and, in addition, provide connection details for the system where the proxy service is running, for example, the host name, the port number to use for connections, and the user credentials required to log on.

   > ### Caution
   >
   > The proxy-server settings you define here are overriden by any SAP HANA system wide setting for a proxy server, for example, defined by the `enforced_outbound_proxy` parameter in the `communication` section of the `xsengine.ini` configuration file.

8. Save the changes you have made to the SMTP configuration.

Related Information

Maintaining SMTP Server Configurations [page 1176]
SMTP Configuration Details [page 1180]

14.1.7.1.1 SMTP Configuration Details

The SMTP configuration defines the details of the SMTP server that is available for use by all applications running on an SAP HANA XS server.

As part of the SMTP configuration, you specify the following options:

- General SMTP settings [page 1181]

- Logon authentication type [page 1181]
- Transport security type [page 1182]
- Socket proxy settings [page 1182]
- Other settings [page 1182]

General SMTP settings

The *General SMTP Settings* screen area of the *SMTP Configurations* tool enables you to maintain the basic details of the system hosting the SMTP server that SAP HANA XS applications use to send e-mail. The following table indicates which information can be maintained.

UI Element	Description	Example
Mail Server Host	The name of the system hosting the SMTP server.	localhost, `<hostname>`,
Mail Server Port	The port to connect to on the SMTP server. The port number will change depending on the choices made for the transport-security settings described below (None, SSL/TLS, STARTTLS)	25, 465, 587

> **Note**
>
> SAP does not provide any e-mail services. You will need to set up your own e-mail server or make arrangements with an external e-mail provider of your choice. You can configure only one SMTP server for each SAP HANA XS instance.

Authentication

The *Authentication* screen area of the *SMTP Configurations* tool enables you to maintain details of the user credentials required to log on to the system hosting the SMTP server and the mechanism used during the logon process to carry out user authentication. The following table indicates which information can be maintained.

UI Element	Description	Example
Authentication Type	The method used by the SMTP server to authenticate the credentials of the user that SAP HANA uses to establish the connection	None, Auto, Logon, Plain, CRAM-MD5, or Digest-MD5.
Username	For all authentication-type options except *None*, the name and password of the user whose credentials SAP HANA XS uses to log on to the SMTP server.	johndoe
Password	For all authentication-type options except *None*, the password of the user whose credentials SAP HANA XS uses to log on to the SMTP server.	*****

Transport Security Settings

The *Transport Security Settings* screen area of the *SMTP Configurations* tool enables you to maintain details of the security type used to encrypt the transport channel between the SAP HANA XS server and the SMTP server. The following table indicates which information can be maintained.

UI Element	Description	Example
Transport Security	The method used by the SMTP server to authenticate the credentials of the user that SAP HANA uses to establish the connection	None, STARTTLS, SSL/TLS
Trust Store	Contains the certificates used to establish trust relationships between servers, for example, SAP HANA XS and the SMTP server	sapspv.pse

Socket Proxy Settings

The *Socket Proxy Settings* screen area of the *SMTP Configurations* tool enables you to maintain details of the system hosting the proxy service used by the SMTP server for Secure Socket (SOCKS) routing. The following table indicates which information can be maintained.

UI Element	Description	Example
SOCKS Proxy	Enable/Disable Socket Secure (SOCKS) routing	N/A
Proxy Host	Name of the system hosting the proxy service for Socket Secure (SOCKS) routing	smtp.host.acme.com
Proxy Port	Port number to use for connections to the proxy server	1080
Proxy Username	Name of the user required to log on to the proxy server	johndoe
Proxy Password	Password for the user required to log on to the proxy server	****

Other Settings

The *Other Settings* screen area of the *SMTP Configurations* tool enables you to maintain additional details of the SMTP server, for example, the connection timeout setting. The following table indicates which information can be maintained.

UI Element	Description	Example
Timeout	Maximum length of time (in milliseconds) that SAP HANA XS must wait for a response from the SMTP server with which it is trying to establish a connection	60,000 milliseconds (1 minute)

Related Information

Create an SMTP Configuration [page 1178]

14.1.8 Maintaining HTTP Access to SAP HANA

Ensure that Web-based applications have access to SAP HANA via HTTP.

To enable access to the services provided by the XS-based applications that you develop for SAP HANA, you need to ensure that client applications can access the SAP HANA XS Web server by HTTP or HTTPS. As part of the configuration process, you also need to configure SSL (for use with secure HTTP), set up the SAP Web Dispatcher (for example, to use non-defualt ports or secure HTTP), and maintain the trust stores that store the certificates required for secure communication. In addition, in a multi-database environment, you also need to configure HTTP access to multi-tenant database containers.

Maintaining HTTP access to SAP H ANA includes one of more of the following tasks:

- Configure HTTPS (SSL) for client application access
  Configure the SAP Web Dispatcher to use HTTPS (SSL) for incoming requests from UI front ends and applications, for example, SAP HANA applications. The requests are then forwarded by the SAP Web Dispatcher to SAP HANA.
- Maintain standard HTTP port numbers for SAP HANA XS
  Check or change the default HTTP port settings, for example, to ensure that standard ports 80 and 443 are used for client access to the SAP HANA XS Web server by HTTP or HTTPS, respectively.
- Configure HTTP access to multi-tenant database containers
  Configure the internal SAP Web Dispatcher so that, in an environment where multiple tenant database containers are available, the SAP Web Dispatcher knows which client requests to dispatch to which tenant database, for example, on the basis of alias DNS names.

Related Information

Configure HTTPS (SSL) for Client Application Access [page 1184]
Maintain Standard HTTP Port Numbers with SAP HANA XS [page 1185]
Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic [page 1187]

14.1.8.1 Configure HTTPS (SSL) for Client Application Access

To improve the security of your SAP HANA landscape, you can configure the SAP Web Dispatcher to use HTTPS (SSL) for incoming requests from UI front ends and applications, for example, SAP HANA applications. The requests are then forwarded to SAP HANA.

Prerequisites

If you want to set up a secure SSL connection (Secure Socket Layer) between client applications and the SAP Web Dispatcher, the following components are prerequisites:

- The CommonCryptoLib library (`libsapcrypto.so`)
  CommonCryptoLib (`libsapcrypto.so`) is installed by default as part of SAP HANA server installation at $DIR_EXECUTABLE.
- You have a role based on the role template `sap.hana.xs.wdisp.admin::WebDispatcherAdmin`. This is required to access the SAP HANA *Web Dispatcher Administration* tool.

Context

The SAP Web dispatcher lies between the Internet and your SAP system. It is the entry point for HTTP(s) requests into your system. To configure the SAP Web Dispatcher to use SSL for inbound application requests, perform the following steps.

Procedure

1. Start the *SAP HANA Web Dispatcher Administration* tool.

   The *SAP HANA Web Dispatcher Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/wdisp/admin/`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *Web Administration Interface* of the *SAP HANA Web Dispatcher Administration* tool.

2. Create an SSL key pair and a certificate request:

   The SSL key pair is created with the default `SAPSSLS.pse` trust store; if you want to create a new SSL key pair, choose *Recreate PSE* in the *PSE Management* tool. To create a certificate request, perform the following steps:

   a. Open the *PSE Management* tool.

      In the *SAP HANA Web Dispatcher Administration* tool, choose *SSL and Trust Configuration* > *PSE Management*.

b. Create the certificate request.

   In the *PSE Management* screen area, choose *Create CA Request*.

c. Submit the generated certificate request to your certificate authority (CA) for signing.

   Copy the contents of the certificate request from the *CA Request of PSE SAPSSLS.pse* screen area and send it to your certificate signing authority.

3. Import the signed certificate.

   Add a copy of the signed certificate to the `SAPSSLS.pse` trust store. The certificate-request response must be generated in the correct format, for example, PKCS#7 certificate chain, which contains both the requester's and the issuing CA's certificates. If the response contains only the requester's certificate in PEM (Privacy Enhanced Mail) format and no CA certificate, the system can still build the correct format. However, in this case, the issuing CA's root certificate must already be available in the same certificate store into which you import the requester's certificate.

   > → Tip
   >
   > Make sure that the date and time settings on the server hosting the SAP Web Dispatcher are correct and synchronized with the certificate authority (CA) that issued the certificate you import, otherwise the certificate might be interpreted as invalid.

   a. Open the *PSE Management* tool.

      In the *SAP HANA Web Dispatcher Administration* tool, choose *SSL and Trust Configuration* > *PSE Management*.

   b. Select the target trust store.

      In the *Manage PSE* screen area, choose *SAPSSLS.pse* from the drop-down menu.

   c. Import the signed certificate request.

      In the *PSE Attributes* screen area, choose *Import CA Response* and copy the signed certificate response from your CA into the *Import CA Request of PSE SAPSSLS.pse* screen area.

14.1.8.2 Maintain Standard HTTP Port Numbers with SAP HANA XS

The default HTTP port settings for SAP HANA XS include an SAP HANA instance number, for example, `80<SAP HANA instance>` (8000). You can change the default settings, for example, to ensure that standard ports 80 and 443 are used for client access to the SAP HANA XS Web server by HTTP or HTTPS.

Prerequisites

To configure the SAP HANA XS server to use the standard HTTP ports `80` and `443`, bear in mind the following prerequisites:

- Superuser authorization is required to bind ports with a number less that (<) 1024 (well-known ports) on a UNIX system
- Neither the ICM process nor the SAP Web Dispatcher has the superuser authorization.

Context

By default, the SAP HANA XS Web server is configured to use the port numbers `80<SAP HANA instance number>` for HTTP and `43<SAP HANA instance number>` for HTTPS requests from clients. You can change this behavior, for example, to configure the SAP HANA XS server to use the standard HTTP ports `80` and `443`, as follows:

Procedure

1. Open the instance profile of your SAP Web Dispatcher.

   The SAP Web Dispatcher profile can be found in the following location in the SAP HANA studio:

   SAP HANA Administration Console > *Configuration* > *webdispatcher* > *[profile]*

2. Check and, if necessary, modify the HTTP/S port settings in the SAP Web Dispatcher profile, as follows:

   ```
   icm/server_port_0 = PROT=HTTP, PORT=80, PROCTIMEOUT=600, EXTBIND=1
   icm/server_port_1 = PROT=HTTPS, PORT=443, PROCTIMEOUT=600, EXTBIND=1
   ```

   Save the changes to the SAP Web Dispatcher profile.

3. Bind the default SSL port to use.

   Since only users with superuser authorization rights can bind ports with a number less that (<) 1024 (well-known ports) on a UNIX system, and the ICM process or the SAP Web Dispatcher should not have these rights (and ICM cannot have them for technical reasons), the port must be bound by an external program and the listen socket then transferred to the calling process. You can use the `icmbnd` command.

   > **Note**
   >
   > The installation process creates the file `icmbnd.new`, which you must rename to `icmbnd` and configure as described below. This applies after a system update, too.

   Since superuser privileges are required to bind ports with a number lower than 1024, you must change the owner and permissions of the `icmbnd` command, for example, from `<SID>adm` to user `root`.

   a. Change the owner of the `icmbnd` command:

      ```
      $> chown root:sapsys icmbnd
      ```

   b. Change the permissions for the `icmbnd` command:

      ```
      $> chmod 4750 icmbnd
      ```

   c. Check the new permissions for the `icmbnd` command:

      ```
      $> ls -al
      rwsr-x 1 root sapsys 1048044 Feb 13 16:19 icmbnd
      ```

Related Information

SAP Web Dispatcher: Binding Ports < 1024 on UNIX

14.1.8.3 Configure HTTP(S) Access to Tenant Databases via SAP HANA XS Classic

To enable Web-based applications to send HTTP(S) requests to tenant databases via the SAP HANA XS classic server, the internal SAP Web Dispatcher must be configured so it knows which requests to dispatch to which database on the basis of DNS alias host names. You do this by specifying the public URL of every tenant database in the `xsengine.ini` configuration file.

Prerequisites

- You are logged on to the system database.
- You have the system privilege INIFILE ADMIN.
- The network administrator has defined an alias hostname in your organization's Domain Name System (DNS) for every tenant database in the SAP HANA system. The alias hostname must refer to the hostname of the machine that is used for HTTP(S) access to the tenant database.
- You have a role based on the role template `sap.hana.xs.wdisp.admin::WebDispatcherAdmin`. This is required to access the SAP HANA Web Dispatcher Administration tool for configuring HTTPS.

Context

The XS classic server allows Web-based applications to access SAP HANA via HTTP(S). The internal Web Dispatcher of the SAP HANA system manages these incoming HTTP(S) requests. To allow applications to send requests to specific databases, every tenant database needs an alias host name. Requests to the alias host name can then be forwarded to the XS server of the corresponding tenant database. Requests with the physical host name in the HTTP host header are forwarded to the XS server running on the system database.

The default HTTP ports are used in all cases, that is, 80`<instance>` (HTTP) and 43`<instance>` (HTTPS). Alias host names are mapped to internal HTTP(S) ports so that incoming requests can be routed to the correct database.

You configure HTTP(S) access to tenant databases by specifying in the `xsengine.ini` file the URLs by which each tenant database is publicly accessible. The system then automatically configures the Web Dispatcher by generating the required profile entries in the `webdispatcher.ini` configuration file. It is not necessary to specify the URL of the system database, this is done automatically.

> **Note**
>
> This automatic configuration of the Web Dispatcher is controlled by the parameter `[profile] wdisp/system_auto_configuration` in the `webdispatcher.ini` configuration file. If this parameter is set to `false`, you need to configure the `webdispatcher.ini` file manually.

For HTTPS access, you must subsequently configure the required client certificates and trust stores using the SAP Web Dispatcher Administration tool. The following approaches are supported:

- Using a single "wildcard" server certificate in a single trust store that covers all databases in the system
  Wildcard certificates are more flexible when tenant databases are frequently added and deleted. However, if you use a wildcard certificate, either the server requires its own sub-domain or you must ensure that the certificate cannot be abused from other servers.

  > **Caution**
  >
  > Do not use a wildcard server certificate if strict isolation between tenant databases is required. If authentication relies on a wildcard certificate and a shared trust store, users of one tenant database will be able to log on to other databases in the system.

- Using individual certificates in individual trust stores for each database
  Individual certificates for each database are more suitable in a flat domain structure for individual servers. They also ensure strict isolation between tenant databases. However, they involve more administrative effort to maintain.

Procedure

1. Specify the public URLs of all tenant databases in the xsengine.ini file in one of the following ways:

Option	Description
SAP HANA studio	1. Open the Administration editor and choose the *Configuration* tab. 2. Navigate to the `xsengine.ini` file and expand the `public_urls` section. 3. For each tenant database in the system, add the new properties `http_url` and `https_url` at the **database layer** and enter its public URL as the value: • `http://<virtual_hostname>:80<instance>` • `https://<virtual_hostname>:43<instance>` > **Note** > The scheme (`http/https`) must be included in the URL.
SQL	For each tenant database, execute the statements: • ALTER SYSTEM ALTER CONFIGURATION ('xsengine.ini', 'database', '`<tenant_DB_name>`') SET ('public_urls', 'http_url') = 'http://`<virtual_hostname>`:80`<instance>`' WITH RECONFIGURE; • ALTER SYSTEM ALTER CONFIGURATION ('xsengine.ini', 'database', '`<tenant_DB_name>`) SET ('public_urls', 'https_url') = 'https://`<virtual_hostname>`:43`<instance>`' WITH RECONFIGURE;

> **Note**
>
> The following values are set at the **default layer** and represent the URLs of the system database:

- `http://$(SAPLOCALHOST):80$(SAPSYSTEM)`
- `https://$(SAPLOCALHOST):43$(SAPSYSTEM)`

By default, the system database initially retrieves any request with the port 80<instance_no>. However, as soon as you configure the URLs of tenant databases, it is available under http://<localhost>:80<instance> only, and not the fully qualified domain name (FQDN). The local host is known to SAP HANA without the FQDN.

If you want to change this default behavior and configure a different URL for the system database, you can do so by executing the following statement:

```
ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini', 'system')
SET('public_urls', 'http_url') = 'http://<virtual_hostname>:80<instance>'
WITH RECONFIGURE;
```

New entries are now created in the `webdispatcher.ini` file at the host layer for every database. You can verify this by executing the following statement (from the system database):

```
SELECT KEY, VALUE, LAYER_NAME FROM SYS.M_INIFILE_CONTENTS WHERE FILE_NAME =
'webdispatcher.ini' AND SECTION = 'profile' AND KEY LIKE 'wdisp/system%'
```

This returns the following result for example:

```
KEY         | VALUE                                                | LAYER_NAME
wdisp/system_0     | GENERATED, SID=SYS, EXTSRV=http://localhost:30014,
SRCVHOST='myhost'              | DEFAULT
wdisp/system_1     | GENERATED, SID=MYD, EXTSRV=http://localhost:30042,
SRCVHOST='mydatabase.example.com'    | HOST
```

2. Optional: Secure incoming communication by configuring HTTPS.

Option	Description
Single certificate for all databases	1. Start the SAP HANA Web Dispatcher Administration tool at http://<localhost>:80<instance>/sap/hana/xs/wdisp/admin/. 2. For the default `SAPSSLS.pse` trust store, create a new SSL key pair and certificate request: 1. From the main menu, choose *SSL and Trust Configuration* > *PSE Management*. 2. From the *Manage PSE* menu, choose *SAPSSLS.pse*. 3. Choose *Recreate PSE*. 4. Enter a distinguished name that matches the host name of all tenant databases. **Example** • Physical host name: myhost.example.com • Tenant host name 1: mydatabase1.example.com • Tenant host name 2: mydatabase2.example.com In this case, you specify **`CN=*.example.com`** as the DN, thus creating a server certificate that matches all tenant databases and the system database. 5. Choose *Create*. 6. Create a certificate request and submit to your certificate authority (CA) for signing (*Create CA Response*). 3. Import the signed certificate For more information, see *Configure HTTPS (SSL) for Client Application Access*.

Option	Description

Individual certificates for each database	1. Start the SAP HANA Web Dispatcher Administration tool at http://<localhost>:80<instance>/sap/hana/xs/wdisp/admin/.
	2. For each tenant database and the system database, create a new trust store with a unique certificate:
	1. From the main menu, choose *SSL and Trust Configuration* > *PSE Management*.
	2. On the PSE management screen, choose *Create New PSE*.
	3. Enter a file name for the new PSE.
	Example
	`example.pse`
	4. Enter the distinguished name:
	`CN=<host name used for the tenant database in the public_urls section of the xsengine.ini file>`
	5. Choose *Create*.
	6. For the new PSE, create a certificate request and submit to your CA for signing (*Create CA Response*).
	7. Import the signed certificate into the new PSE (*Import CA Response*).
	3. Configure the Web Dispatcher to use multiple certificates:
	1. In the `webdispatcher.ini` file, create or change the parameter `[profile] icm/ssl_config_0`, specifying as the value:
	`ID=ssl_config_main, CRED=SAPSSLS.pse, SNI_CREDS=<semicolon (';') separated list of database PSE files>`
	2. Add **`,SSLCONFIG=ssl_config_main`** to the value of the `icm/server_port` parameter for the HTTPS port (by default `icm/server_port_1`).
	Example
	`icm/server_port_1 = PROT=HTTPS,PORT=4443$(SAPSYSTEM),PROCTIMEOUT=600, SSLCONFIG=ssl_config_main`

Results

You can access the XS server of tenant databases via the configured URLs.

→ Tip

If you experience slow response times when accessing the XS server of a tenant database (for example, Web-based applications running on the tenant database), this indicates that the server is not able to resolve the host name correctly using the DNS and retries repeatedly. If this is the case, contact your network administrator for a detailed problem analysis.

As a workaround, you can manually override virtual host name resolution on the machine where the browser is running by modifying the `/etc/hosts` file on the local machine. In this file, append a new line, starting with the static IP address of the server, followed by the virtual host name of your tenant database, for example, "10.20.30.40 mydatabase.example.com". To edit this file you need admin or root privileges.

Next Steps

Optional: Enable access to Web-based applications from the SAP HANA studio.

Some Web-based tools are accessible from the SAP HANA studio, for example, the SAP HANA cockpit and SAP HANA Lifecycle Management tool. If you want to be able to access these tools from a tenant database registered in the studio, you must specify the alias hostname in the properties. You can do this as follows:

1. In the *Systems* view, right-click the tenant database and choose *Properties*.
2. Open the *XS Properties* page and enter the alias hostname in the *XS Host* field.

Related Information

Configure HTTPS (SSL) for Client Application Access [page 1184]
Using SAP Web Dispatcher for Load Balancing with Tenant Databases [page 115]

14.1.9 Maintaining Single Sign-On for SAP HANA XS Applications

You can configure SAP HANA applications to use single sign-on (SSO) authentication to confirm the logon credentials of a user calling an application service. SAP HANA supports SSO certificates based on the Security Assertion Markup Language (SAML) or X.509.

If you want your the SAP HANA XS applications to use an SSO certificate based on SAML or X.509 as the logon authentication method, you must perform the following high-level steps:

1. Maintain the SAP HANA trust store.
   SAP HANA makes use of multiple trust stores.

   > ⓘ Note
   >
   > The trust stores listed below are located in the **file system**. In some cases, it is possible and recommended to use trust stores that exist in the database as database objects. In-database trust stores (referred to as certificate collections) contain the required client certificates, which are also stored in the database. We recommend using in-database certificate collections where possible. For more information, see *Managing Client Certificates*.

   - The SAP HANA trust store (`sapsrv.pse`)
     Used for secure SQL and SAML or OAuth scenario, `sapsrv.pse`. This trust store is not created automatically. You must generate it manually (for example using the SAP Web Dispatcher administration tool or the SAPGENPSE tool) and store it in the $SECUDIR directory.

     > → Recommendation
     >
     > For user authentication based on X.509 certificates and SAML assertions, we recommend creating separate certificate collections with the purposes *X.509* and *SAML* instead of using the file system-based trust store `sapsrv.pse`.

- The SAP Web Dispatcher trust store (`SAPSSLS.pse`)
  Required for secure connections using the Secure Socket Layer (SSL) protocol, `SAPSSLS.pse` is installed automatically and is available by default.
- The SAP Logon Ticket trust store (`saplogon.pse`)
  Optional: `saplogon.pse` is only necessary if an SAP HANA XS application requires an SAP logon ticket from a user at logon.

  > → Recommendation
  >
  > For user authentication based on logon tickets, we recommend creating a certificate collection with the purpose *SAP LOGON* instead of using the file system-based trust store `saplogon.pse`.

- The client authentication trust store (`SAPSSLC.pse`)
  Optional: `SAPSSLC.pse` is only required for client connections, for example, that use the SQL client interface (`hdbsql`).

2. Choose the SSO authentication method and configure the trust relationships:
   Trust relationships are required between SAP HANA and any remote system providing services that an SAP HANA XS application requires.
   - SSO with X.509 certificates
     Add the root certificate of the Certification Authority (CA) for trusted X.509 certificates to both the SAP HANA trust store **and** the trust store for the SAP Web Dispatcher.
   - SSO with SAML certificates
     Obtain, authenticate, and import the SAML identity provider (IDP) metadata (an XML document) for the SAML service provider.
3. Maintain the SSO provider for SAP HANA XS
   Maintain a runtime configuration for the SAP HANA application, which indicates that user authentication is by means of SSO certificates based on either SAML or X.509.

Related Information

Managing Client Certificates [page 542]
Configure SSO with X.509 Authentication for SAP HANA XS Applications [page 1193]
Configure SSO with SAML Authentication for SAP HANA XS Applications [page 1195]

14.1.9.1 Configure SSO with X.509 Authentication for SAP HANA XS Applications

SAP HANA applications can use single sign-on (SSO) authentication with X.509 certificates to confirm the logon credentials of a user calling an application service.

Prerequisites

- You have a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator`.
- The CommonCryptoLib library (`libsapcrypto.so`) is installed and available.
- A certificate collection with the purpose *X.509* is available. For more information, see *Managing Client Certificates in the SAP HANA Database*.
- The SAP Web Dispatcher trust store (`SAPSSLS.pse`) is available.

Context

To enable SAP HANA applications to use single sign-on (SSO) authentication with X.509 certificates to confirm the logon credentials of a user, you need to add the root certificate of the Certification Authority that issues trusted X.509 certificates to both the SAP HANA trust store for X.509 authentication and the trust store of the SAP Web Dispatcher, `SAPSSLS.pse`.

Procedure

1. Add the root certificate (for example, `SSO_CA.der`) to the SAP HANA trust store, that is the certificate collection with purpose *X.509*.
   a. Open the SAP HANA cockpit.
   b. Open the *Certificate Store* app.
   c. Import the root certificate into the certificate store.
   d. Open the *Certificate Collections* app.
   e. Select the collection with purpose *X.509*.
   f. Add the root certificate to this collection.
2. Add the root certificate (for example, `SSO_CA.der`) to the SAP Web Dispatcher trust store (`SAPSSLS.pse`).
   a. Start the *SAP HANA Web Dispatcher Administration* tool.
   b. Open the *PSE Management* tool.

      In the *SAP HANA Web Dispatcher Administration* tool, choose *SSL and Trust Configuration* > *PSE Management*.

c. Specify the trust store (PSE file) for the import operation.

   In the *PSE Management* screen area, choose *SAPSSLS.pse* from the *Manage PSE* drop-down list.

d. Import the `SSO_CA.der` certificate.

   In the *Trusted Certificates* screen area, choose *Import Certificate*.

Alternatively, you can also use the sapgenpse tool to import the `SSO_CA.der` certificate.

```
./sapgenpse maintain_pk -p /usr/sap/<SAPHANAInstance>/HDB<InstNo>/
<Hostname>/sec/SAPSSLS.pse -a SSO_CA.der
```

3. Maintain the authentication settings in the runtime configuration for your SAP HANA XS application.

   You can use the Web-based SAP HANA XS Administration *Trust Manager* tool to complete this step. The tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > The user maintaining the security settings needs the privileges granted by the SAP HANA XS role *TrustStoreAdministrator*.

4. Create a database user whose identity is defined in an X.509 certificate issued by your CA.

   a. Create a new user in the SAP HANA database based on the details specified in an existing X.509 certificate.

      The following example shows how to use the SQL statement `CREATE USER WITH IDENTITY` to create the database user “MyUserName” and the corresponding X.509 certificate:

      ```
      CREATE USER MyUserName WITH IDENTITY 'CN=MyUserName, O=SAP-AG, C=DE' ISSUER
      'CN=SSO_CA, O=SAP-AG, C=DE' FOR X509
      ```

   b. Import into the Web browser the X.509 certificate that is to be used to authenticate the new database user.

5. Use a Web browser to test the logon authentication settings for the SAP HANA application.
   When you enter the URL for your application in the Web browser, the Web browser prompts you to select a certificate, which enables you to log on without supplying logon credentials manually.

Related Information

Managing Client Certificates [page 542]

14.1.9.2 Configure SSO with SAML Authentication for SAP HANA XS Applications

SAP HANA applications can use single sign-on (SSO) authentication with SAML assertions to confirm the logon credentials of a user calling an application service. SAML assertions are certificates that comply with the Security Assertion Markup Language.

Prerequisites

- You have an advanced understanding of how SAML works.
- The CommonCryptoLib library (`libsapcrypto.so`) is installed and available on the SAP HANA server.
- You are authorized to edit the certificate collection with purpose *SAML*. You need system privilege `CERTIFICATE ADMIN` and object privilege `ALTER` on the collection. For more information, see *Managing Client Certificates*
- An SAML identity provider (IDP) is available and the corresponding SAML metadata (in the form of an XML document). For more information see *Add an SAML Identity Provider*.
- You have root/administrator access to the SAP HANA system that is configured to act as an SAML **service** provider.
- To maintain security and authentication settings for SAP HANA XS applications, you must have a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator`. To maintain SAML settings for SAP HANA XS applications, you need a role based on the role template `sap.hana.xs.admin.roles::SAMLAdministrator`.

Context

To enable SAP HANA applications to use single sign-on (SSO) authentication with SAML assertions to confirm the logon credentials of a user, you must copy the SAML certificate from the SAML IDP metadata document and add the certificate to the SAP HANA trust store for SAML authentication.

Procedure

1. Gather the metadata for the SAML identity provider (IDP).

   This SAML IDP metadata typically takes the form of an XML document, which you can obtain from your security system administrator.
2. Extract the certificate string (which is DER encoded) from the SAML IDP metadata document.
   The certificate string is located in the `ds:X509Certificate` tag. For the SAP ID service, the certificate string could look like the following (incomplete) code example:

   ```
   MIICHTCCAYagAwIBAgIETKTcJjANBgkqhkiG9w0BAQUFADBTMQswCQYDVQQGEwJERTEPMA0G...
   ```

3. Paste the extracted SAML certificate string into a file called `sapid.cer`.
4. Add the BEGIN and END tags to the SAML certificate.

   The following example of a SAML certificate is incomplete; it is intended for illustration purposes only.

   ```
   -----BEGIN CERTIFICATE-----
   MIICHTCCAYagAwIBAgIETKTcJjANBgkqhkiG9w0BAQUFADBTMQswCQYDVQQGEwJERTEPMA0G...
   -----END CERTIFICATE-----
   ```

5. Import the contents of the SAML certificate (`sapid.cer`) into the SAP HANA trust store, that is the certificate collection with purpose *SAML*.
   a. Open the SAP HANA cockpit.
   b. Open the *Certificate Store* app.
   c. Import the SAML certificate (`sapid.cer`) into the certificate store.
   d. Open the *Certificate Collections* app.
   e. Select the collection with purpose *SAML*.
   f. Add the SAML certificate (`sapid.cer`) to this collection.
6. Configure your SAP HANA system to act as an SAML service provider.

   For more information, see *Configure an SAP HANA System as an SAML Service Provider*.
7. Maintain the authentication settings in the runtime configuration for the SAP HANA XS application for which you want to enable SSO with SAML authentication.

   You can use the Web-based *SAP HANA XS Administration Tool* to complete this step.
   The tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > The user maintaining the authentication settings in an application's runtime configuration needs the privileges granted by the SAP HANA XS role *RuntimeConfAdministrator*.

Related Information

Maintaining SAML Providers [page 1167]
Managing Client Certificates [page 542]

14.1.9.3 Configure SSO with SAP Logon Tickets for SAP HANA XS Applications

SAP HANA applications can use single sign-on (SSO) authentication with SAP logon tickets to confirm the logon credentials of the user calling an application service.

Prerequisites

- You have administrator access to the SAP HANA system hosting the applications to which you want to enable access with SAP logon tickets.
- To maintain security and authentication settings for SAP HANA XS applications, you must have a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator`.
- The CommonCryptoLib library `libsapcrypto.so` is installed and available.
- A certificate collection with the purpose *SAP LOGON* is available. For more information, see *Managing Client Certificates*.

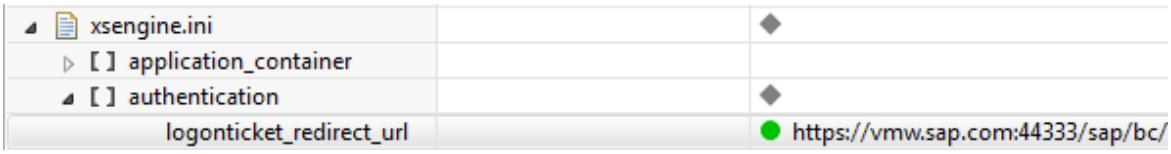
Context

To enable SAP HANA applications to use single sign-on (SSO) authentication with SAP logon tickets to confirm the logon credentials of a user requesting an application service, you must ensure that an SAP server is available that can issue SAP logon tickets. In addition, you need to add the server certificate of the ticket-issuing system to the SAP HANA trust store for authentication using logon tickets.

Procedure

1. Add the server certificate of the SAP system that issues SAP logon tickets to the SAP HANA trust store, that is the certificate collection with purpose *SAP LOGON*.
   a. Open the SAP HANA cockpit.
   b. Open the *Certificate Store* app.
   c. Import the server certificate of the ticket-issuing system into the certificate store.
   d. Open the *Certificate Collections* app.
   e. Select the collection with purpose *SAP LOGON*.
   f. Add the server certificate of the ticket-issuing system to this collection.
2. In SAP HANA, configure the details of the server that issues SAP logon tickets.

   This step is optional but ensures that an SAP logon ticket can always be obtained in those cases where no SAP logon ticket is immediately available for the user trying to log on.



a. Start the SAP HANA studio and open the *Administration* perspective.
b. In the *Configuration* tab, expand (or add) the section *xsengine.ini* › *authentication*.
c. Set (or add) the parameter: `logonticket_redirect_url`.

   Enter the URL that points to the system and service issuing SAP logon tickets, for example:

   ```
   https://<hostname>:<portnumber>/<path/to/logon_ticket/service>
   ```

   - `<hostname>`
     The hostname of the server issuing/storing the SAP logon tickets
   - `<portnumber>`
     The port number accepting connections on the target server issuing/storing the SAP logon tickets
   - `</path/to/logon_ticket/service>`
     Path to the service on the target system which handles the request for the SAP logon ticket. You can write your own custom ABAP service to handle these requests.

   For example, the following URL would enable access to the **custom** (user-defined) SAP logon ticket service `zredirectwlogon` using port `44333` on the ABAP server `host.acme.com`:

   ```
   https://host.acme.com:44333/sap/bc/zredirectwlogon?sap-
   client=<SAPClientNr>.
   ```

3. Maintain the runtime configuration for the application that you want to use SAP logon tickets for user authentication.

   You can use the Web-based *SAP HANA XS Administration Tool* to complete this step.
   The tool is available on the SAP HANA XS Web server at the following URL:
   `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`. Choose *XS Artifact Administration*.

   > ### Note
   > The user maintaining the security settings needs the privileges granted by the SAP HANA XS role *RuntimeConfAdministrator*.

   a. Locate the root package of the application whose runtime configuration you want to modify.
      Use the *Packages* list in the *Application Objects* pane.
   b. In the *Security & Authentication* tab, enable support for *SAP Logon/Assertion Ticket*.
   c. Save the changes you have made.

Related Information

Managing Client Certificates [page 542]

14.1.9.4 Configure Outbound SSO with Assertion Tickets

Assertion tickets are a form of bearer token that one application server uses to identify and authenticate a user on another application server, for example, in a single-sign-on (SSO) scenario. You can set up SAP HANA to function as the provider of the assertion tickets required to log on to a remote SAP server.

Prerequisites

To configure SAP HANA to use SAP assertion tickets to authenticate users who log on with SSO, note the following prerequisites:

- Your SAP HANA system is configured to use SSL
- You have administrator access to the SAP HANA system hosting the applications to which you want to enable access with SAP assertion tickets.
- To maintain security and authentication settings for SAP HANA XS applications, you must have a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator`. To maintain an HTTP destination, you need a role based on the role template `sap.hana.xs.admin.roles::HTTPDestAdministrator`.
- You know the system ID (SID) and client number of the SAP HANA system
- You know the system ID (SID) and client number of the remote SAP ABAP server that hosts the HTTP service (assertion-ticket provider) used by your XSJS application
- You have the permissions required to run transaction **STRUSTSSO2** in the ABAP system with which you want to establish a trust relationship.
- The CommonCryptoLib library `libsapcrypto.so` is installed and available on your SAP HANA system.
- You have read SAP Note 1982597 concerning SAP logon tickets and assertion tickets which are created with UTF-8.

Context

SAP HANA XS enables you to build XSJS applications that use single sign-on services with authentication using SAP assertion tickets to consume additional Web services, for example, provided by a remote ABAP application server. If the XSJS application service requires access to remote services, you can create an HTTP destination that defines the logon details required by the remote ABAP system and specifies SSO with SAP assertion tickets as the logon authentication method. The assertion ticket is included in the header of the HTTP request sent by the application service; the remote system reads the HTTP header and uses the assertion to log the requesting user on automatically.

Procedure

1. Create the SAP HANA trust store for the assertion tickets, for example, `saplogonSign.pse`.

This trust store is used to issue the assertion tickets required for automatic logon to remote SAP systems using SSO.

```
sapgenpse gen_pse -p saplogonSign.pse "CN=<HOST>.<DOMAIN>, OU=<INSTANCE>,
O=<ORG>, C=<COUNTRY>"
```

You are prompted to have the ticket signed by a Certificate Authority (CA):

a. Copy the certificate request and have it signed by a known CA service.
b. Copy the signed certificate results from the CA to the directory `/usr/sap/<SID>/HDB<Instance Number>/<machine name>/sec` on your SAP HANA system and name the file `saplogonSign.cer`.
c. Import the signed certificate into the trust store.

```
./sapgenpse import_own_cert -c saplogonSign.cer -p saplogonSign.pse -r
SAPNetCA.cer
```

2. Export the certificate that SAP HANA uses to sign assertion tickets.

   You need to save the exported certificate to a local file for future use.

   a. Export the SAP HANA certificate from the SAP HANA trust store, for example, using the following command:

   ```
   sapgenpse export_own_cert -p saplogonSign.pse
   ```

   b. Copy the output to a local file on your system.

3. Set up the trust relationship between SAP HANA and the remote SAP ABAP system you want to enable automatic logon with SSO and assertion tickets.

   The remote SAP system hosting the HTTP service you want your XSJS application to use must trust the SAP HANA system hosting the XSJS service itself and acting as a provider of SAP assertion tickets.

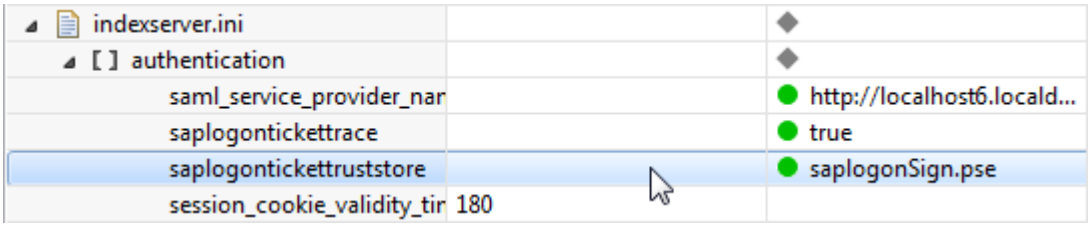
   a. Log on to the target ABAP system and run transaction **STRUSTSSO2**.
   b. Select the system PSE (trust store).
   c. Choose the *import certificate* button in the certificate section.
   d. Select the SAP HANA certificate you signed in the previous step and import it.
   e. Choose the *Add to certificate list* button.
   f. Choose the *Add to ACL* button.
   g. Provide the system ID (SID) for the SAP HANA system; the client number is `000`.
   h. Save the configuration.

4. Import the certificate of the system you want to trust for inbound SSO.

   > **Note**
   >
   > This step is optional; it is only required if you want to use SAP logon tickets for inbound SSO requests, too.

5. On the SAP HANA system, edit the configuration variable used to specify the name of the trust store for SAP assertion tickets.

   Start the SAP HANA studio's *Administration Console* perspective and edit the parameter `saplogontickettruststore`.You can find the `saplogontickettruststore` parameter in *[indexserver | xsengine].ini* > *authentication* > *saplogontickettruststore*.



6. Maintain an HTTP destination for the XSJS service that needs access to a remote SAP system and set the authentication type to *SAP Assertion Ticket*.

   You define the details of an HTTP destination in a configuration file that requires a specific syntax. The configuration file containing the details of the HTTP destination must have the file extension `.xshttpdest`.

   > **Caution**
   >
   > The HTTP destination configuration and the XSJS application that uses it must reside in the same application package. An application cannot reference an HTTP destination configuration that is located in another application package.

   a. Create a plain-text file called `<MyHTTPdestination>.xshttpdest` and open it in a text editor.
   b. Use the following code to help you define the HTTP destination details.

      > **Note**
      >
      > Change the entries for the host name, port, system ID and client to suit your own requirements.

      ```
      host = "<ABAP.server_.name>";
      port = <ABAP_HTTPS_PortNumber>;
      description = "my SAP assertion ticket target";
      useSSL = true;
      pathPrefix = "";
      authType = AssertionTicket;
      useProxy = false;
      proxyHost = "";
      proxyPort = 0;
      timeout = 0;
      remoteSID = "<ABAP_SID>";
      remoteClient = "<ABAP_ClientNumber>";
      ```

   c. Save and activate the file.

      > **Note**
      >
      > By default, saving the modified file automatically commits the saved version to the repository; you do not need to commit the file before activating it.

7. View the activated HTTP destination.
   You can use the *SAP HANA XS Administration Tool* to check the contents of an HTTP destination configuration.

   > **Note**
   >
   > To make changes to the HTTP Destination configuration, you must use a text editor, save the changes and reactivate the file.

   a. Open a Web browser.

b. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > To access details of HTTP destinations in the *SAP HANA XS Administration Tool*, you must have a role based on the role template `sap.hana.xs.admin.roles::HTTPDestAdministrator`.

c. Locate the package containing the HTTP destination `<MyHTTPdestination>.xshttpdest`.

   Expand the nodes in the *Application Objects* pane to locate the package where the HTTP destination resides and select the HTTP destination to display details in the right pane.

8. Check the specified system ID (SID) and the client of the remote SAP system referenced in the HTTP destination.
   a. Enable the *SAP Assertion Ticket* radio button.
   b. Check (or enter) the SID and client number for the remote SAP system in the *SAP SID* and *SAP Client* text boxes respectively.
9. Save the changes to the HTTP destination and use it in an XSJS application service.

   > → Tip
   >
   > You can reference an HTTP destination from an XSJS service using the function `$.net.http.readDestination("<packageName>", "<HTTPDestinationName>")`

14.1.10 Maintaining User Self Service Tools

User self-service tools enable SAP HANA users to trigger account-related tasks, for example, the creation of a new database account.

By default, the user self-service tools are disabled. The SAP HANA administrator must activate the user self-service feature to provide users with access to embedded tools they can use to request the creation of a new user account in the SAP HANA database or request a new password.

Setting up and maintaining user-self-service tools for SAP HANA includes the following high-level tasks:

- Enable user self-service tools
- Request a new user account
- Display a list of the current user requests
- Reject a user/user request
- Enable access to the user-self-service administration tool

The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`

> ⓘ Note
>
> To log on, use the name and password of the user who has a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Enabling and maintaining the tools required to manage user self-service requests in SAP HANA involves the creation of a dedicated technical user and the assignment of dedicated roles.

- Administrator
  The user who manages the self-service requests and access lists must be assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`; the user self-service administrator is the same user as the user associated with the email address defined in the `xsengine.ini` parameter `sender_email`. The self-service administrator receives an e-mail in response to each self-service request; the e-mail contains a list of tasks to perform.
- Technical user
  A dedicated technical user, who is used to execute tasks associated with user self-service requests, for example, sending e-mails in response to user requests. Technical users cannot be used to log on to SAP HANA.

Related Information

Enable User Self-Service Tools [page 1203]
User Self-Service Roles [page 1205]

14.1.10.1 Enable User Self-Service Tools

User self-service tools are not enabled by default; they must be activated by the SAP HANA administrator.

Prerequisites

To enable user self-service tools in SAP HANA, you must have the following privileges:

- Access to SAP HANA as SAP HANA database administrator
- Access to specific features provided by the SAP HANA XS administration tools, which requires the privileges granted by the following roles:
  - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
  - `sap.hana.xs.admin.roles::SQLCCAdministrator`
  - `sap.hana.xs.admin.roles::SMTPDestAdministrator`
  - `sap.hana.xs.ide.roles::SecurityAdmin`
  - `sap.hana.xs.selfService.admin.roles::USSAdministrator` (to log on to the user self-service adminstration tool)
- Access to the following SAP HANA tools:
  - *SAP HANA XS Administration Tool*
  - *SAP HANA Web-based Development Workbench*
  - *SAP HANA USS Administration Tool* (user self-service administration tool)

Context

By default, SAP HANA user self-service tools are disabled; the tools are neither visible in the user interface nor configured in SAP HANA. To provide access to embedded tools that enable users to request the creation of a new user account in the SAP HANA database or set a new password, the SAP HANA administrator must activate and set up the user self-service feature.

Procedure

1. Configure the XSSQLCC technical user required to run the user self-service tools.

   A technical user is required to execute user self-service requests; the technical user must be granted a role based on the role template `sap.hana.xs.selfService.user.roles::USSExecutor` and associated with the XSSQLCC artifact `selfService.xssqlcc`.
2. Set the required user-self-service parameters in the `xsengine.ini` file.

   As part of the process of enabling user self-service tools in SAP HANA, you must set a number of configuration parameters, for example, to specify the email address to use when responding to user requests or enable support for password-reset services. The parameters must be set in the `user_self_service` section of the `xsengine.ini` file.

   > ⓘ Note
   >
   > If the section `user_self_service` does not already exist, the SAP HANA administrator must create it.
3. Configure the SMTP server that SAP HANA XS applications can use to send e-mails.

   An SMTP server is required to send automatic e-mails in response to the requests users make with SAP HANA user-self-service tools.

   > ⓘ Note
   >
   > You can configure only one SMTP server per SAP HANA XS server. If an SMTP server is already available, you can skip this step.
4. Configure access to the user self-service administration tools.

   You must assign a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator` to the user who requires access to the user-self-service administration tools. The user self-service administrator maintains user self-service requests and access blacklists and whitelists.

   > → Tip
   >
   > The user self-service administrator is the user who owns the e-mail address defined in the `sender_email` parameter in the `user_self_service` section of the `xsengine.ini` SAP HANA configuration file.

Related Information

Set up the Technical User for Self-Service Tools [page 1206]
Configure an SMTP Server for User Self-Service Tools [page 1207]
Configure Access to User-Self-Service Administration Tool [page 1209]

14.1.10.1.1 User Self-Service Roles

Dedicated roles are provided to enable access to and the administration of user-self-service tools.

User-self-service tools enable users to request basic database-account services using tools displayed in the user interface. For example, if the self-service tools are enabled, users can request the creation of a new account or a password reset if a password has been forgotten. Additional tools are provided to help administrate the user-self-service requests.

> → Recommendation
>
> As repository roles delivered with SAP HANA can change when a new version of the package is deployed, either do not use them directly but instead as a template for creating your own roles, or have a regular review process in place to verify that they still contain only privileges that are in line with your organization's security policy. Furthermore, if repository package privileges are granted by a role, we recommend that these privileges be restricted to your organization's packages rather than the complete repository. To do this, for each package privilege (`REPO.*`) that occurs in a role template and is granted on `.REPO_PACKAGE_ROOT`, check whether the privilege can and should be granted to a single package or a small number of specific packages rather than the full repository.

User Self-Service Roles

SAP HANA Role	Description
sap.hana.xs.selfService.user.roles::USSAdministrator	Role assigned to the user responsible for administrating the requests sent by users using self-service tools. For example, it provides access to the *USS Administration* tool, which enables the activation of users who request a new user account in the SAP HANA database and allows the user-self-service administrator to maintain self-service-specific blacklists for user requests, e-mail addresses, domains, and IP addresses. The *USS Administrator* role also provides access to the tools required to assign roles to (and activate) users in SAP HANA, for example: • System privileges: USER ADMIN • Object privileges: SELECT on the tables USERS (SYS) and USER_PARAMETERS (SYS)
sap.hana.xs.selfService.user.roles::USSExecutor	Role assigned to the technical user that will be used to respond to and execute user-self-service requests, for example, to create an new account or request a new password.

14.1.10.1.2 Set up the Technical User for Self-Service Tools

Configure the configuration connection (XSSQLCC) and the technical user which are required to execute user self-service requests.

Prerequisites

To complete the steps in this task, you must have the following privileges:

- Access to SAP HANA as the administrator
- Access to specific features provided by the *SAP HANA XS Administration Tool* and the *SAP HANA Web-based Development Workbench*, which requires roles based on the following role templates:
  - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
  - `sap.hana.xs.admin.roles::SQLCCAdministrator`
  - `sap.hana.xs.ide.roles::SecurityAdmin`

Context

A technical user is required to execute user self-service requests; the technical user must have a role based on the role template `sap.hana.xs.selfService.user.roles::USSExecutor` and associated with the design-time XSSQLCC artifact `selfService.xssqlcc`.

Procedure

1. Create the XSSQLCC technical user required to execute the user self-service requests.
   a. Open the *SAP HANA Web-based Development Workbench* and start the*Security* tool.

      The *Security* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/ide/security`.

      > **Note**
      >
      > Access to the *Security* tool in the *SAP HANA Web-based Development Workbench* requires a role based on the role template `sap.hana.xs.ide.roles::SecurityAdmin`.

   b. Right-click the node *Security* > *Users* and choose *New User*
   c. Specify the required details for the new technical user.

      You must provide a name and authentication credentials.
   d. Assign a role based on the role template `sap.hana.xs.selfService.user.roles::USSExecutor` to the new technical user.
   e. Save your changes to add the new technical user.
2. Assign the new technical user to the `selfService.xssqlcc` artifact.

The technical user you assign to the `selfService.xssqlcc` artifact executes all user-self-service requests, which requires a role based on the role template `sap.hana.xs.selfService.user.roles::USSExecutor`. The `selfService.xssqlcc` artifact provides the appropriate access to SAP HANA.

a. Start the Web-based *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > To edit `xssqlcc` artifacts with the *SAP HANA XS Administration Tool*, you must have roles based on the following role templates: `sap.hana.xs.admin.roles::RuntimeConfAdministrator` and `sap.hana.xs.admin.roles::SQLCCAdministrator`.

b. Locate the artifact SQL connection-configuration artifact `selfService.xssqlcc`.

   In the *Application Objects* screen, navigate to the package `/sap/hana/xs/selfService/user`.

c. Assign a technical user to the `selfService.xssqlcc` artifact.

   This is the technical user who will be used to execute all user-self-service requests. The user must be assigned a role based on the role template `sap.hana.xs.selfService.user.roles::USSExecutor`. You must provide the user name and the corresponding password.

Related Information

Enable User Self-Service Tools [page 1203]

14.1.10.1.3 Configure an SMTP Server for User Self-Service Tools

An SMTP server is required to enable SAP HANA to respond to user self-service requests.

Prerequisites

To complete the steps in this task, you must have the following privileges:

- Access to SAP HANA as the administrator
- Access to specific features provided by the *SAP HANA XS Administration Tool* and the *SAP HANA Web-based Development Workbench*, which requires roles based on the following role templates:
  - `sap.hana.xs.admin.roles::RuntimeConfAdministrator`
  - `sap.hana.xs.admin.roles::SMTPDestAdministrator`

Context

To enable SAP HANA to send automatic e-mails in response to the requests users make with SAP HANA user-self-service tools, you must configure a new SMTP server, or make SAP HANA aware of an existing SMTP server.

> ⓘ Note
>
> You can configure only one SMTP server per SAP HANA XS server. If an SMTP server is already configured, you can use the configured server; you do not have to complete this task.

Procedure

1. Start the Web-based *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > To access to the *SAP HANA XS Administration Tool*, you need a role based on the role template `sap.hana.xs.admin.roles::RuntimeConfAdministrator`.

2. Start the *SMTP Configuration* tool .

   > ⓘ Note
   >
   > To access to the *SMTP Configuration* tool in the *SAP HANA XS Administration Tool*, you need a role based on the delivered role `sap.hana.xs.admin.roles::SMTPDestAdministrator`.

3. Specify the details of the SMTP server that the user-self-service tools use to reply to service requests.

   You need to specify the fully qualified domain name of the SMTP server and the port to use for connections, for example, 25 (standard).

   > → Tip
   >
   > For more information about setting up an SMTP server, see *Related Links* below.

Related Information

Enable User Self-Service Tools [page 1203]
Maintaining SMTP Server Configurations [page 1176]

14.1.10.1.4 Configure Access to User-Self-Service Administration Tool

SAP HANA provides an administration tool that enables you to maintain user self-service requests.

Context

Access to the user-self-service administration tools is only possible to users with a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`. The user self-service administrator maintains user self-service requests and access-control blacklists and whitelists.

> → Tip
>
> The user self-service administrator is the user who owns the e-mail address defined in the `sender_email` parameter in the `user_self_service` section of the `xsengine.ini` SAP HANA configuration file.

Procedure

1. Open the *SAP HANA Web-based Development Workbench* and start the *Security* tool.

   The *Security* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/ide/security`.

   > ⓘ Note
   >
   > To access to the *Security* tool in the *SAP HANA Web-based Development Workbench*, you need a role based on the role template `sap.hana.xs.ide.roles::SecurityAdmin`.

2. Configure the user-self-service administrator.

   You can create a new user or assign a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator` to an existing user.

   a. In the *Security* tool, right-click the node *Security* › *Users* and choose the user for whom you want to enable access to the user-self-service administration tools.
   b. Assign a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator` to the selected user.
   c. Save your changes.

3. Log on to the user-self-service administration tool as the new user-self-service administrator.

   Verify that you have the permissions requried to access to the *USS Administration* tool; the tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`.

   > ⓘ Note
   >
   > To log on to the *USS Administration* tool, use the name and password of the user to whom you assigned the role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator` in the previous step.

Related Information

Enable User Self-Service Tools [page 1203]
Display all User Self-Service Requests [page 1213]
Maintain User Self-Service Access Lists [page 1222]

14.1.10.1.5 Maintain User Self-Service Initialization Parameters

Selected INI parameters can be used to configure how the USS tools respond to user requests and which actions are allowed by default.

Prerequisites

SAP HANA user roles are used to determine the level of access to the features provided by the SAP HANA XS administration tools. To access the tools required to maintain user self-service requests, you must have a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Context

As part of the process of enabling user self-service tools in SAP HANA, you must set a number of configuration parameters, for example, to activate the self-service tools, specify the email address to use when responding to user requests, or enable support for password-reset services. The parameters you maintain here are synchronized with the corresponding parameters in the `user_self_service` section of the `xsengine.ini` file for the SAP HANA system where you want make self-service tools available.

To display and maintain the initialization parameters for the user self-service tools, perform the following steps:

Procedure

1. Start the user-self-service administration tool.

   The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`

   > **Note**
   >
   > To log on, use the name and password of the user who has been assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

2. Display the list of user-self-service requests.

   When you open the *USS Administration* tool, the list of user-self-service requests is displayed by default.

   > **Note**
   >
   > You can also use the *USS Administration* tool to maintain access lists.

3. Choose the *INI Parameters* tool.
4. Set the initialization parameters as required.

   Some parameters are enabled (true) or disabled (false); other parameters require a value to be set, for example, a user's e-mail address or the maximum number of times a user can request a new account with USS tools.

   > **Note**
   >
   > The parameters you maintain here are synchronized with the corresponding parameters in the `user_self_service` section of the `xsengine.ini` file for the SAP HANA system where you want make self-service tools available.

Related Information

User Self-Service Initialization Parameters [page 1211]

14.1.10.1.5.1 User Self-Service Initialization Parameters

Initialization (INI) parameters can be used to configure which USS tools are enabled and how the USS tools react to user requests.

In the *USS Administration* tool, the *INI Parameters* tool displays the mandatory parameters that must be set to enable and configure user-self-service and, in some cases, specify how they can be used. The following table indicates which parameters must be set.

> **Note**
>
> The USS initialization parameters you set with USS administration tools correspond to (and are synchronized with) the SAP HANA parameters listed in the `user_self_service` section of the `xsengine.ini` configuration file.

USS INI Parameter Details

UI Element	Description	Parameter Name	Default
Automatic User Creation	Controls if a user creation request requires approval from user administration. In both cases the administrator has to assign roles to the new user. • *Disabled*: Requests for a new user account require administrator approval for account activation. • *Enabled*: The user is automatically created **and activated** as a restricted user.	`automatic_user_creation`	Disabled/False
Forgot Password	Defines if the system supports password recovery with user-self-service tools. The parameter controls not only the display of the *Forgot Password* button in the UI logon screen but also the enablement of the corresponding user-self-service backend services.	`forgot_password`	Disabled/False
Request New user	Enables system support for user-self-service tools. The parameter controls not only the display of the *Request New User* button in the UI logon screen but also the enablement of the corresponding user-self-service backend services.	`request_new_user`	Disabled/False
Reset Locked User	Enables support for a password reset for a locked user. Reset password will be forbidden for locked users if the value is *Disabled*.	`reset_locked_user`	Disabled/False
Sender E-Mail Address	The email address used for sending out auto-generated replies to user self-service requests, for example, `uss.admin@acme.com`. Ideally, this is the e-mail address used by the self-service administrator, who is assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator` and maintains self-service requests and access lists.	`sender_email`	None
Token Expiry Time	The time duration (in seconds) for which a generated token (and the corresponding request for a new user or password reset) is valid.	`token_expiry_time`	3600
User Creation Request Count	The number of times a user can use user-self-service tools to request a new user account. The user is determined by a combination of user name and e-mail address.	`user_creation_request_count`	3

Optional USS Parameters

It is possible to customize the background image displayed in the logon Web page, for example, by specifying the URL to the image displayed as background in the logon screen. However the following prerequisites apply:

- The image file specified in the URL must be reachable by http(s)

- The URL does not require authentication or authorization
- The recommended minimum resolution of the specified background image is: 1600*1200
- A technical user has to be assigned to the XSSQLCC artifact `/sap/hana/xs/selfService/user/selfService.xssqlcc`. The technical user must also be assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSExecutor`. This user will be used to query the details from the server.

> **Note**
>
> The parameter `login_screen_background_image` must be set in the `httpserver` section of the SAP HANA `xsengine.ini` configuration file and can only be set with SAP HANA studio tools.

Optional User Self-Service Configuration Parameters

Parameter Name	Section Name	Description	Example	Default
`login_screen_background_image`	httpserver	URL to the image displayed as background in the logon screen	/sap/hana/xs/ui/Image.jpg	None

Related Information

Maintain User Self-Service Initialization Parameters [page 1210]

14.1.10.2 Display all User Self-Service Requests

Display a list of all the user creation requests which have been sent using user self-service tools.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS administration tools. To access the tools required to maintain user self-service requests, you must have a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Context

The user-self-service administrator is the user associated with the e-mail address defined in the `xsengine.ini` parameter `sender_email`. The user-self-service administrator can use the *USS*

Administration tool to view a list of all the self-service requests received from users. Each user self-service request includes the following details:

- User name
- Creation date and time
- Number of pending self-service requests made by the same user

To display all user self-service requests, perform the following steps:

Procedure

1. Start the user-self-service administration tool.

   The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`.

   > ⓘ Note
   >
   > To log on, use the name and password of the user who has been assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

   When you open the *USS Administration* tool, the list of user-self-service requests is displayed by default.
2. Display the access-control list that you want to maintain.

   You can maintain access-control lists for the following conditions:
   - Domain
   - E-mail address
   - IP ranges
3. Maintain entries for the selected access-control list.

   You can use the *Add* and *Delete* buttons to manage the list entries.

   > → Tip
   >
   > To delete an entry from an access-control list, first check one or more items in the list and choose *Delete*.

Related Information

Display all User Self-Service Requests [page 1213]
Activate a User Account [page 1219]
Reject a User Self-Service Request [page 1220]

14.1.10.3 Request a New User Account

Request a new user account with user-self-service tools.

Prerequisites

- User-self-service tools are enabled in SAP HANA
- The required technical user (with the role *USSExecutor* is configured and available to respond to user-self-service requests

Context

If the self-service tools are enabled, a user can use the tools to request a new user account in the SAP HANA database. A valid e-mail address is required to complete the account-creation process, and the administrator must activate the new account and assign user roles and privileges.

To request a new database account in SAP HANA, a user must perform the following steps:

Procedure

1. Logon to SAP HANA using the Web-based interface.

   The SAP HANA *Logon* screen is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/formLogin/login.html`.
2. Request a new user SAP HANA account.

   Click the *Request Account* link in the bottom right-hand side of the logon screen.
3. Specify basic details for the new user.

   In the *Request Account* screen, you must supply a name and a valid e-mail address, which the user-self-service tools use to respond to the request.
   a. Enter a name for the new database user.
   b. Enter a valid e-mail address for the new database user.

      The e-mail address is used to sent the user messages with links to use to start the account activation process.
4. Submit the request for a new account.
   After submitting the account-creation request, the user receives the following automatically generated e-mails:
   - Address verification
     An e-mail with a link that verifies the target e-mail address
   - User-self-service request administratration
     An e-mail that contains the following links:

- Open the *SAP HANA XS Administration Tool* tool that enables an account be set up and activated for the new user
- Display a list of all pending user-self-service requests

5. Set a password and security question for the new user account

   The user requesting the new database account must set a password and choose a security question that is used in the event of a forgotten-password request.An answer must be supplied for the selected security question.

6. Activate the new user account.

   The user self-service administrator must activate the new user account to enable the new user to log on to SAP HANA. Activation involves assigning roles to the new user as well as privileges, for example: objects, application, package.

14.1.10.4 Maintain Your User Profile

Each user account is associated with a profile; the user who owns the profile must adjust the settings to suit personal preferences.

Prerequisites

- User-self-service tools are enabled in SAP HANA.
- A user profile exists; a user profile is created automatically on activation of a user account in SAP HANA.
- The profile owner has the privileges granted by the role `sap.hana.xs.formLogin.profile::ProfileOwner`.

Context

When a new user account is activated, the corresponding account profile is created with default settings. The new user must log on to SAP HANA and adjust some of the default settings, for example, the default password. It is also mandatory to choose a security question and set the corresponding answer.

Procedure

1. Log on to SAP HANA using the Web-based interface.

   The SAP HANA *Logon* screen is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/formLogin/login.html`.

2. Start the profile manager.

   The *Manage Profile* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/formLogin/profile/`.

3. Maintain your security settings.

   It is mandatory to choose a security question from the drop-down list and provide a corresponding answer. You can also change the e-mail address to use for communication.

   > **Note**
   >
   > The question and answer are used to confirm logon credentials, whenever the owner of the profile attempts to make any changes to the profile.

4. Change the initial (default) password.

   > **Note**
   >
   > In SAP HANA, rules apply that restrict which characters you can use in the password you set.

5. Maintain your profile preferences.

   You can change set the language locale for your account and set preferences for the way that the date and time is displayed, for example:

   - *Date Format*: *YYYY-MM-DD* (“2014-12-25”)
   - *Time Format*:
     - *HH24:MI* (“15:30”)
     - *HH12:MI* (“3:30pm”)
   - *Locale*: *English (en)*

   > **Note**
   >
   > Application developers need to ensure that the applications they create are able to take account of the preference set in a user's profile.

Related Information

Request a New User Account [page 1215]

14.1.10.4.1 User Profile Details

Each user account has a corresponding account profile.

When a new user account is activated, the corresponding account profile is created with default settings. The new user must log on to SAP HANA and adjust some of the default settings, for example, the default password. It is also mandatory to choose a security question and set the corresponding answer. The *User Self Services Manage Profile* tool displays the following screens to help you maintain details of the SAML service provider:

- Security Settings [page 1218]
- Preferences [page 1218]
- Change Password [page 1218]

Security Settings

The *Security Settings* screen area in the USS *Manage Profile* tool enables you to maintain details of the security settings for your SAP HANA user account. The following table indicates which details can be maintained.

UI Element	Description	Example
Email Address	The e-mail address of the user to whom the account and profile belong. USS notifications are sent to the specified address.	Kwame.Ampomah@acme.com
Security Question	The security question to ask when you make any changes to the user profile details.	What is your favorite sport?
Security Answer	Text string that you use an the answer to the security question	squash

Preferences

The *Preferences* screen area in the USS *Manage Profile* tool enables you to maintain details of the display preferences for your SAP HANA user account. The following table indicates which details can be maintained.

UI Element	Description	Example
Date Format	The way in which the date is displayed in the applications you use, for example, *2014-12-25*	YYYY-MM-DD
Time Format	The way in which the time is displayed in the applications you use, for example, *15:30* (HH24:MI) or *3:30pm* (HH12:MI)	HH24:MI
Locale	The language environment and settings to apply for the applications you use	English (en) or Chinese (zh)

Change Password

The *Change Password* screen area in the USS *Manage Profile* tool enables you to maintain details of your SAP HANA user account. The following table indicates which details can be maintained.

UI Element	Description	Example
Old Password	The initial password assignen when the account was activated or, if changed, the currently valid password	*****
New Password	The news password	*****
Repeat Password	Confirm the new password you entered in *New Password*	*****

Related Information

Maintain Your User Profile [page 1216]

14.1.10.5 Activate a User Account

Enable a new user account in the SAP HANA database in response to a user self-service request.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS administration tools. To access the tools required to maintain user self-service requests, you must have a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Context

When a user requests a new account, the user account is created but disabled by default. The user who sent the request cannot use the account to log on to SAP HANA until the SAP HANA user-self service administrator activates the account. The self-service administrator must manually activate the account and assign the necessary roles, too.

> **Note**
>
> On activation of the new user account, an e-mail is automatically sent to the user containing the security token required to enable the new user to set a password for the new account.

To activate a new account in response to a user self-service request, perform the following steps:

Procedure

1. Start the user-self-service administration tool.

   The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`.

   > **Note**
   >
   > To log on, use the name and password of the user who has been assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

2. Display the list of user-self-service requests.

   When you open the *USS Administration* tool, the list of user-self-service requests is displayed by default.

> **Note**
>
> You can also use the *USS Administration* tool to maintain access lists.

3. Assign roles to a new user.
   a. In the *Username* column of the *User Self Service Requests* screen, check the box next the user you want to activate.
   b. In the *Administration* column, choose *Assign Roles*.

      The link opens the *Security* tool in the SAP HANA Web-based Development Workbench and displays the selected user. Select the appropriate roles to assign to the new user from the list of roles displayed.

      > **Note**
      >
      > To help decide which roles are appropriate for the user request, use the path indicated in the *Request Origin* column to see which tool the user is trying to access. For example, `/sap/hana/ide/editor` is the SAP HANA Editor tool, which requires a role based on the role template `sap.hana.ide.roles::EditorDeveloper`.

4. Activate the selected new user.

   In the *User Self Service Requests* page, choose *Activate and Notify* to send an e-mail to the corresponding user indicating that the requested account is active and ready for use.

Related Information

Reject a User Self-Service Request [page 1220]
Maintain User Self-Service Access Lists [page 1222]
Display all User Self-Service Requests [page 1213]

14.1.10.6 Reject a User Self-Service Request

Refuse a self-service request to create a new user account in the SAP HANA database.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS administration tools. To access the tools required to maintain user self-service requests, you must have a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Context

When a user requests a new account, the user account is created but disabled by default. The user who sent the request cannot use the account to log on to SAP HANA until the SAP HANA user-self service administrator

activates the account and assign the appropriate roles. The user-self-service administrator can also choose to reject the request for a new user account in the SAP HANA database, for example, by adding the user to the user-requests blacklist.

> **Note**
>
> On activation of the new user account, an e-mail is automatically sent to the user containing the security token required to enable the new user to set a password for the new account.

To refuse a self-service request to create a new user account in the SAP HANA database, perform the following steps:

Procedure

1. Start the user-self-service administration tool.

   The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`.

   > **Note**
   >
   > To log on, use the name and password of the user who has been assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

2. Display the list of user-self-service requests.

   When you open the *USS Administration* tool, the list of user-self-service requests is displayed by default.

   > **Note**
   >
   > You can also use the *USS Administration* tool to maintain access lists.

3. Reject the user's request for a new database account.
   a. In the *Username* column of the *User Self Service Requests* screen, check the box next the user, whose request for a new account you want to reject.
   b. Choose *Add to blacklist* in the bottom right-hand corner of the screen.

      The link opens the *Security* tool in the SAP HANA Web-based Development Workbench and displays the selected user. Select the appropriate roles to assign to the new user from the list of roles displayed.
4. Check the rejected user has been added to the user-requests blacklist.

Related Information

Maintain User Self-Service Access Lists [page 1222]
Request a New User Account [page 1215]

14.1.10.7 Maintain User Self-Service Access Lists

Access to self-service tools can be controlled using denylists and allowlists, for example, for email addresses.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS administration tools. To access the tools required to maintain user self-service requests, you must have a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Context

The user self-service administrator can control access to self-service features by maintaining denylists and allowlists for the following areas:

- User requests
- Network domains
- IP addresses
- E-mail addresses
- DB users

> ### Note
>
> Users whose requests exceed the value set in the `xsengine.ini` parameter `user_creation_request_count` are no longer able to submit any requests. If necessary, the administrator can add such users to the access denylist.

To display all user self-service access lists, perform the following steps:

Procedure

1. Start the user-self-service administration tool.

   The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`.

   > ### Note
   >
   > To log on, use the name and password of the user who has been assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

2. Display the list of user-self-service requests.

   When you open the *USS Administration* tool, the list of user-self-service requests is displayed by default.

> **Note**
>
> You can also use the *USS Administration* tool to maintain access lists.

Related Information

User Self-Service Access Lists [page 1223]

14.1.10.7.1 User Self-Service Access Lists

Access to self-service features is controlled by blacklists and whitelists.

The user self-service administrator can control access to user-self-service tools using the *USS Administration*by maintainging access lists. The access lists included with the *USS Administration* are described in the following table.

User Self-Service Access List Details

List Name	Description
User Requests	A list of all pending requests sent with user-self-service tools, including the name of the user who sent the request and the corresponding e-mail address. Users who have more requests than the value set in the `xsengine.ini` parameter `user_creation_request_count` are automatically added to the access blacklist.
Network Domains	A list of network domains, which can be used to permit or deny user self-service requests from one or more specific domains, for example, "`acme.com`". If a user self-service request for a new user account arrives from a user with an e-mail address associated with a whitelisted domain, the new user account is created as a **restricted** user and activated without requiring any administrator intervention. Users on the domain black list are no longer permitted to create a user self-service request.
IP Addresses	A list of IP addresses (or names), which can be used to permit or deny user self-service requests from one or more specific IP addresses, for example, "`*.122.10`". The same rules for blacklists and whitelists apply as for network domains above.
E-mail addresses	A list of e-mail addresses, which can be used to permit or deny user self-service requests from a specific e-mail address, for example, "`joe.doe@acme.com`" or "`jane.doe@acme.com`". The same rules for blacklists and whitelists apply as for network domains and IP addresses above.

List Name	Description
DB Users	The names of the database users who are not allowed to change their respective SAP HANA password using USS reset-password tools, for example, `joedoe` or `janedoe`. The following additional restrictions apply: • By default, it is not possible to use USS tools to reset the password for the SYSTEM user. • The USS administrator cannot add to the *DB Users* list any user who logs on to SAP HANA with single sign-on (SSO) credentials. • Users who log on to SAP HANA with SSO credentials cannot use USS tools to reset their password.

14.1.10.8 Maintain User Self-Service E-Mail Templates

Default templates enable you to format the contents of the auto-generated e-mails sent when user self-service (USS) tools are employed to request a new account in SAP HANA or recovery a forgotten password.

Prerequisites

SAP HANA uses roles to determine the level of access to the features provided by the SAP HANA XS administration tools. To access the tools required to maintain user self-service requests, you must have a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

Context

The user self-service administrator can modify the contents of the automatically generated e-mails that are sent to users during the USS account-creation process. Templates exist for the responses to the following actions: user requests, account activation, and forgotten passwords.

To display and maintain the current e-mail templates for user self-service features, perform the following steps:

Procedure

1. Start the user-self-service administration tool.

   The *USS Administration* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/selfService/admin`

   > **Note**
   >
   > To log on, use the name and password of the user who has been assigned a role based on the role template `sap.hana.xs.selfService.admin.roles::USSAdministrator`.

2. Display the list of user-self-service requests.

   When you open the *USS Administration* tool, the list of user-self-service requests is displayed by default.

   > **Note**
   >
   > You can also use the *USS Administration* tool to maintain access lists.

3. Choose the *Email Templates* tool.
4. Choose the required e-mail template.

   The following templates are available for automatically generated e-mails:

   - User request
     E-mail sent in response to a user self-service request for a new SAP HANA user account
   - User activation
     E-mail sent when a new SAP HANA user account has been activated
   - Password Recovery
     E-mail sent in response to a user self-service request to set a new SAP HANA password, for example, because the user has forgotten the current password.

Related Information

User Self-Service E-Mail Templates [page 1225]

14.1.10.8.1 User Self-Service E-Mail Templates

USS provides templates that can be used to format the content of auto-generated e-mails.

In the *USS Administration* tool, the *Email Templates* tool displays information about the templates used to format the content of auto-generated e-mails that are used during the process of creating a new SAP HANA user account. You can use the *Email Templates* tab to maintain the following details:

- User Request [page 1225]
- User Activation [page 1226]
- Forgot Password [page 1226]

User Request

The *User Request* tab in the *Email Templates* tool enables you to maintain templates that are used to generate the e-mails sent in response to a user request to create a new account in SAP HANA; the e-mails are sent to the USS administrator and the user who submitted a USS request. The following table indicates which information can be viewed and modified.

User Request E-Mail Template Details

UI Element	Description	Example
To	The email address of the USS administrator	admin.uss@acme.com
Subject	The text you want to appear in the e-mail's *Subject* box	New user account
Body	The text of the e-mail sent either to the USS admin indicating that a new request for a SAP HANA user account has been received and needs attention or to the user who subbmited a request and indicating that the request for a new account has been received and is being processed	Dear USS Admin, ...

User Activation

The *User Activation* tab in the *Email Templates* tool enables you to maintain templates for the account-activation e-mails sent to the user who uses USS tools to submits a request for a new account in SAP HANA; the e-mail informs the user that the requested account is active and can be used to log on to SAP HANA.. The following table indicates which information can be viewed and modified.

User Activation E-Mail Template Details

UI Element	Description	Example
To	The email address of the user whose new accounts has been activated	admin.uss@acme.com
Subject	The text you want to appear in the e-mail's *Subject* box	SAP HANA account status
Body	The text of the e-mail sent either to the new SAP HANA user indicating that an account has been activated and can be used to log on to SAP HANA	Dear [`<User Name>`], ...

Forgot Password

The *Forgot Password* tab in the *Email Templates* tool enables you to maintain the template used to generate e-mails that are sent to SAP HANA users who submit a USS request to reset a password. The following table indicates which information can be viewed and modified.

Forgot Password E-Mail Template Details

UI Element	Description	Example
To	The email address of the user who submitted a request to reset a password	jane.doe@acme.com
Subject	The text you want to appear in the e-mail's *Subject* box	Reset account password
Body	The text of the e-mail to the SAP HANA user indicating that a request to reset an SAP HANA password has been received and action is required from the user	Dear [`<User Name>`], ...

Related Information

Maintain User Self-Service E-Mail Templates [page 1224]

14.1.11 Scheduling XS Jobs

Scheduled jobs define recurring tasks that run in the background. The JavaScript API `$.jobs` allows developers to add and remove schedules from such jobs.

If you want to define a recurring task, one that runs at a scheduled interval, you can specify details of the job in a `.xsjob` file. The time schedule is configured using `cron`-like syntax. You can use the job defined in an `.xsjob` file to run an XS Javascript or SQLScript at regular intervals. To create and enable a recurring task using the xsjob feature, you perform the following high-level tasks:

> **Note**
>
> The tasks required to set up a scheduled job in SAP HANA XS are performed by two distinct user roles: the application developer and the SAP HANA administrator. In addition, to maintain details of an XS job in the *SAP HANA XS Administration Tool*, the administrator user requires the privileges granted by the role template `sap.hana.xs.admin.roles::JobAdministrator`.

Setting up Scheduled Jobs in SAP HANA XS.

Step	Task	User Role	Tool
1	Create the function or script you want to run at regular intervals	Application developer	Text editor
2	Create the job file `.xsjob` that defines details of the recurring task	Application developer	Text editor
3	Maintain the corresponding runtime configuration for the xsjob	SAP HANA administrator	XS Job Dashboard
4	Enable the job-scheduling feature in SAP HANA XS	SAP HANA administrator	XS Job Dashboard
5	Check the job logs to ensure the job is running according to schedule.	SAP HANA administrator	XS Job Dashboard

Related Information

The XSJob File [page 1240]
Tutorial: Schedule an XS Job [page 1236]
XS Job File Keyword Options

14.1.11.1 Maintain XS Job Details

XS job schedules are defined by developers; the XS job-scheduling feature must be set up by a system administrator.

Prerequisites

To enable the XS Job schedule feature in SAP HANA XS, the following prerequisites apply:

- You have administrator access to an SAP HANA system.
- You have been granted a role based on the role template `sap.hana.xs.admin.roles::JobAdministrator`.

  > → Tip
  >
  > For more information about XS classic roles, see *Related Information* below

- You have access to the XS Job Dashboard, which requires the role `JobSchedulerAdministrator`.
- An XS job file that has been activated in the repository.

Context

To enable developers to define and deploy job schedules using the XS job feature, the system administrator must first set up the environment and enable some essential options.

Procedure

1. Enable the job-scheduling feature in SAP HANA XS.

   This step requires the permissions granted to the SAP HANA administrator.

   > ⓘ Note
   >
   > It is not possible to enable the scheduler for more than one host in a distributed SAP HANA XS landscape.

   a. In the *XS Job Dashboard* set the *Scheduler Enabled* toggle button to **YES**.

   Toggling the setting for the *Scheduler Enabled* button in the *XS Job Dashboard* also changes the current value of the SAP HANA configuration variable *xsengine.ini* > *scheduler* > *enabled*, which is set in the *Configuration* tab of the SAP HANA studio's *Administration* perspective.
2. Maintain the XS job's runtime configuration.

   a. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

b. Open the *XS Job Dashboard*.

> **Note**
>
> Access to the XS Job Dashboard requires the role `JobSchedulerAdministrator`. To maintain details of an XS job using the Web-based *XS Administration Tool* you need the privileges granted in the SAP HANA user role `sap.hana.xs.admin.roles::JobAdministrator`.

c. Maintain the details of the XS job.

In the *Job Details* tab, select the XS Job whose details you want to maintain. In the *Configuration* tab, you need to specify the following details:

- *User*
  The user account in which the `xscron` job runs, for example, **`SYSTEM`**
- *Password*
  For security reasons, you must provide a password for the specified user.

  > **Note**
  >
  > If you do not provide a user password, you cannot save the changes to the XS Job object's runtime configuration.

- *Locale*
  The language encoding required for the locale in which the `xscron` job runs, for example, **`en_US`**
- *Start/Stop time*
  An optional value to set time during which the `xscron` job runs. You must enter the values using the syntax used for the SAP HANA data type `LocalDate` and `LocalTime`, for example, **`2013-11-05 00:30:00`** (thirty minutes past midnight on the 5th of November 2013.
- *Active*
  Enable or disable the job schedule

d. Save the job.

Choose *Save Job* to save and activate the changes to the job schedule.

3. Check the logs to ensure the job is running according to schedule.

You can view the list of `xsjob` schedules in the *Job Details* tab of the *XS Job Details* window. The information displayed includes the XS cron setup that defines the schedule, the current status of the job schedule, as well as the start and finish times.

Related Information

The XS Job Dashboard [page 1230]
The XS Job File [page 1240]
SAP HANA XS Classic Administration Roles [page 1137]

14.1.11.1.1 The XS Job Dashboard

The *XS Job Dashboard* is the central point of control for monitoring and maintaining job schedules that have been defined using the XS Job syntax.

The *XS Job Dashboard* displays details of the currently active job schedules that have been configured for the selected SAP HANA system, for example, using XS job files.The XS job file uses a cron-like syntax to specify the schedule at which the service defined in an XS JavaScript or SQLScript must run. You can use the *Scheduler Enabled* button in the *XS Job Dashboard* to enable schedules for all XS jobs globally.

> ⓘ Note
>
> Toggling the setting for the *Scheduler Enabled* button also changes the current value of the SAP HANA configuration variable *xsengine.ini* > *scheduler* > *enabled*, which is set in the *Configuration* tab of the SAP HANA studio's *Administration* perspective.

For each XS job displayed in the *XS Job Dashboard*,you can see the following details:

> → Tip
>
> Access to the XS Job Dashboard tool is only available to users who have been assigned the XS classic administrator role "`JobSchedulerAdministrator`". For more information about XS classic administration roles, see *Related Information* below.

- *Name*
  The name of the XS Job; this is name of the design-time artifact in the SAP HANA repository, for example, `MyJob.xsjob`
- *Package*
  The name of the repository package that contains the XS Job
- *User*
  The name of the user whose database account is used to run the XS Job schedule
- *Status*
  The current status of the XS job schedule, for example, *ACTIVE/INACTIVE*; you can change the status in the *XS Job Details* screen
- *Start/Stop time*
  An optional value to set the period of time during which the job runs. You must enter the values using the syntax used for the SAP HANA data type `LocalDate` and `LocalTime`, for example, **`2014-11-05 00:30:00`** (thirty minutes past midnight on the 5th of November 2014.
- *Session Timeout(s)*
  The number of times that the scheduled job run encountered a session timeout
- *Last Run Status*
  The status of the scheduled job when it last ran, for example: *Success*, *Error*, or *Running*

Related Information

Maintain XS Job Details [page 1228]
SAP HANA XS Classic Administration Roles [page 1137]

Tutorial: Schedule an XS Job [page 1236]

14.1.11.1.2 XS Job Details

Details of the runtime configuration of XS Job schedules and the XS jobs the schedules are used to manage.

In the *XS Job Dashboard*, the *XS Job Details* tab displays information about the currently active job schedules that have been configured for the selected SAP HANA system and the corresponding XS job files. You can use the *XS Job Details* tab to maintain the following details of the XS Jobs' runtime configuration:

- General Job Details [page 1231]
- Runtime Configuration [page 1232]
- Log Cleanup [page 1233]

Job Details

The *Job Details* tab in the *XS Job Details* tool enables you to view details of the XS Jobs that you have defined and scheduled to run, for example, the name of the XS job and a short description. The following table indicates which information can be viewed.

> **Note**
>
> The details displayed are defined in the design-time artifact that describes the selected XS Job.

Job Details

UI Element	Description	Example
Name	Text string used to specify the name (including full repository path) of the XS Job scheduled to run.	sap.hana.testtools::schedule
Description	A short description of the XS job defined in *Name*	Run XSUnit
Action	Text string used to specify the path to the function to be called as part of.the XS Job defined in *Name*	sap.hana.testtools:TestRunner.xsjs::run

Runtime schedules for XS Jobs contain the following details.

> **Note**
>
> Some of the values described (for example, *Origin* or *Changed ...*) are read only; it is not possible to modify them.

Schedules

UI Element	Description	Example
ID	The ID allocated to the job schedule	3
XSCron	The schedule for the specified task (defined in the "action" keyword); the schedule is defined using cron-like syntax.	2015 * * fri 12 0 0

UI Element	Description	Example
Parameter	A value to be used during the action operation. You can add as many parameters as you like as long as they are mapped to a parameter in the function itself.	Depends on job
Planned Time	The time at which an XS job is expected to run; if it does not run as planned, it is added to the job queue.	2014-11-05 00:30:00
Status	Indicates if the schedule is active or inactive	Active
Start Time	An optional value signifying the beginning of the period of time (schedule) during which the XS job runs	2014-11-05 00:30:00
Finish Time	An optional value signifying the end of the period of time (schedule) during which the XS job runs	2014-11-12 00:30:00
Time Taken (s)	The amount of time taken (in seconds) for the job/action to complete	5
Description	A short (**optional**) description of the XS job schedule. → Tip It is not possible to show the *Description* of a deleted job schedule, even if a description was defined when configuring the original job schedule. If the *Description* field in the job *Logs* view is empty, either no description was provided for the corresponding job schedule or the job schedule has been deleted. If no description was provided when configuring a job schedule, the *Description* field in the *Job Details* tab **always** remains empty.	gfn test schedule
Origin	The type of object used to define the schedule: *DESIGNTIME* (repository artifact) or *RUNTIME* (catalog object).	DESIGNTIME
Changed By	Name of the SAP HANA user who added or changed the XS job schedule	johndoe
Changed At	Time at which the schedule was changed	2015-01-30 14:19:59

Configuration

The *Configuration* tab in the *XS Job Details* tool enables you to maintain details of the runtime configuration for XS Jobs that you have scheduled to run. The following table indicates which information can be maintained.

XS Job Configuration

UI Element	Description	Example
User	The user account in which the xscron job runs.	SYSTEM
Password	Password for the specified *user*	****
Locale	The language encoding required for the locale in which the xscron job runs	en_US

UI Element	Description	Example
Start Time	Start time for the XS Job using the syntax required by the SAP HANA data type `LocalDate` and `LocalTime`	2013-11-05 00:30:00
End Time	End time for the XS Job using the syntax required by the SAP HANA data type `LocalDate` and `LocalTime`	2013-11-05 00:30:00
Session Timeout	Time in seconds for which the session is valid	0
Active	Indicates if the schedule is active or inactive	Active

Log Cleanup

The *Log Cleanup* tab in the *XS Job Details* tool enables you to create an XS Job that cleans up the logs of all XS Job currently running in the system. You can also create one schedule for each job in the system and allow users to configure the schedule in the *Job Details* dialog.

By default, XS Job logs are not cleaned up; no logs or log entries are deleted. If a cleanup of XS Job logs is required, the parameters can be set so that only those job-log entries for an XSJob that are older than N days are deleted, where N can be configured as a job parameter. Users can also specify the frequency of the cleanup schedule. The following table indicates which information can be maintained.

> ⚠ Restriction
>
> To enable or disable the cleanup of XS Job logs, you need the permissions assigned to the *JobAdministrator* role.

XS Job Log Cleanup

UI Element	Description	Example
Enabled	Enable the log-cleanup schedule	Yes
XSCron	The schedule for the specified XS Job log-cleanup task; the schedule is defined using cron-like syntax. In this example, the cleanup is scheduled to run every last Sunday of the month at 09:00 hours. (9am)	* * * -1.sun 9 0 0
Day	The number of days for which logs are retained (not cleaned up). For example, 1 retains all XS job logs from the day before the schedule starts and deletes all job logs that are two days old or older.	1

Related Information

SAP HANA XS Classic Administration Roles [page 1137]
Scheduling XS Jobs [page 1227]

14.1.11.2 Clean up XS Job Logs

Clean up the log entries generated in the SAP HANA database by the XS jobs that are running in the SAP HANA system.

Prerequisites

To enable the XS Job schedule feature in SAP HANA XS, the following prerequisites apply:

- You have administrator access to an SAP HANA system.
- You have been granted a role based on the role template `sap.hana.xs.admin.roles::JobAdministrator`.
- You have enabled the job-scheduling feature in SAP HANA XS.
- You have maintained details of the XS Job whose log entries you want to clean up.
- You have enabled the XS Job *sap.hana.xs.admin.jobs.server.common::cleanJobLog* that is used to clean up job-log entries
- You have activated the SQLCC artifact *sap.hana.xs.admin.jobs.server.common::cleanJobLog.xssqlcc* that is used by the cleanup job; this artifact creates a connection to SAP HANA with the *JobLogAdmin* privileges required to remove entries from the XS-job log (as defined in *cleanJobLog*)

> **Note**
>
> To activate the SQLCC artifact *sap.hana.xs.admin.jobs.server.common::cleanJobLog.xssqlcc*, you need the privileges provided with the XS admin role `sap.hana.xs.admin.roles::SQLCCAdministrator`.

Context

XS jobs write their logs to the table `_sys_xs.job_log` in the SAP HANA database. Since this table can grow in size very quickly, as more and more jobs and schedules are created, it is recommended to clean up the old job log entries. You can set up an XS Job that runs at a defined schedule and deletes all old log file entries for a particular XS job from the SAP HANA XS job-log table.

Procedure

1. Maintain the XS job's runtime configuration.
   a. Start the *SAP HANA XS Administration Tool*.

      The *SAP HANA XS Administration Tool* is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.
   b. Open the *XS Job Dashboard*.

> **Note**
>
> To maintain details of an XS job using the Web-based *XS Administration Tool* you need the privileges granted in the SAP HANA user role *sap.hana.xs.admin.roles::JobAdministrator*.

2. Configure details of the XS job schedule.

   In the *Job Details* tab, select the XS Job whose details you want to maintain. In the *Configuration* tab, you need to specify the following details:

   - *User*
     The user account in which the `xscron` job runs, for example, **`SYSTEM`**
   - *Password*
     For security reasons, you must provide a password for the specified user.

     > **Note**
     >
     > If you do not provide a user password, you cannot save the changes to the XS Job object's run-time configuration.

   - *Locale*
     The language encoding required for the locale in which the `xscron` job runs, for example, **`en_US`**
   - *Start/Stop time*
     An optional value to set time during which the `xscron` job runs. You must enter the values using the syntax used for the SAP HANA data type `LocalDate` and `LocalTime`, for example, **`2013-11-05 00:30:00`** (thirty minutes past midnight on the 5th of November 2013.
   - *Active*
     Enable or disable the job schedule.

3. Ensure that the old log entries written by the XS job are cleaned up.

   To enable a scheduled clean up of log entries in the SAP HANA database, you need to set up the following details:

   - *Enabled*
     Set the status of the job schedule used to clean up the XS job-related log entries
   - *XSCron*
     Define the schedule using XS cron syntax (year, month, day, day of the week, hour, minute, second) at which the cleanup job runs.
     `* * * -1.sun 9 0 0`
     This example runs the job on the last Sunday of every month at 9am.
   - *Day*
     Specify the number of days for which log entries should be **retained**. For example, to delete all log entries that are older than two days, enter the value "**`2`**".

4. Save the job.

   Choose *Save Job* to save and activate the changes to the job schedule.

5. Check the status of the new job and schedule.

   You can view the list of `xsjob` schedules in the *Job Details* tab of the *XS Job Details* window. The information displayed includes the XS cron setup that defines the schedule, the current status of the job schedule, as well as the start and finish times.

6. Check the logs to ensure the job is running according to schedule.

> → Tip
>
> If the *Description* field for a specific job log is empty in the *View Logs* list, this is typically an indication that either no description was defined for the job or the log has been deleted as part of the cleanup operation.

Related Information

Maintain XS Job Details [page 1228]
The XS Job Dashboard [page 1230]
XS Job Details [page 1231]

14.1.11.3 Tutorial: Schedule an XS Job

The `xsjob` file enables you to run a service (for example, an XS JavaScript or an SQLScript) at a scheduled interval.

Prerequisites

- You have access to an SAP HANA system.
- You have access to the XS Job Dashboard, which requires the role `JobSchedulerAdministrator`
- You have a role based on the role template `sap.hana.xs.admin.roles::JobAdministrator`.
- You have a role based on the role template `sap.hana.xs.admin.roles::HTTPDestAdministrator`.

> ⓘ Note
>
> This tutorial combines tasks that are typically performed by two different roles: the application developer and the database administrator. The developer would not normally require the privileges granted to the `sap.hana.xs.admin.roles::JobAdministrator` role, the `sap.hana.xs.admin.roles::HTTPDestAdministrator` role, or the SAP HANA administrator.

Context

In this tutorial, you learn how to schedule a job that triggers an XS JavaScript application that reads the latest value of a share price from a public financial service available on the Internet. You also see how to check that the XS job is working and running on schedule.

To schedule an XS job to trigger an XS JavaScript to run at a specified interval, perform the following steps:

Procedure

1. Create the application package structure that contains the artifacts you create and maintain in this tutorial.

   Create a root package called `yahoo`. You use the new `yahoo` package to contain the files and artifacts required to complete this tutorial.

   ```
   /yahoo/
          .xsapp               // application descriptor
          yahoo.xsjob          // job schedule definition
          yahoo.xshttpdest     // HTTP destination details
          yahoo.xsjs           // Script to run on schedule
   ```

2. Write the XS JavaScript code that you want to run at the interval defined in an XS job schedule.

   The following XS JavaScript connects to a public financial service on the Internet to check and download the latest prices for stocks and shares.

   Create an XS JavaScript file called `yahoo.xsjs` and add the code shown in the following example:

   ```
   function readStock(input) {
       var stock = input.stock;

       var dest = $.net.http.readDestination("yahoo", "yahoo");
       var client = new $.net.http.Client();
       var req = new $.web.WebRequest($.net.http.GET, "/d/quotes.csv?f=a&s=" +
   stock);
       client.request(req, dest);
       var response = client.getResponse();
       var stockValue;
       if(response.body)
           stockValue = parseInt(response.body.asString(), 10);
       var sql = "INSERT INTO stock_values VALUES (NOW(), ?)";
       var conn = $.db.getConnection();
       var pstmt = conn.prepareStatement(sql);
       pstmt.setDouble(1, stockValue);
       pstmt.execute();
       conn.commit();
       conn.close();
   }
   ```

   Save and activate the changes in the SAP HANA Repository.

   > ### Note
   >
   > Saving a file in a shared project automatically commits the saved version of the file to the repository, To explicitly commit a file to the repository, right-click the file (or the project containing the file) and choose *Team* > *Commit* from the context-sensitive popup menu.

3. Create an HTTP destination file using the wizard to provide access to the external service (via an outbound connection).

   Since the financial service used in this tutorial is hosted on an external server, you must create an HTTP destination file, which provides details of the server, for example, the server name and the port to use for HTTP access.

> **Note**
>
> To maintain the runtime configuration details using the Web-based *XS Administration Tool* you need the privileges granted in the SAP HANA user role *sap.hana.xs.admin.roles::HTTPDestAdministrator*.

Create a file called `yahoo.xshttpdest` and add the following content:

```
host = "download.finance.yahoo.com";
port = 80;
```

Save and activate the changes in the SAP HANA Repository.

4. Create the XS job file using the wizard to define the details of the schedule at which the job runs.

   The XS job file uses a `cron`-like syntax to define the schedule at which the XS JavaScript must run. This job file triggers the script `yahoo.xsjs` on the 59th second of every minute and provides the name "SAP.DE" as the parameter for the stock value to check.

   Create a file called `yahoo.xsjob` and add the following code:

```
{
    "description": "Read stock value",
    "action": "yahoo:yahoo.xsjs::readStock",
    "schedules": [
       {
          "description": "Read current stock value",
          "xscron": "* * * * * * 59",
          "parameter": {
             "stock": "SAP.DE"
             }
       }
    ]
}
```

   Save and activate the changes in the SAP HANA Repository.

5. Maintain the XS job's runtime configuration.

   You maintain details of an XS Job's runtime configuration in the *XS Job Dashboard*.

   > **→ Tip**
   >
   > Access to the XS Job Dashboard tool is only available to users who have been assigned the XS classic administrator role "`JobSchedulerAdministrator`". For more information about XS classic administration roles, see *Related Information* below.

   a. Start the *SAP HANA XS Administration Tool*.

      The *SAP HANA XS Administration Tool* is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   b. Maintain the details of the XS job.

      > **Note**
      >
      > To maintain details of an XS job using the Web-based *XS Administration Tool* you need the privileges granted in the SAP HANA user role *sap.hana.xs.admin.roles::JobAdministrator*.

      You need to specify the following details:

      - *User*
        The user account in which the job runs, for example, **SYSTEM**

- *Password*
  The password required for user, whose account is used to run the job.
- *Locale*
  The language encoding required for the locale in which the job runs, for example, **`en_US`**
- *Start/Stop time*
  An optional value to set the period of time during which the job runs. Enter the values using the syntax used for the SAP HANA data type `LocalDate` and `LocalTime`, for example, **`2014-11-05 00:30:00`** (thirty minutes past midnight on the 5th of November 2014).
- *Active*
  Enable or disable the job schedule
- *Session timeout*
  Specify the session timeout for this XSJob in seconds. If you specify a value of 0 (zero) seconds for the XSJob's session timeout, the XSJob checks if a value is defined for the `sessiontimeout` key in the section `scheduler` of the `xsengine.ini` file. If no such key exists, the default session timeout of 900 seconds is used. If you want to define a non-default value for the scheduler's `sessiontimeout` key, you must create the key in the `scheduler` section of the `xsengine.ini` file and supply the desired timeout value, for example, 600 seconds.

  > ⚠ Caution
  >
  > It is not recommended to specify a value of 0 (zero) for the `sessiontimeout` key; this disables the session-timeout feature for all jobs started by the scheduler.

c. Save the job.

   Choose *Save Job* to save and activate the changes to the job schedule.

6. Enable the job-scheduling feature in SAP HANA XS.

   This step requires the permissions granted to the SAP HANA administrator.

   > ⓘ Note
   >
   > It is not possible to enable the scheduler for more than one host in a distributed SAP HANA XS landscape.

   a. In the *XS Job Dashboard* set the *Scheduler Enabled* toggle button to **YES**.

   Toggling the setting for the *Scheduler Enabled* button in the *XS Job Dashboard* changes the value set for the SAP HANA configuration variable *xsengine.ini* > *scheduler* > *enabled*, which is set in the *Configuration* tab of the SAP HANA studio's *Administration* perspective.

7. Check the job logs to ensure the XS job is active and running according to the defined schedule.

   You can view the `xsjob` logs in the *XS Job Dashboard* tab of the *SAP HANA XS Administration Tool*.

   > ⓘ Note
   >
   > To maintain details of an XS job using the Web-based *XS Administration Tool* you need the privileges granted in the SAP HANA user role *sap.hana.xs.admin.roles::JobAdministrator*.

   If the job does not run at the expected schedule, the information displayed in the xsjob logs includes details of the error that caused the job to fail.

Related Information

The XS Job File [page 1240]
XS Job-File Keyword Options
The XS Job Dashboard [page 1230]
SAP HANA XS Classic Administration Roles [page 1137]

14.1.11.3.1 The XS Job File

The `.xsjob` file defines the details of a task that you want to run (for example, an XS JavaScript or an SQLScript) at a scheduled interval.

The XS job file uses a `cron`-like syntax to define the schedule at which the service defined in an XS JavaScript or SQLScript must run, as you can see in the following example, which runs the specified job (the stock-price checking service `yahoo.xsjs`) on the 59th second minute of every minute.

```
{
    "description": "Read stock value",
    "action": "yahoo:yahoo.xsjs::readStock",
    "schedules": [
       {
          "description": "Read current stock value",
          "xscron": "* * * * * * 59",
          "parameter": {
             "stock": "SAP.DE"
             }
       }
    ]
}
```

When defining the job schedule in the `xsjob` file, pay particular attention to the entries for the following keywords:

- `action`
  Text string used to specify the path to the function to be called as part of the job.

  ```
  "action": "<package_path>:<XSJS_Service>.xsjs::<FunctionName>",
  ```

  > **Note**
  > You can also call SQLScripts using the `action` keyword.

- `description`
  Text string used to provide context when the XSjob file is displayed in the *SAP HANA XS Administration* tool.
- `xscron`
  The schedule for the specified task (defined in the `"action"` keyword); the schedule is defined using `cron`-like syntax.
- `parameter`
  A value to be used during the action operation. In this example, the parameter is the name of the stock `SAP.DE` provided as an input for the parameter (`stock`) defined in the `readStock` function triggered by

the `xsjob` action. You can add as many parameters as you like as long as they are mapped to a parameter in the function itself.

The following examples illustrate how to define an `xscron` entry including how to use expressions in the various `xscron` entries (day, month, hour, minutes,...):

- `2013 * * fri 12 0 0`
  Every Friday of 2013 at 12:00 hours
- `* * 3:-2 * 12:14 0 0`
  Every hour between 12:00 and 14:00 hours on every day of the month between the third day of the month and the second-last day.

  > → Tip
  >
  > In the day field, third from the left, you can use a negative value to count days backwards from the end of the month. For example, `* * -3 * 9 0 0` means: three days from the end of every month at 09:00.

- `* * * * * */5 *`
  Every five minutes (`*/5`) and at any point (`*`) within the specified minute.

  > ⓘ Note
  >
  > Using the asterisk (*) as a wild card in the seconds field can lead to some unexpected consequences, if the scheduled job takes less than 59 seconds to complete; namely, the scheduled job restarts on completion. If the scheduled job is very short (for example, 10 seconds long), it restarts repeatedly until the specified minute ends.

  To prevent short-running jobs from restarting on completion, schedule the job to start at a specific second in the minute. For example, `* * * * * */5 20` indicates that the scheduled job should run every five minutes and, in addition, at the 20th second in the specified minute.
- `* * * -1.sun 9 0 0`
  Every last Sunday of a month at 09:00 hours

Related Information

XS Job File Keywords
Tutorial: Schedule an XS Job [page 1236]

14.1.12 Maintaining Translation Text Strings

Maintain the translated text strings used in an application's user interface, error messages, and documentation.

For the purposes of localisation (L10N), you can provide the text strings displayed in an application's user interface in multiple languages, for example, English, French, or Chinese. You can also provide notifications and error messages in the same, local languages. To manage and maintain these translated text strings, SAP HANA provides an online translation tool (OTT). The translation of the text strings themselves can be performed

manually or with suggestions provided by an external service, for example, SAP Translation Hub. Access to external translation services is not covered by the SAP HANA license and usually requires a user account.

Setting up and maintaining the online translation tools for SAP HANA includes the following high-level tasks:

- Enabling the translation tool
- Accessing packages in the SAP HANA repository
- Maintaining text strings in the source and target languages
  This tasks involves maintaining the contents of the following SAP HANA tables:
  - ACTIVE_CONTENT_TEXT
  - ACTIVE_CONTEXT_TEXT_CONTENT
  - ACTIVE_OBJECT_TEXT
  - ACTIVE_OBJECT_TEXT_CONTENT
- Enabling access to a remote text-translation service (**optional**)

  > ⚠ Restriction
  >
  > Access to external translation services is not granted in the SAP HANA license. To use external translation services such as the *SAP Translation Hub*, an additional license is required. In addition, the *SAP Translation Hub* is currently available only for Beta testing.

- Maintaining HTTP destinations for any remote systems that provide services used by the *Online Translation Tool* (**optional**)
  Remote translation services such as *SAP Translation Hub* can provide access to a database of translated text strings, which are used to provide suggestions in the target language. To access such a remote service, you must maintain an HTTP destination (or extend an existing destination) that provides details of the host system where the translation service is running as well as a valid user account and logon authentication. You must also ensure that a trust relationship exists between the translation server and SAP HANA, for example, by importing the translation server's client certificate into the SAP HANA trust store.

The SAP HANA *Online Translation Tool* is available on the SAP HANA XS Web server at the following URL:

```
http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/translationTool/
```

> → Tip
>
> The privileges required to use the SAP HANA *Online Translation Tool* (OTT) are granted by the role template`sap.hana.xs.translationTool.roles::translator`.

Related Information

Create and Edit Text Translations [page 1243]
Export and Import Translated Text [page 1247]
SAP Translation Hub Cloud Service (beta)

14.1.12.1 Create and Edit Text Translations

Maintain translations for text strings displayed in an SAP HANA application's user interface.

Prerequisites

To maintain translated text for an application in SAP HANA XS, the following prerequisites apply:

- You have access to an SAP HANA system.
- You have the privileges required to access the repository packages containing the text strings to be localized/translated.
- You have a role based on the role template `sap.hana.xs.translationTool.roles::translator`.
- If you want to make use of optional external translation services, you must maintain access to the translation server system.

  > **Restriction**
  >
  > Access to external translation services is not granted in the SAP HANA license. To use external translation services such as the *SAP Translation Hub*, an additional license is required. The *SAP Translation Hub* is currently available only for BETA testing.

  Details of the remote systems where the translation service is running (for example, SAP Translation Hub) are defined in HTTP destination configuration files along with details of any corresponding user account and authentication certificates.

Context

An application's user interface and notifications can be translated from the original source language (for example, English) into one or more local (target) languages, for example, French, Spanish, or Japanese. You can either translate the texts manually or with the help of an (optional) external translation service. To provide translations of the UI text strings for your SAP HANA application, perform the following steps:

Procedure

1. Start the *SAP HANA Online Translation Tool*.

   The *SAP HANA Online Translation Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/translationTool`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who

> logs on must also have the privileges required to perform the tasks associated with the maintenance of translation texts.

2. Select the delivery unit that contains the application with the text strings you want to translate.

   Use the *Delivery Unit* drop-down list to select a delivery unit.

   > → Tip
   >
   > The name of the vendor associated with the selected delivery unit is displayed automatically in the *Vendor* field, for example, *acme.com*; the vendor name cannot be changed here.

3. Select the package that contains the text strings you want to translate.

   Use the *Package* drop-down list to select a package. If the selected package contains text elements, they are displayed alphabetically in a list.

   > → Tip
   >
   > The original source language associated with the contents of the selected package is displayed automatically.

4. Enable access to a text-translation service, for example, *SAP Translation Hub*. (**optional**).

   > ⚠ Restriction
   >
   > Access to external translation services is not granted in the SAP HANA license. To use external translation services, an additional license is required.

   If you want to make user of the services provided by a translation server, you need to maintain an HTTP destination **extension** that provide details of the host system where the translation service is running; access to the translation service usually requires a user account and logon authentication. You must also ensure that a trust relationship exists between the translation server and SAP HANA, for example, by importing the translation server's client certificate into the SAP HANA trust store that you are using to handle authentication for this HTTP destination.

   The HTTP destination configuration `sap.hana.xs.translationTool.server:translationService.xshttpdest` defines details of the server hosting the SAP Translation Hub service. Although you cannot edit this destination configuration, note that you can use an HTTP destination **extension** to change the details, for example, to point to an alternative host name.

5. Add a translation for a text element.

   For a given text element in the *Text ID* list, you can provide a suitable translation in one or more languages, for example: French (*fr*), Spanish (*es*), and Japanese (*ja*).

   a. Expand the desired UI text element.

      In the *Text ID* list, locate and expand the element for which you want to provide a translation.

   b. Add a translation.

      Choose *Add Translation*.

   c. Select the desired language for the translation from the *Target Language* drop-down list.

   d. In the *Target Language Text* box, type the translation for the selected text element.

> → Tip
>
> If the *SAP Translation Hub* option is enabled, language-specific suggestions for possible translation matches are provided as you type. If you see a suggestion that is suitable, use the mouse to select the suggested text.

e. Add another translation.

   Choose *Add Translation*

f. Edit an existing translation

   Choose the *Edit* icon next to the translation you want to modify and make the required changes.

6. Save your additions and changes.

   Choose *Save* to store the added translations or any modifications in the appropriate tables in the SAP HANA database.

Related Information

Online Translation Tool Details [page 1245]
Export and Import Translated Text [page 1247]
Edit an HTTP Destination Runtime Configuration [page 1154]
Managing Trust Relationships [page 1160]

14.1.12.1.1 Online Translation Tool Details

Display details of the source text for an application's user interface elements and, if available, any available translations.

The *Online Translation Tool* tool enables you to view details of the text elements contained in the individual packages of an SAP HANA application. The following table indicates which information can be viewed.

> ⓘ Note
>
> The privileges required to use the SAP HANA *Online Translation Tool* (OTT) are granted by the role template `sap.hana.xs.ott.roles::translator`.

Translation Text Details

UI Element	Description	Example
Delivery Unit	Name of the SAP HANA delivery unit (DU) that contains the default text strings for which a translation is required along with the name of the vendor associated with the selected delivery unit	ACME_XS_BASE - acme.com
Package	The name of (and path to) the package containing the text strings for which a translation is required	acme.com.app.ui.login

UI Element	Description	Example
Source language	Short name of the source language for the text strings contained in the selected package, for example: en (English), fr (French), ja, (Japanese)	en
Target Language	Long or short name of the target language for the text strings contained in the selected package, for example: Bulgarian (bg), French (fr), Japanese (ja)	Chinese (zh)
Domains	The SAP product-specific translation domain to which the selected DU/package belongs, for example, *Financial Accounting* or *Customer Relationship Managment*. Domains are used in the translation process to determine the correct terminology for a text string that has to be translated; the same text might require a different translation depending on the domain (or application) in which it is used. Suggestions from a remote translation service such as the SAP Translation Hub are restricted to the currently selected domain.	"Basis", or "Accounting - General"
Enable Translation Hub	Enable automatic suggestions (in the *Target language text* box) for translation texts using a remote service such as SAP Translation Hub; the suggestions are provided by a remote translation database. **⚠ Restriction** Access to external translation services is not granted in the SAP HANA license. To use external translation services such as the *SAP Translation Hub*, an additional license is required. The *SAP Translation Hub* is currently available only for BETA testing. Access to the remote translation service usually requires a user account and logon authentication. You also need to maintain an HTTP destination (or extend an existing one) for the translation server system and ensure the server system is trusted by SAP HANA, for example, by importing the translation server's client certificate into the SAP HANA trust store.	Yes/No
Text ID	The name/ID of the UI element for which a text string is required. This could be a tab title, a box name, a notification, or an error message.	LOGON_LABEL
Default Text	The text string associated with the text ID	HANA Logon
Target Language Text	Proposed/accepted translation (in the target language) of the text string displayed (in the source language) in the *Default Text* field. Activate the *Enable Translation Hub* option to enable auto-suggestions in the target language.	-
Source Object	The name of the design-time artifact that contains the UI text strings.	logonForm.hdbtextbundle

Related Information

Create and Edit Text Translations [page 1243]
Export and Import Translated Text [page 1247]
Managing Trust Relationships [page 1160]

14.1.12.2 Export and Import Translated Text

Transport text translations between systems using the industry-standard, XML-based `xliff` format.

Prerequisites

To export and import translated text for an application in SAP HANA XS, the following prerequisites apply:

- You have access to an SAP HANA system.
- You have access to the repository packages containing the text strings to be localized/translated.
- You have been granted a role based on the role template `sap.hana.xs.translationTool.roles::translator`.

Context

An application's user interface and notifications can be translated from the original source language (for example, English) into one or more target local languages, for example, French, Spanish, or Japanese. To provide translations of the UI text strings for your SAP HANA application, perform the following steps:

Procedure

1. Start the *SAP HANA Online Translation Tool*.

   The *SAP HANA Online Translation Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/translationTool`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must also have the privileges required to perform the tasks associated with the maintenance of translation texts.

2. Select the delivery unit that contains the application with the text strings you want to translate.

   Use the *Delivery Unit* drop-down list to select a delivery unit.

> → Tip
>
> The name of the vendor associated with the selected delivery unit is displayed automatically in the *Vendor* field, for example, *acme.com*. You cannot change this here.

3. Select the package that contains the text strings you want to translate.

   Use the *Package*drop-down list to select a package. If the selected package contains text elements, they are displayed automatically in an alphabetically ordered list.

   > → Tip
   >
   > The original source language associated with the contents of the selected package is displayed automatically.

4. Export the UI text elements from the local source system.

   You can export the translation texts to an archive on a local file system using the industry-standard, XML-based `xliff` format.

5. Import the UI text elements to the remote target system.

   You can import the translation texts into SAP HANA from an archive whose content are stored using the industry-standard, XML-based `xliff` format.

6. Confirm that the import operation was successful.

   Check the status of the following tables in the SAP HANA database:

   - ACTIVE_CONTENT_TEXT
   - ACTIVE_CONTEXT_TEXT_CONTENT
   - ACTIVE_OBJECT_TEXT
   - ACTIVE_OBJECT_TEXT_CONTENT

Related Information

Online Translation Tool Details [page 1245]
Create and Edit Text Translations [page 1243]

14.1.13 Maintaining HTTP Traces for SAP HANA XS Applications

HTTP tracing for individual SAP HANA XS applications can be enabled on the SAP HANA Web Dispatcher.

The *SAP HANA XS Administration Tools* include the *SAP Web Dispatcher HTTP Tracing* application, which you can use to enable and disable HTTP tracing on the SAP Web Dispatcher for SAP HANA XS applications.

> ⓘ Note
>
> SAP HANA uses roles to grant access to the features provided by the *SAP HANA XS Administration Tool*. To access the administration tools required to manage HTTP tracing on the SAP Web Dispatcher, you

must have a role based on the role template `WebDispatcherHTTPTracingAdministrator`. The role template `WebDispatcherHTTPTracingViewer` contains the privileges for read-only access to the *SAP Web Dispatcher HTTP Tracing* tool.

You can use the *SAP HANA XS Administration Tools* to perform the following tasks:

- Display a list of all traced applications
  List all applications defined in the system. Details include the application's metadata, information about HTTP tracing configuration for the particular application, the status of the XS job that starts the tracing process, and HTTP tracing log information.
- Enable HTTP tracing
  Enable HTTP tracing for selected SAP HANA XS applications
- Disable HTTP tracing
  Disable HTTP tracing for selected SAP HANA XS applications

Tracing is managed by the XS job `sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob`, which runs at a predefined schedule. If you enable or disable HTTP tracing, you must modify the XS job file accordingly.

→ Tip

Administrator access to the XS job details requires the privileges granted by the role template `sap.hana.xs.admin.roles::JobAdministrator`. These privileges are already included in the `WebDispatcherHTTPTracingAdministrator` role template, which is required to use the *SAP Web Dispatcher HTTP Tracing*.

HTTP tracing is enabled by setting configuration parameters in SAP HANA XS (`xsengine.ini`) and the SAP Web Dispatcher (`webdispatcher.ini`). If an SAP HANA XS application is defined in a parameter in `xsengine.ini`, then HTTP tracing is enabled for the specified application. If not, then HTTP tracing is disabled for the application.

ⓘ Note

If HTTP tracing is disabled for an application, the corresponding HTTP trace parameters in `xsengine.ini` and `webdispatcher.ini` are removed. If you re-enable HTTP tracing on the SAP Web Dispatcher for the same application, the required parameters are recreated automatically.

Connections to the database are performed with the SQL auto-user defined in */sap/hana/xs/admin/webdispatcher/server/common/httpTracing.xssqlcc*.

Related Information

SAP HANA XS Classic Administration Roles [page 1137]
Enable HTTP Tracing for an SAP HANA XS Application [page 1252]
Maintain XS Job Details [page 1228]

14.1.13.1 Display the HTTP Trace Status of SAP HANA XS Applications

Display a list of SAP HANA XS applications which shows the status of HTTP tracing.

Prerequisites

To use the *SAP HANA XS Administration Tool* to view the current status of HTTP tracing for SAP HANA XS applications, the following prerequisites apply:

- You have administrator access to an SAP HANA system.
- You have been granted roles based on one of the following role templates:
  - `sap.hana.xs.admin.roles::WebDispatcherHTTPTracingViewer`
  - `sap.hana.xs.admin.roles::WebDispatcherHTTPTracingAdministrator`

Context

To use the *SAP HANA XS Administration Tool* to display a list of applications and the HTTP trace status, perform the following steps:

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *SAP Web Dispatcher HTTP Tracing* tool.

2. Start the *SAP Web Dispatcher HTTP Tracing* tool.

   In the list of XS Administration tools, choose *SAP Web Dispatcher HTTP Tracing*.

3. Display a list of the SAP HANA XS applications running on the system to which you are connected; the *HTTP Tracing Enabled* column indicates (*Yes*/*No* if HTTP tracing is enabled for the application.

   > → Tip
   >
   > You can use the search box to display a list of only those applications that match a particular string, for example, "**`admin`**".

Related Information

Enable HTTP Tracing for an SAP HANA XS Application [page 1252]
Application HTTP Tracing Details [page 1251]

14.1.13.1.1 Application HTTP Tracing Details

Display a list of the SAP HANA XS applications for which HTTP tracing is enabled on the SAP Web Dispatcher.

The *XS Applications* tab in the *SAP Web Dispatcher HTTP Tracing* tool enables you to view a list of the SAP HANA XS applications for which HTTP tracing is enabled on the SAP Web Dispatcher. The following table indicates which information can be viewed.

> → Tip
>
> You can use the search box to display a list of the applications that match a particular string, for example, "`admin`".

Job Details

UI Element	Description	Example
SAP Web Dispatcher HTTP Tracing Job	The SAP HANA XS job used to start the tracing operation for the listed applications	httptracing.xsjob
ACTIVE/INACTIVE	The current status of the HTTP tracing job that manages the tracing operation for the selected applications	ACTIVE
Application Name	The full path to (and the name of) the SAP HANA XS application for which HTTP tracing is enabled on the SAP Web Dispatcher	sap.hana.xs.admin
Delivery Unit	The name of the delivery unit that contains the application specified in *Application Name*	HANA_XS_ADMIN
Vendor	The name of the vendor responsible for the creation and maintenance of the delivery unit that contains the traced application	sap.com
HTTP Tracing Enabled	The current tracing status: No (disabled); yes (enabled)	Yes

Related Information

Enable HTTP Tracing for an SAP HANA XS Application [page 1252]

14.1.13.2 Enable HTTP Tracing for an SAP HANA XS Application

HTTP tracing on the SAP Web Dispatcher can be enabled for one or more SAP HANA XS applications

Prerequisites

To enable HTTP tracing on the SAP Web Dispatcher for SAP HANA XS applications, the following prerequisites apply:

- You have administrator access to an SAP HANA system.
- You have been granted a role based on the role template `sap.hana.xs.admin.roles::WebDispatcherHTTPTracingAdministrator`.
- The XS job *sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob* is configured and running. (By default, the job runs at 12:00 every day.)
- The XS SQL connection configuration */sap/hana/xs/admin/webdispatcher/server/common/httpTracing.xssqlcc* is active (available by default).

Context

To enable HTTP tracing on the SAP Web Dispatcher for an application, you must perform the following steps:

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > ⓘ Note
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *SAP Web Dispatcher HTTP Tracing* tool.

2. Start the *SAP Web Dispatcher HTTP Tracing* tool.

   In the list of XS administration tools, choose *SAP Web Dispatcher HTTP Tracing*.

3. Display a list of the SAP HANA XS applications running on the system to which you are connected.

   > → Tip
   >
   > You can use the search box to display a list of only those applications that match a particular string, for example, "**`admin`**".

4. Enable HTTP tracing for an application.

   In the *XS Applications* tab, the *HTTP Tracing Enabled* column indicates if HTTP tracing is enabled or not (*Yes*/*No*) for the application.

   a. In the *XS Applications* tab, choose *Edit*.
   b. **Check** the box for the application for which you want to enable HTTP tracing.
   c. In the *XS Applications* tab, choose *Save*.

   Saving the changes to the configuration enables HTTP tracing and automatically sets the following configuration parameters (keys):

   - Configuration section: `webdispatcher.ini/profile`
     - key
       `icm/HTTP/logging_n`
     - value
       `PREFIX=/path/to/app/, LOGFILE=$(_LOCAL_HOST_NAME)/trace/access_log_app-%y-%m-%d, MAXSIZEKB=10000, SWITCHTF=day, LOGFORMAT=SAP, FLUSH=1`
   - Configuration section: `xsengine.ini/customer_usage`
     - key=`/path/to/appname/`
     - value=`icm/HTTP/logging_n`

     > → Tip
     >
     > This is the value defined for the key `webdispatcher.ini/profile`.

5. Update the XS job used to start the trace operation.

   The XS job `sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob` is used to stop and start HTTP tracing on the SAP Web Dispatcher for individual XS applications. The current status of the XS job is indicated in the *SAP Web Dispatcher HTTP Tracing* dialog.

   a. In the *SAP Web Dispatcher HTTP Tracing* dialog, click the link to the XS job `sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob`.

      The *XS Job Details* window displays a brief description of the XS job and information about any configured schedules.
   b. Choose the *Configuration* tab to set up the XS job.
   c. Type the name of a user with the required permisson to run the XS job and the corresponding password.
   d. Check the *Active* box.
   e. Choose *Save Job* to update the XS job and start the HTTP tracing.

      > ⓘ Note
      >
      > A user name and password are required to save the changes you make to the XS job.

6. Check the new log file is created and contains entries.

   The log file is located in the folder you specified in the `webdispatcher.ini/profile` key `icm/HTTP/logging_n`, for example:

   `LOGFILE=$(_LOCAL_HOST_NAME)/trace/access_log_app-%y-%m-%d`

   Where `app` is the name of the application whose HTTP traffic you are tracing.

Related Information

Application HTTP Tracing Details [page 1251]
SAP HANA XS Classic Configuration Parameters [page 1140]
SAP HANA XS Classic Administration Roles [page 1137]

14.1.13.3 Disable HTTP Tracing for an SAP HANA XS Application

HTTP tracing on the SAP Web Dispatcher can be disabled for one or more SAP HANA XS applications.

Prerequisites

To enable HTTP tracing on the SAP Web Dispatcher for SAP HANA XS applications, the following prerequisites apply:

- You have administrator access to an SAP HANA system.
- You have been granted a role based on the role template `sap.hana.xs.admin.roles::WebDispatcherHTTPTracingAdministrator`.
- The XS job *sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob* is configured and running. (By default, the job runs at 12:00 every day.)
- The XS SQL connection configuration */sap/hana/xs/admin/webdispatcher/server/common/httpTracing.xssqlcc* is active (available by default).

Context

To disable HTTP tracing on the SAP Web Dispatcher for an application, you must perform the following steps:

Procedure

1. Start the *SAP HANA XS Administration Tool*.

   The *SAP HANA XS Administration Tool* tool is available on the SAP HANA XS Web server at the following URL: `http://<WebServerHost>:80<SAPHANAinstance>/sap/hana/xs/admin/`.

   > **Note**
   >
   > In the default configuration, the URL redirects the request to a logon screen, which requires the credentials of an authenticated SAP HANA database user to complete the logon process. The user who logs on must have the privileges required to perform administration tasks with the *SAP Web Dispatcher HTTP Tracing* tool.

2. Start the *SAP Web Dispatcher HTTP Tracing* tool.

   In the list of XS administration tools, choose *SAP Web Dispatcher HTTP Tracing*.

3. Display a list of the SAP HANA XS applications running on the system to which you are connected.

   > → Tip
   >
   > You can use the search box to display a list of only those applications that match a particular string, for example, "`admin`".

4. Disable HTTP tracing for an application.

   In the *XS Applications* tab, the *HTTP Tracing Enabled* column indicates if HTTP tracing is enabled or not (*Yes*/*No*) for the application.

   a. In the *XS Applications* tab, choose *Edit*.
   b. **Uncheck** the box for the application for which you want to **disable** HTTP tracing.
   c. In the *XS Applications* tab, choose *Save*.

   Saving the changes to the configuration disables HTTP tracing for the selected application and **removes** the following parameters (keys):

   - Configuration section: `webdispatcher.ini/profile`
     - key=`icm/HTTP/logging_n`
   - Configuration section: `xsengine.ini/customer_usage`
     For example:
     - key=`/path/to/appname/`

5. Update the XS job used to stop the trace operation.

   The XS job `sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob` is used to stop and start HTTP tracing on the SAP Web Dispatcher for individual XS applications. The current status of the XS job is indicated in the *SAP Web Dispatcher HTTP Tracing* dialog.

   a. In the *SAP Web Dispatcher HTTP Tracing* dialog, click the link to the XS job `sap.hana.xs.admin.webdispatcher.jobs::httptracing.xsjob`.

      The *XS Job Details* window displays a brief description of the XS job and information about any configured schedules.
   b. Choose the *Configuration* tab to set up the XS job.
   c. Type the name of a user with the required permisson to run the XS job and the corresponding password
   d. Uncheck the *Active* box.
   e. Choose *Save Job* to update the XS job and stop the HTTP tracing.

      > ⓘ Note
      >
      > A user name and password are required to save the changes you make to the XS job.

6. Check that tracing has been switched off and **no** new logs files are being created.

   The log files for the traced application are located in the folder you specified in the `webdispatcher.ini/profile` key `icm/HTTP/logging_n`, for example:

   ```
   LOGFILE=$(_LOCAL_HOST_NAME)/trace/access_log_app-%y-%m-%d
   ```

   Where `app` is the name of the application whose HTTP traffic you are are tracing.

Related Information

SAP HANA XS Classic Administration Roles [page 1137]
SAP HANA XS Classic Configuration Parameters [page 1140]

14.2 Maintaining the SAP HANA XS Advanced Model Run Time

Maintain the SAP HANA XS advanced model run-time environment.

From HANA 1.0 SPS 11, SAP HANA includes an additional run-time environment for application development: SAP HANA extended application services (XS), advanced model. SAP HANA XS, advanced model represents an evolution of the application server architecture within SAP HANA by building upon the strengths (and expanding the scope) of SAP HANA extended application services (XS), classic model. SAP recommends that customers and partners who want to develop new applications use SAP HANA XS advanced model.

> → Tip
>
> If you want to migrate existing XS classic applications to run in the new XS advanced run-time environment, SAP recommends that you first check the features available with the installed version of XS advanced; if the XS advanced features match the requirements of the XS classic application you want to migrate, then you can start the migration process.

SAP HANA extended application services, advanced model (XS advanced) is a platform that enables the management of polyglot Web-based applications built on micro-services but also supports the more traditional "monolithic" application. That means that, while XS classic applications were restricted to a server-side JavaScript dialect called XSJS, applications in XS advanced can be written in a programming language of your choice. By default, the platform supports Java, Node.js and XSJS, and Python, but also contains a plug-in concept that enables the use of additional "custom" languages, too. Moreover, in contrast to XS classic, XS advanced applications are not executed by a central server. Instead, each application has its own server and run-time environment – an architecture that enables a very flexible composition of applications using the concept of micro-services.

In XS advanced, an application can either be comprised of several independent and replaceable components or it can reuse services that are already present on the platform, and each component can be developed, maintained, and updated individually. This allows a high level of flexibility in terms of deployment. For example, an application can be composed of a Java and a Node.js component, each of which can be developed and updated independently, and, if required, the same application can reuse an additional service written in Python.

It is the task of the XS advanced run time to try to reduce to a minimum the additional complexity this advanced concept might introduce. For example, the XS advanced run-time environment is responsible for connecting micro-services together to form a complete application. XS advanced starts, stops, and monitors applications, and provides a simple means of connecting applications to backing services such as the SAP HANA database (using a concept called “service brokers”, which provide components for user authentication and the higher level application deployment services). Note that XS advanced is designed in a way that makes it possible to write business applications that can be moved from the on-premise platform to the SAP Cloud Platform and vice versa.

→ Tip

For general information, advice, and a list of frequently asked questions about XS advanced issues and solutions, see *Related Information* below.

Downloading XS Advanced from SAP Marketplace

SAP HANA Extended Application Services, advanced model, is available not only on the SAP HANA media but also as a separate component on SAP Marketplace. Users with the required S-User ID can download the latest version of XS advanced component in the package `SAP EXTENDED APP SERVICES 1` from the following location:

Service Marketplace > *Software Downloads [Downloads]* > *SUPPORT PACKAGES & PATCHES* > *By Alphabetical Index (A-Z)* > *H* > *SAP HANA PLATFORM EDITION*:

- *SAP HANA PLATFORM EDITION 2.0* > *SAP EXTENDED APP SERVICES 1*

→ Tip

SAP HANA Extended Application Services, advanced model, is backwards compatible; you can provide access to new features by installing the latest version of the XS advanced component even on older versions of SAP HANA. To download the package `SAP EXTENDED APP SERVICES 1`, see *SAP Software Download Center* in *Related Information* below.

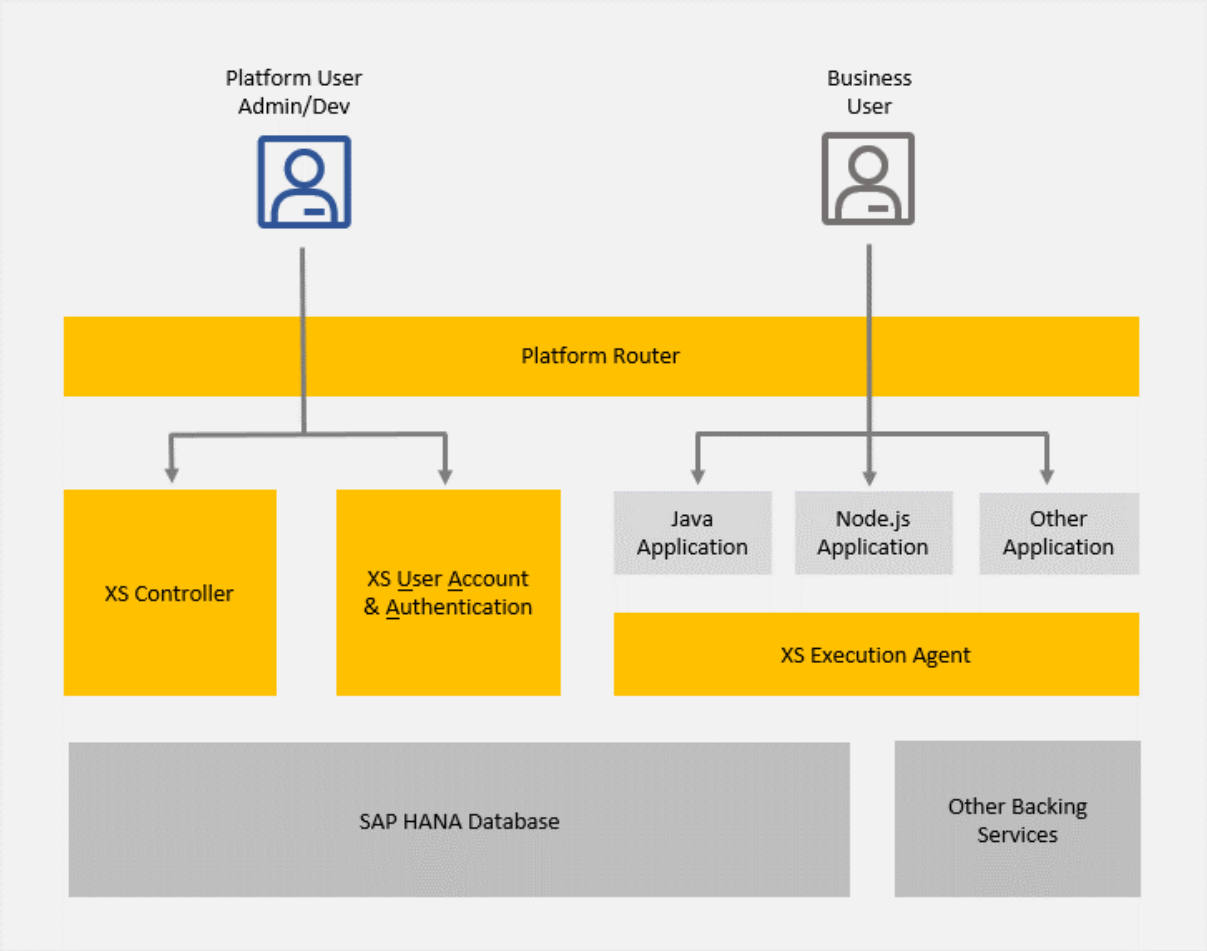
Related Information

SAP Note 2596466
SAP Software Download Center (Logon required)
SAP Note 2741459
SAP HANA XS Advanced Migration Guide

14.2.1 XS Advanced Platform Components

An overview of the main components of the XS advanced model run-time environment.

The SAP HANA XS advanced model run-time environment is comprised of the components illustrated in the following diagram and described in more detail in the corresponding sections.



The XS Advanced Controller

The XS advanced Controller (XS Controller) is a central component of the platform and is deployed once within an SAP HANA system. The XS Controller has the following purposes:

1. The XS Controller provides an application programming interface (API) for the platform user (administrators or application developers). The API enables users to perform the following tasks:
   - Maintain applications and service instances (resources provided by backing services)
   - Maintain XS advanced role collections (by means of an API provided by the XS User Account and Authentication service)
   - Maintain application run-time environments
   - Maintain application build packs (a convenient way to bring in custom programming languages for applications)

   > → Tip
   >
   > A command-line based administration tool (the `xs` client) and a GUI based tool (the XS Advanced Admin Cockpit) are provided to help administrators maintain the XS advanced model run-time platform. For more information, see *Related Information* below.

2. The XS Controller maintains the state of the XS advanced platform and communicates with a group of Execution Agents which are used to start, monitor, or stop applications running in the XS advanced run-time environment.

The XS Advanced Execution Agent

The Execution Agent manages and monitors application instances. Several Execution Agents can be connected to the XS Controller, for example, if it is necessary to scale out application instances to several hosts.

Managed by the `xsexecagent` service, Execution Agents are primarily responsible for starting and stopping application instances. To be reachable for end users, launched application instances typically provide a public HTTP port. The Execution Agent checks the availability of this end point regularly. Instances of an application that are no longer reachable are restarted automatically.

In a distributed system, different instances of the same application do not necessarily run on the same host, although it is possible to pin applications to a particular host. If spaces are mapped to different OS users, the Execution Agents also ensure that application instances running in different spaces are not visible to each other at the operating system (OS) level.

XS Advanced User Account and Authentication

The XS advanced User Account and Authentication (UAA) is the component that handles user-authentication requests for platform users as well as for business users.

The UAA service is a multi-tenant identity management service. The UAA's primary role is as an OAuth2 provider, issuing tokens for client applications to use when they act on behalf of XS advanced (or Cloud Foundry) users. The UAA can also authenticate users with their XS advanced (or Cloud Foundry) credentials and can act as an SSO service using those credentials (or others).

As well as various other management functions, the UAA provides end points for the management of user accounts and for registering OAuth2 clients.

You can configure an SAP HANA system to act as a service provider for XS advanced applications that use Single Sign On (SSO) authentication based on Security Assertion Markup Language (SAML) certificates. An SAML identity provider is used by the SAML service provider to authenticate users signing in by means of a single sign-on mechanism. If SAP HANA XS advanced applications are configured to use SAML assertions as the logon authentication method, then an SAML IdP is required. It is also possible to assign roles automatically to users who log on to an application by means of single sign-on with SAML assertions.

> → Tip
>
> For more information about maintaining SAML Identity Providers (IdP) in XS advanced, see *Related Information* below.

XS Advanced Platform Router

The Platform Router is the Web-dispatcher component that exposes all public end points of XS advanced platform components as well as for the running applications. The Platform Router is also responsible for load-balancing requests between several instances of an application. One or more end points (so called routes) can be exposed per application.

XS Advanced Application Instances

By default, the XS advanced platform supports applications written in Java, Node.js, server-side JavaScript (XSJS), and Python, as well as other “custom” languages.

When an application is started by a platform user, the Execution Agent starts one or more instances of this application. Each application is run in its own environment and file-system sandbox. In this way, several application instances can be started, for example, depending on the current load an application has to take, without affecting the other application instances.

The number of application instances can be adapted (scaled) during run time. All instances of an applications are exposed under the same route, which means they share the same URI, a feature that enables transparent load-balancing. Using a round-robin algorithm, the Platform Router balances the load by distributing requests between available application instances. Sticky sessions are used to ensure proper caching of data inside single application instances.

XS Advanced Backing Services

Backing Services provide resources to applications. Arbitrary backing services can be connected to the XS advanced platform by so-called “Service Brokers”. The service broker provides a standardized interface to the XS Controller to fetch credentials for a particular backing service. The credentials are exposed to applications by means of so-called “service instances”.

> → Tip
>
> For more information about the XS advanced Service Broker, see *Related Information* below.

A platform user can create a service instance and bind this service instance to a particular application, while in the background the XS Controller will fetch the corresponding credentials. When starting the application, the XS Controller injects these credentials into the environment of the application instance, which enables the application to use the backing service.

SAP HANA Database

The XS advanced platform uses SAP HANA as persistence as well as an infrastructure provider to manage the life cycle of XS advanced services, so it is important that you monitor the operation of SAP HANA databases on a regular basis.

Although SAP HANA actively alerts you about critical problems and situations, keeping an eye on resource usage and performance can help you identify patterns, forecast requirements, and recognize when something is wrong.

You can monitor SAP HANA using the SAP HANA Cockpit. The tools in the SAP HANA Cockpit rely on the monitoring and alerting information provided by system views and the statistics service.

Platform Administration

During installation of the SAP HANA XS advanced run-time component, a user named `XSA_ADMIN` is created and granted the privileges required to administrate the XS advanced run-time environment. The `XSA_ADMIN` user can be used to log in to the XS advanced platform, perform administration tasks, or create further administration or restricted users, for example, using the following tools:

- Command-line tools:
  - `xs`
    Maintain XS advanced run-time components (applications, services, brokers, etc.)
  - `XSA`
    Stop, start XS advanced run-time instances.

    > ⓘ Note
    >
    > Some `XSA` command require `XSA_ADMIN` user privileges; most `XSA` commands require only `<sid>adm` privileges, for example, `XSA restart`.
- GUI-based tools:
  XS Advanced Admin Cockpit.

Related Information

The XSA Command Reference [page 1370]
The XS Command-Line Interface [page 1262]
Maintaining the XS Advanced Run-time Environment with SAP HANA XS Advanced Cockpit [page 1387]
Managing SAML Identity Providers in XS Advanced [page 1423]
Maintaining Application Services in XS Advanced (XS Advanced Developer Guide)

14.2.2 Maintaining the XS Advanced Run-time Environment with a Command-Line Interface

Use command-line tools to administrate and maintain XS advanced-model run-time components.

SAP HANA provides the following tools to enable you to maintain your XS advanced run-time environment from the command line:

- The `xs` command-line interface
  A selection of utilities to help you maintain not only the applications that are deployed to the XS advanced run-time environment, but also the run-time environment itself, for example, the organizations and spaces, and the users who need access and use it.

  ```
  xs <command> [<ARGUMENTS>] [<OPTIONS>]
  ```

  The `xs` CLI is automatically installed along with the XS advanced platform on an SAP HANA system. If the `xs` CLI is not added to the SAP HANA system's `<PATH>` environment variable, you can find the executable in the default location `/bin/xs`. The `xs` CLI can also be used from a remote computer which connects over a secure connection to the SAP HANA system where you want to perform the administration tasks.

  > → Tip
  >
  > To use the `xs` CLI, you must first log on to an SAP HANA system as the operating-system user `<SID>adm` and then log on to the `xs` CLI, for example, as a user with `XSA_ADMIN` privileges. For more information about how to log on to the `xs` CLI, see *Related Information* below.

- The `XSA` command-line interface
  A selection of utilities to help you maintain XS advanced run-time instances, for example, enable, disable, restart XS advanced, or maintain domain certificates, etc.

  ```
  XSA {COMMAND} [--OPTIONS]
  ```

  > ⚠ Restriction
  >
  > To use the `XSA` command, you must log on as the operating-system user `<SID>adm`.

Related Information

Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
Offline Maintenance and Diagnostics of XS Advanced with the XSA CLI [page 1369]
Maintaining the XS Advanced Run-time Environment with a Graphical User Interface [page 1387]

14.2.2.1 Maintaining XS Advanced Run-Time Components with the XS CLI

Use the `xs` command-line interface (CLI) to manage the XS advanced run-time environment.

SAP HANA provides a command-line interface that enables you to maintain not only the applications that are deployed to the XS advanced run-time environment, but also specific elements of the run-time environment itself, for example, the components that enable it, and the users who access and use it. For example, you can use the XS CLI to maintain and manage the following components:

- Logon and setup
- XS advanced applications, routes, and tasks
- Organizations and spaces
- Domains and certificates
- Services
- Build packs, run-time environments, and the blob store
- XS advanced users
- Security aspects
- XS advanced application deployment, installation, and life cycle

> → Tip
>
> You can install and use the `xs` command-line interface tools over a secure connection from a remote machine. For more information about where to find and how to download the `xs` command-line interface tools, see *The XS Command-Line Interface* in *Related Information*.

Related Information

The XS Command-Line Interface [page 1262]
Logging on and Getting Started with the XS CLI [page 1265]

14.2.2.1.1 The XS Command-Line Interface

A list of all the categories and areas covered by the `xs` command-line interface (CLI).

The `xs` command-line interface is a set of tools that enable the administration of the XS advanced run time from the command-line. The `xs` CLI is automatically installed on the SAP HANA system during installation

of XS Advanced. However, you can also install the `xs` CLI tools on a remote machine and log on to the XS advanced run-time environment from the remote machine using a secure connection.

> → Tip
>
> The XS advanced, command-line client is available for download from SAP Support Portal for those people with the required S-User ID. Alternatively, you can find it on the SAP HANA installation media.

Downloading the `xs` CLI from the SAP Support Portal

Users with the required S-User ID can download the `xs` CLI from the SAP Support Portal:

SAP Support Portal > *Software Downloads [Downloads]* > *SUPPORT PACKAGES & PATCHES* > *By Alphabetical Index (A-Z)* > *H* > *SAP HANA PLATFORM EDITION*:

- *SAP HANA PLATFORM EDITION 2.0* > *XS RUNTIME 1*

Downloading the `xs` CLI from the SAP HANA Media

The `xs` CLI is also available for download from the SAP HANA installation media. You can find the Zip archive `xs.onpremise.runtime.client*` in the following location on the SAP HANA media:

`DATA_UNITS/XSA_CLIENT_10/xs.onpremise.runtime.client_<platform>-<version>.zip`

> → Tip
>
> Extract the contents of the Zip archive to the desired location. On Unix machines, the default location is the directory `/bin`.

Usage

```
xs <command> [<ARGUMENTS>] [<OPTIONS>]
```

To display information about a specific `xs` command:

```
xs help <command>
```

To display a list of all available `xs` commands, along with information about environment variables, and global options:

```
xs help -a
```

XS Command Overview

Command Category	Description
Logon and setup	User logon, view user-organization (and space) targets, set API URLs
Application Management	Maintain SAP HANA XS applications: list, deploy, start, stop, stage, [...]
Services Management	Maintain SAP HANA XS services: list, create, delete, bind, update, [...]

Command Category	Description
Organizations	Maintain user organizations: create, list, rename, delete, [...]
Spaces	Manage user spaces: create, list, rename, delete, [...]
Domains	Manage XS advanced domains: create, list, delete, set certificates, [...]
Certificates	Manage XS advanced certificates: set, unset, list trusted certificates
Routes	Maintain application routes: create, list, map, unmap, delete, [...]
Build Packs	Maintain application build-packs: create, list, update, rename, delete, [...]
Run-time Environments	Maintain XS run-times: create, list, display information, search, update, delete, [...]
Tasks	Maintain XS application-related tasks: list, run, cancel
User Administration	Maintain SAP HANA users: create, list, purge, delete, set/unset organizations, spaces, roles, [...]
Administration	Maintain XS application traces and backups
Tenant Databases	Manage and maintain tenant databases and their mapping to organizations and spaces
Configuration	Set and maintain environment variables and groups
Blob Store	Manage and maintain the contents of the blob store
Advanced	Retrieve and display OAuth tokens for the current session
Other Commands	Display information about the SAP HANA XS version, CLI, and system
Plug-ins	Additional commands as plug-ins; install/remove product components, deploy multi-target applications (MTA)

Related Information

SAP Software Download Center (Logon required)
Logging on and Getting Started with the XS CLI [page 1265]
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)

14.2.2.1.2 Logging on and Getting Started with the XS CLI

Get to know the basic commands available with the `xs` command-line interface.

In this section, you learn how to use the `xs` command-line interface (CLI) to perform the following tasks:

- Display a list of all `xs` commands
- Display details of a specific `xs` command
- Log on to (and out of) an instance of SAP HANA XS, advanced model

> → Tip
>
> For information about predefined SAP HANA XS advanced users, for example, system and technical users, see *User Administration and Authentication in SAP HANA XS Advanced* in the *SAP HANA Security Guide*.

Getting Help

The `xs` command-line interface provides comprehensive help for each `xs` command. To display an overview of all available `xs` commands included with the `xs` CLI, use the help command as shown in the following example:

```
$ xs help -a
```

To display usage details of a specific `xs` command, add the name of the command to the help command as shown in the following example:

```
$ xs help <command>
```

To display details of the current version of the `xs` command-line client, use the `version` command as shown in the following example:

```
$ xs version
```

Logging on

During logon the `xs` CLI tries to connect to the so-called API URL of the XS Controller. To find out the current API URL, log on to the SAP HANA system as the operating-system user `<SID>adm` and run the following command:

```
$ xs-admin-login --api
https://api.example.org:31030
```

You can use the URL displayed in the command output to log on to the XS advanced instance with the `xs` CLI from a remote session:

> → Tip
>
> If you do not specify any user name in the `xs logon` command, the `xs` CLI prompts you for the logon credentials you want to use.

```
$ xs login -a https://api.example.org:31030
USERNAME> XSA_ADMIN
PASSWORD> *********
Authenticating...
ORG: myorg
```

By default, the logon process targets the organization that was created during installation, for example, `myorg`. The first time you log on to XS advanced, you are asked to specify the organizational space you want to log on to from a list of available spaces, as illustrated in the following example:

```
Existing spaces:
0. PROD
1. SAP
SPACE> 0
SPACE: PROD
```

Choose the space you want to log on to by entering the corresponding number in the list, for example, `0` to select `PROD` as the target logon space. After successful logon, XS advanced displays a short summary of the logon details, as illustrated in the following example:

```
API endpoint:  https://api.example.org:31030 (API version: 1)
User:          XSA_ADMIN
Org:           myorg
Space:         PROD
```

If used remotely, the `xs` client needs a trust certificate to establish a secure connection. For more information about obtaining this trust certificate, see Trust Certificates [page 1267].

Alternatively, if you fully trust the network connection to the server, you can skip the SSL validation process, as shown in the following example:

> ⚠ Caution
>
> It is strongly recommended **not** to skip SSL validation in a production environment.

> {·} Sample Code
>
> ```
> xs login -a <API URL> -u <username> --skip-ssl-validation
> ```

> → Tip
>
> If you already logged on to an SAP HANA system as the operating-system user `<SID>adm`, you can use the command `xs-admin-login` (without any parameters or options) as a shortcut to log on to the XS advanced run time as user `XSA_ADMIN`. The `xs-admin-login` command automatically sets up the `xs` CLI and logs you on as `XSA_ADMIN` after prompting for the corresponding password.

Logging in via an External Identity Provider (IDP)

If the log-in user resides in an external IDP, you can use the option `--sso` to log in with a one-time password that is generated during the login process. As illustrated in the following example, the command output displays a link to the user-account and authentication (UAA) page that provides the one-time password you need to log in to XS advanced. To receive the one-time password, you need to log in, for example, via an external IDP.

```
$ xs login -sso

API_URL: https://api.test.example.com:30033
One Time Code ( Get one at https://uaa-server.test.example.com:30033/uaa-
security/passcode)>
```

The `One Time Code` line in the output provides a link to a `uaa-server`. When you click the link, you are forwarded to the UAA log-in page. On the UAA log-in page, you can choose the *SAML* option, which is a link that forwards you to the IDP log-in page. If you are logged in successfully, you will be redirected to the page that will show you the one-time password in the form of a *Temporary Authentication code*, which you can copy and then paste into the command-line terminal session where you are running the `xs login` command.

Trust Certificates

If you log on to XS advanced from a remote machine, the `xs` CLI requires a trusted certificate to establish a secure connection, unless `https` is disabled or a certificate signed by a well-known CA is used at the XS Controller. For your convenience, a public trust certificate is stored on the SAP HANA system at the following location:

`/hana/shared/<SID>/xs/controller_data/controller/ssl-pub/router/default.root.crt.pem`

The certificate file can be copied to the client, where it can then be consumed by using the `--cacert` option, as illustrated in the following example:

```
$ xs login -a https://api.example.org:8080 --cacert <PATH>
```

`<PATH>` indicates the location on the client machine of the file `default.root.crt.pem` which contains the certificate to use to establish the trusted connection.

> → Tip
>
> Neither of these steps is necessary if a certificate signed by a well-known CA is available. For more information about certificates and trusted connections, see *Maintaining Domains* in *Related Information*.

Logging off

To log out of the `xs` command-line interface client session, use the the `logout` command as shown in the following example:

```
$ xs logout
```

Related Information

Maintaining Trust Certificates in XS Advanced [page 1321]
Maintaining Domains and Certificates in XS Advanced [page 1324]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
SAP HANA Security Guide

14.2.2.1.3 Displaying the System Overview

Display an overview of the current configuration and status of the main XS advanced components.

If you are logged on to the XS advanced platform with the `xs` command-line interface, you can display an overview of all the most important components of the XS advanced instance, for example, the Execution Agents and the XS Advanced Controller, as illustrated in the following example of the `xs system-info` command:

Output Code

Displaying XS Advanced Information

```
$ xs system-info
Getting system infrastructure information...
```

Restriction

To execute the `xs system-info` command, you need XS advanced administrator permissions.

The output generated by the `xs system-info` command includes information about the following XS advanced components:

- XS Controller Server Version [page 1269]
- Execution Agents [page 1271]
- Registered Service URLs [page 1271]
- Applications [page 1272]
- Services [page 1272]
- Organizations And Spaces [page 1273]

XS Advanced Controller

The command `xs system-info` displays information about the XS Controller, as shown in the following example:

Output Code

`xs system-info` Command Output

```
Controller server version information:
------------------------------------------------------------------------------
---
name                      XS Controller
support                   http://service.sap.com/message
build                     v1.0.150
api version               1
software version          1.1.0
core version              1.1.0
content version           1.1.0
state                     READY
description               SAP HANA XS Runtime on premise
controller endpoint       https://example.org:30030
authorization endpoint    https://example.org/uaa-security
accept encoding           gzip, x-gzip
usage                     apps: 6, routes: 4, services: 16
                          app instances: 0 starting, 5 running, 7 stopped, ...
                          0 crashed, 0 timed out
database type             HANA_MULTI
database info             HDB 2.00.059.04.1655794895
runtime database          SYSTEMDB
port range                51000-51500
used ports                49 of 501 (27 by XSA apps, 22 reserved for SAP
services)
```

The following table lists and explains the information that the `xs system-info` command displays about the XS Controller.

XS Controller Information

Information	Description
`name`	The name of the XS advanced component
`support`	The URL to use to contact a support team
`build`	The built version of the XS Controller
`api version`	The version of the REST interface provided by the XS Controller
`software version`	The installed version of the XS advanced system components. **Note** In a standard installation, `software version` and `content version` must match.

Information	Description
`content version`	The installed version of the XS advanced core applications package (for example, `deploy-service` and `product-installer`). **Note** In a standard installation, `software version` and `content version` must match.
`state`	The current state of the XS advanced run-time, which can be one of the following: • `STARTUP` XS advanced is currently starting • `READY` XS advanced is installed correctly • `INSTALLATION_PENDING` The initial XS advanced installation is not yet complete • `UPDATE_PENDING` The XS advanced update is not yet complete
`description`	A short summary of the type of installed component, for example, `SAP XS Runtime on premise`
`controller endpoint`	The REST end point for requests to the XS Controller API (also known as the "API URL")
`authorization endpoint`	The REST end point for authorization requests
`usage`	Information about the number of applications and application instances and their current status, for example: `starting`, `running`, `stopped`, etc.
`database type`	The type of the underlying SAP HANA database installed; this can be either of the following: • `HANA_MULTI` SAP HANA multiple database-container system • `HANA_SINGLE` SAP HANA single database-container system
`database info`	The version of the underlying SAP HANA database
`runtime database`	The database containing the persistence of the XS Advanced system components
`port range`	The current router port range defined in the `xscontroller ini` file with the properties `router_portrange_start` and `router_portrange_end`. For more information, see *XS Advanced System Configuration Parameters* in *Related Information* below.
`used ports`	The current utilization of the router port range

XS Advanced Execution Agents

The Execution Agents section of the information displayed by the `xs system-info` includes details of the execution agents that are currently connected to the XS advanced Controller, as displayed in the following example output:

`xs system-info` command output

```
Execution agents:
------------------------------------------------------------------------------
index          1
host           host1.example.org:29881
created at     21-Feb-2023 10:18:51
port range     50000-50999
used ports     66 of 1000 (26 by XS advanced apps, 40 reserved for SAP services)
os.arch        amd64
os.name        Linux
os.version     5.3.18-59.24-default
version        v1.1.0
index          2
host           host2.example.org:23456
created at     21-Feb-2023 10:18:51
port range     50000-50999
used ports     63 of 1000 (24 by XS advanced apps, 40 reserved for SAP services)
os.arch        amd64
os.name        Linux
os.version     5.3.18-59.24-default
version        v1.0.150
```

In this example, two execution agents on "`host1.example.org`" and "`host2.example.org`" are connected to schedule application instances on. For each execution agent, additional information is displayed as followed:

- The host on which the Execution Agent is running
- The date and time when the Execution Agent was started
- The port range the Execution Agent uses for applications on its host
- The platform and version of the SAP Java Virtual Machine (JVM) that is running the Execution Agent
- The version of the Execution Agent

> → Tip
>
> If you need to scale out your application, you can add additional execution agents to an SAP HANA system by adding additional SAP HANA hosts with role "xs_worker".

XS Advanced Service URLs

The `xs system-info` command displays information about the services URLs registered by deployed XS advanced applications. XS advanced applications can maintain a service URL during installation to publish their application end point:

> {·} Output Code
>
> `xs system-info` Command Output
>
> ```
> Registered service URLs:
> ```

```
------------------------------------------------------------------------------
   deploy-service         https://example.org:51004
   product-installer      https://example.org:51005
   [...]
```

XS Advanced Applications Overview

The `xs system-info` command uses the "`Applications`" section to display statistics about the applications deployed in the XS advanced system and the current known status, as illustrated in the following example output:

Output Code

`xs system-info` Command Output

```
Applications:
------------------------------------------------------------------------------
   Deployed applications: 32
   Running applications:  26
   Stopped applications:  6
```

XS Advanced Services Overview

The `xs system-info` command uses the "`Services`" section to display statistics about the services deployed in the XS advanced system and the current known status, as illustrated in the following example output:

Output Code

`xs system-info` Command Output

```
Services:                   total   bound   unbound
------------------------------------------------
   Managed services:           34      33         1
   User provided services:      1       1         0
```

Organizations And Spaces in XS Advanced

The `Organizations and Spaces` section of the information displayed by the `xs system-info` command displays statistics about the distribution of applications across the configured organizations and spaces, as illustrated in the following example:

Output Code

`xs system-info` Command Output

```
Organizations and spaces:
   org      space    user       apps    alerts    services    bound    unbound
s.keys
-------------------------------------------------------------------------------
----
   myorg
   |-       PROD     xsaxsa       5                   4        4
0        0
   |-       SAP      sapxsaxsa    27                  31       30
1        0
-------------------------------------------------------------------------------
----
   total:                         32                  35       34
1        0
```

In the example displayed, the organization "`myorg`" contains two spaces named "`PROD`" and "`SAP`". In addition to the number of applications deployed in each space, the information displayed also indicates the name of the operating-system user (for example, `<SID>xsa`) used to start the XS advanced application.

Related Information

Maintaining Services in XS Advanced [page 1305]
Maintaining the XS Advanced Run-time Environment with a Command-Line Interface [page 1261]
XS Advanced System Configuration Parameters [page 1480]

14.2.2.1.4 Maintaining Organizations and Spaces in XS Advanced

Use the `xs` command-line interface to configure and maintain organizations and spaces in XS advanced.

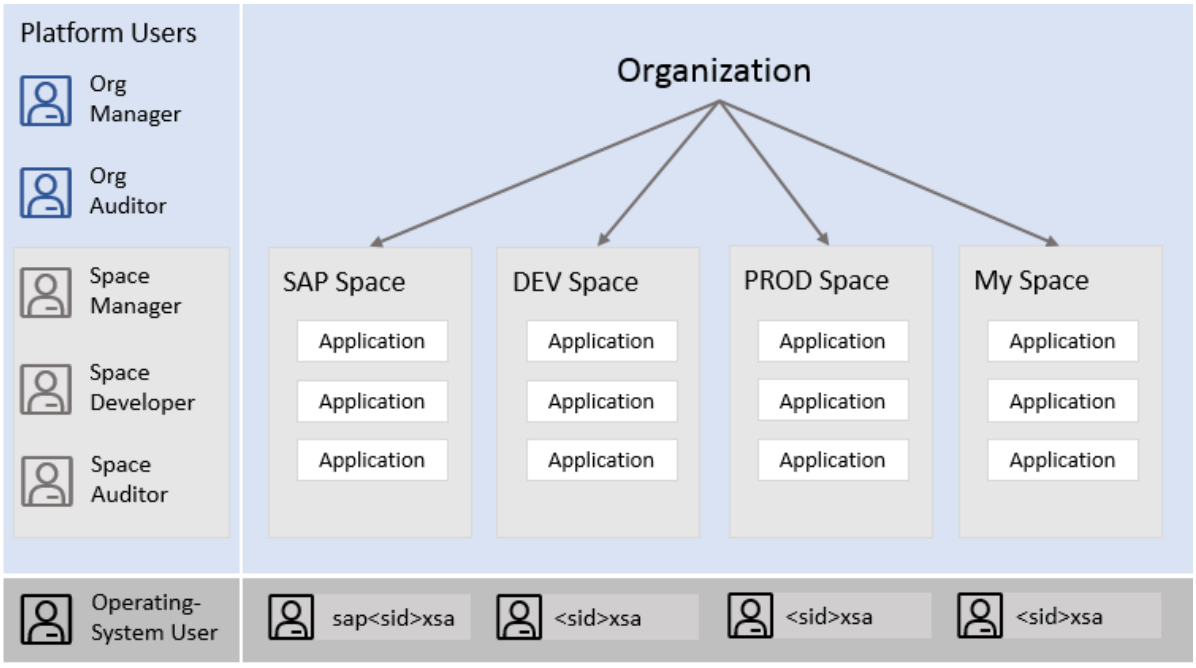
The examples in this section show how to use the `xs` command-line interface (CLI) to perform the following tasks:

- Navigating Through Organizations and Spaces [page 1274]
- Maintaining Spaces [page 1276]
- Maintaining Organizations [page 1276]
- Isolating Applications at the Operating-System Level [page 1277]
- Custom Operating-System Users [page 1278]

Overview

XS advanced applications can be grouped together by means of a hierarchy of so-called organizations and spaces. An organization forms a logical group of spaces. Information about application domains and server certificates is also tied to organizations. In XS advanced, a "space" is not only a way to form a logical group of applications, it is also a trust zone. Applications cannot use any resources that were created in another space (for example, service instances), and application processes are also isolated at the space level by using operating-system users to start the application processes, as described in Custom Operating-System Users [page 1278] below.

> → Tip
>
> See *Maintaining Platform Users* in *Related Information* for more information about user roles in organizations and spaces. For more information about the underlying security concepts for organizations and spaces, see the *SAP HANA Security Guide* in *Related Information* below.



Navigating Through Organizations and Spaces in XS Advanced

During the installation of XS advanced, one organization and two spaces are created by default:

- An organization with a custom name
- A space named `SAP` which contains the default XS advanced applications provided by SAP
- A space for custom development or deployment; by default, this space is named `PROD`.

To display a list of all organizations configured in an XS advanced run-time instance, run the following command, which in this example, shows the organizations `acme` and `corp`:

```
$ xs orgs
```

```
Getting orgs...

name
-----
acme
corp
```

To view the spaces configured in the currently targeted organization, use the command `xs spaces`, as shown in the following example:

```
$ xs spaces
Getting spaces in org acme as XSA_ADMIN ...
name   user
---------------
DEV    xsaxsa
PROD   xsaxsa
SAP    sapxsaxsa
```

> → Tip
>
> The space `name` is shown along with the name of the operating-system `user` whose account is used to start applications in the specified space, for example, the user `xsaxsa` starts applications in spaces `DEV` and `PROD`, and the user `sapxsaxsa` starts applications in space `SAP`.

You can set the target for the `xs` CLI to a specific organization and space by using the `xs target` command, as shown in the following example:

```
$ xs target -o <org-name> -s <space-name>
```

When you have set the target organization and space, you can use the `xs apps` command to display a list of the applications currently running there:

```
$ xs apps

Getting apps in org "acme" / space "SAP" as XSA_ADMIN...
Found apps:

name                  state     instances memory   disk         urls
-------------------------------------------------------------------------------
auditlog-db           STOPPED   0/1       16.0 MB  <unlimited>  <none>
auditlog-server       STARTED   1/1       256 MB   <unlimited>  https://
acme.org:51003
auditlog-broker       STARTED   1/1       64.0 MB  <unlimited>  https://
acme.org:51004
deploy-service        STARTED   1/1       280 MB   <unlimited>  https://
acme.org:51005
component-registry-db STOPPED   0/1       16.0 MB  <unlimited>  <none>
product-installer     STARTED   1/1       256 MB   <unlimited>  https://
acme.org:51006
xsa-cockpit           STARTED   1/1       128 MB   <unlimited>  https://
acme.org:51007
```

To display details of the currently targeted organization and space, use the command `xs target` without any parameters or options, as illustrated in the following example:

```
$ xs target

API endpoint: https://xsa.acme.com:30030 (API version: 1)
User:         XSA_ADMIN
Org:          acme
```

```
Space:          SAP
```

To change to a specific target organization and space, use the `xs target` command with the options `-o <ORG> -s <SPACE>`, as illustrated in the following example:

```
$ xs target -o acme -s DEV
```

Maintaining Spaces in XS Advanced

In some advanced use cases, the default space setup is not sufficient to meet the needs of your environment. For example, if you want to separate applications currently in development from applications that have been tested (so that they run in different spaces), you can create additional spaces, for example, named `DEV` and `TEST`. Similarly, you can separate different development teams by ensuring that they work in different spaces. To create a new space, user the command `xs create-space`, as shown in the following example:

```
$ xs create-space DEV

Creating space DEV in org acme as XSA_ADMIN...
OK
```

The command shown in the example above will create an empty space named `DEV` within the currently targeted organization `acme`.

> → Tip
>
> The `xs create-space` command does not automatically switch target to the new space `DEV`; you must manually set the new space as the new target, for example, with the command `xs target -s <SPACE>`.

To delete an existing space, use the command `xs delete-space`, as shown in the following example:

```
$ xs delete-space DEV

Really delete the space "DEV" in org "acme"? (y/n) > y
Deleted space "DEV".
```

You can change the name of an existing space, too, as illustrated in the following example, which shows how to change the name of the space "DEV" to "TEST":

```
$ xs rename-space DEV TEST

Renaming space "DEV" in org "acme"...
OK
```

Maintaining Organizations in XS Advanced

Organizations are particularly useful if you want to expose one group of applications by means of a different domain name to the one used used by another group of applications. Hence, organizations can be used

to reflect the organizational structure of your company or corporation. In order to create a new, empty organization, use the command "xs create-org":

```
$ xs create-org myorg
OK
```

> → Tip
>
> The `xs create-org` command does not automatically switch target to the new organization `myorg`; you must manually set the new organization as the new target, for example, with the command `xs target -o <ORGANIZATION>`.

To delete an existing organization, use the command `xs delete-org`, as shown in the following example:

```
$ xs delete-org myorg

Deleting org "myorg"...
Really delete the org "myorg"? (y/n) > y
Deleted org "myorg".
```

> ⚠ Caution
>
> The `xs delete-org` command also deletes **all spaces** contained in the deleted organization along with all related resources such as applications, service instances, and routes. Note that all applications running in any of the deleted organization's spaces must be stopped before you delete the organization.

You can change the name of an existing organization, too, as illustrated in the following example, which shows how to change the name of the organization "acme" to "test":

```
$ xs rename-org acme test
OK
```

Isolating XS Advanced Applications at the Operating-System Level

Instances of running application can be isolated at the space level. This means that processes of application instances started in the same space are run as the same operating system user. You can also specify that application processes from different spaces are executed as different operating system users. In this way, application processes from an isolated space are not allowed to access the process environment and the file system of other applications or the SAP HANA system itself.

During installation, two restricted operating system users are automatically created by `hdblcm` and assigned to default XS advanced spaces. These users are automatically maintained by `hdblcm` if an additional xs_worker host is added to the SAP HANA system or during any rename operation. The following table shows the mapping of the default operating system users to XS advanced spaces;

Default Username-Space Mapping in XS Advanced

User Name	Space Name
sap<SID>xsa	SAP

User Name	Space Name
<SID>xsa	Custom space (for example, PROD); this is the default user name for newly created spaces

In this example, application processes in the space SAP run as user sap<SID>xsa while application processes in the space PROD will run as user <SID>xsa. User names are combined with the System ID (SID) of the SAP HANA system. This is to prevent any clashes between user names when several SAP HANA systems are installed on the same host. To change the operating system user mapping for a particular space, use the command `xs update-space`, as shown in the following example:

```
$ xs update-space <space-name> -u <OS-user-name>
```

> **Note**
>
> You cannot change the mapping between operating-system user and a space if applications are still running inside the targeted space. You must stop all applications that are running in a space before changing the operating-system user mapping.

To revert to the default settings for the mapping between space and operating-system user, use the --unset-user option as shown in the following example:

```
$ xs update-space <space-name> --unset-user
```

Custom Operating-System Users

The command `xs update-space` does not create the underlying operating system user. If a custom operating system user is required in addition to the default operating system users, the new custom operating-system user must be created first and on all SAP HANA hosts performing the role of `xs_worker` or `xs_standby`.

> **Note**
>
> The operating-system user associated with a space only requires a restricted set of privileges.

The operating-system user associated with a space only requires a restricted set of privileges:

- The operating system user does **not** need any elevated or superuser privileges
- The operating system user must **not** be in the group `sapsys`
- No secondary groups are required
- No `home/` directory is required
- No user logon is required

In addition, if you use a SAP HANA version older than HANA 2.0 SPS 02 Rev. 1, it is necessary to adapt the `sudo` configuration file `/etc/sudoers`. For more details about how to set up a new operating system user and adapt the `sudoers` configuration files, see also SAP Note 2243156.

Related Information

Maintaining Platform Users in XS Advanced [page 1281]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]
SAP HANA Security Guide

14.2.2.1.5 Maintaining Organization Quotas

Set limits for application instances for XS advanced organizations and maintain organizational quota plans.

Organization quotas are named sets of restrictions applied to an organization by an organization manager. For example a quota might limit an organization to create up to 10 application instances. Because organization quotas are a collection of such individual limits they are also referred to as quota plan, quota definition or simply quotas. The limits supported by organization quotas are shown in the table below.

XS Advanced Organization Limits

Limit	Description
App instance limit	The total number of instances of started applications allowed in the specified organization. Instances of stopped applications do not count toward this instance limit. **Note** "-1" represents an unlimited amount. (Default: "-1")
Max instances per application	The total number of running instances allowed per application in the specified organization. **Note** "-1" represents an unlimited amount. (Default: "100")
Service instances limit	The total number of service instances allowed in the specified organization. **Note** "-1" represents an unlimited amount. (Default: "-1")

When a new organization is created, an organization quota named "default" is automatically assigned to it; this is referred to as the "default quota". Although the default quota cannot be deleted, the values can be modified by an administrative user. Only an XS advanced controller administrator can use the quota commands to create, update, or delete an organization quota. An organization manager can use the quota commands to assign a quota to an organization.

Creating a Quota Plan for an Qrganization

The following example shows how to use the `xs create-quota` command to set a quota plan for the current target organization:

```
xs create-quota quota1 -a 10 -ma 5 -s 4
```

The command illustrated in the example above creates a quota named "`quota1`" that limits the total amount of started application instances (-a) to “10”; the maximum amount of running instances per application (-ma) to “5”; and the total amount of service instances (-s) to “4”. Where no option is specified, the default value for the respective limit is assigned.

Reading an Qrganization Quota Plan

The following example shows how to use the `xs quotas` command to display a list of all existing organization quota plans:

```
xs quotas

Getting Quota Definitions...

name      app instances   max instances per app   service instances
------------------------------------------------------------------
default   unlimited       100                     unlimited
quota1    10              5                       4
```

To display details of a specific quota, use the `xs quota <QUOTA_NAME>` command, as illustrated in the following example:

```
xs quota quota1

Property                  Value
----------------------------------
Name                      quota1
Service instances limit   unlimited
App instance limit        unlimited
Max instances per app     100
```

Assigning a Quota Plan to an Qrganization

The following example shows how to use the `xs set-quota` command to assign a quota plan to a specific organization:

```
xs set-quota org1 quota1
```

The command illustrated in the example above assigns the quota named "`quota1`" to the organization named "`org1`". Assigning of a quota to an organization will fail if the limits set in the assigned quota are already being broken by the organization. For example if the organization has 100 application instances, then a quota with a total application instance limit of “10” cannot be assigned.

> **Note**
>
> A quota can be assigned to multiple organizations. Any changes made to the quota will affect all the organizations it is assigned to.

To view the quota definition assigned to an organization use the command "xs org". The example below demonstrates an elaborate use:

```
xs org org1

Getting info for org "org1" as user "XSA_ADMIN"...

name:     org1
domains:  example.org
quota:    quota1
spaces:   space1
created:  Feb 13, 2023 8:55:36 AM
updated:  Feb 13, 2023 8:55:36 AM
```

Deleting a Quota Plan

The following example shows how to use the `xs delete-quota` command to unassign and remove a quota plan:

```
xs delete-quota quota1
```

The command illustrated in the example above deletes the quota named "`quota1`".

> **Restriction**
>
> An organization quota can only be deleted if it is not assigned to any organization. The quota named "default" cannot be deleted.

Related Information

Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
Maintaining Organizations and Spaces in XS Advanced [page 1273]

14.2.2.1.6 Maintaining Platform Users in XS Advanced

Use the `xs` command-line interface to maintain XS advanced users.

Users who have access to the XS advanced platform (so-called platform users) can be sorted into the following categories:

- Platform administrators

For example, `XSA_ADMIN`, who are allowed to perform any platform operation in any organization and space.

- Restricted users
  For example, developers and organization or space manager, who have restricted permissions in specific organizations or spaces. A role model exists for restricted platform users, which is defined in the *SAP HANA Security Guide*. For more details about this user model, see *Related Information*.

Maintaining SAP HANA Users

You can use SAP HANA database users as XS advanced users. By default, SAP HANA database users are assigned the role `XS_PUBLIC`. The `xs` command-line interface provides commands that enable you to maintain XS Advanced SAP HANA users. To create a new database user with XS advanced capabilities, use the command `xs create-user`, as illustrated in the following example:

> → Tip
>
> The option --platform automatically assigns the role collection `XS_CONTROLLER_USER` described in *Predefined Role Collections* below.

```
xs create-user <USERNAME> --platform
```

XS advanced users can change their own password by logging in to the XS advanced platform with the `xs` CLI and running the command `xs passwd`, as illustrated in the following example:

```
xs passwd
```

XS advanced administrators can change or reset the password of other users by specifying the name of the user whose password they want to change, as illustrated in the following example:

```
xs passwd <OTHER_USER>
```

XS administrators can also use the `xs` CLI to delete a user of the XS advanced platform, as illustrated in the following example:

```
xs delete-user <OTHER_USER>
```

To complete the configuration of SAP HANA users as XS advanced users, you can use the `xs` CLI to assign one or more predefined role collections to the XS advanced users as described in the following sections.

Viewing Role Collections

New XS advanced platform users are created by assigning predefined platform role collections to existing SAP HANA users, for example, with the `xs` CLI, as described in this section. The following table shows the XS advanced role collections that exist for platform users:

> ### Note
>
> You can also use IdP users as XS advanced platform users. However, it is not possible to assign role collections to IdP users, you must map a SAML group, instead.

XS Advanced Role Collections

Role Collection	Description
`XS_CONTROLLER_ADMIN`	The platform administrator
`XS_CONTROLLER_ADMIN_READ_ONLY`	The platform administrator with read only permissions
`XS_CONTROLLER_GLOBAL_AUDITOR`	Similar to the platform administrator but only with some restrictions like system configuration and credentials access
`XS_CONTROLLER_USER`	Enables a restricted user who has refined permissions in organizations and spaces
`XS_CONTROLLER_CREDENTIALS_VIEWER`	Similar to the controller user with additional application credentials access
`XS_CONTROLLER_AUDITOR`	Enables a restricted read-only user who has refined permissions on organizations and spaces
`XS_USER_ADMIN`	Enables a user who can create and maintain other users
`XS_AUTHORIZATION_ADMIN`	Enables a user who can view, create and maintain role collections
`XS_MONITOR_ADMIN`	Enables a user who can set or change the trace level.

Viewing Available Role Collections

You can use the command `xs role-collections` to view all available XS advanced role collections. The command output shows details of role collections for the XS advanced platform as well as the role collections created for XS advanced applications, as illustrated in the following example:

> ### Sample Code

```
$ xs role-collections

Getting role collections as user "XSA_ADMIN"...

role collection                  description
-------------------------------------------------------------------
AUDITLOG_VIEWER
XS_AUTHORIZATION_ADMIN           Authorizations for XS role builder
XS_AUTHORIZATION_DISPLAY         Authorizations for XS role viewer
XS_USER_ADMIN                    Admin authorizations for XS user management
XS_USER_DISPLAY                  Display authorizations for XS user
management
```

```
XS_USER_PUBLIC                    Default authorizations for XS user
XS_MONITOR_ADMIN                  Authorizations for XS monitoring management
XS_MONITOR_DISPLAY                Authorizations for XS monitoring display
XS_CONTROLLER_ADMIN               Authorizations for XS controller admin
XS_CONTROLLER_ADMIN_READ_ONLY     Authorizations for XS controller read-only
admin
XS_CONTROLLER_GLOBAL_AUDITOR      Authorizations for XS controller global
auditor
XS_CONTROLLER_USER                Authorizations for XS controller user
XS_CONTROLLER_AUDITOR             Authorizations for XS controller auditor
XS_CONTROLLER_CREDENTIALS_VIEWER  Authorizations for XS controller
credentials viewer
```

Viewing Assigned Role Collections

To view the role collections assigned to a specific user, use the command `xs assigned-role-collections`, as illustrated in the following example for user `XSA_ADMIN`:

Output Code

Role Collections Assigned to a User

```
$ xs assigned-role-collections XSA_ADMIN

role collection           description
-------------------------------------------------------------------
XS_AUTHORIZATION_ADMIN   Authorizations for XS role builder
XS_CONTROLLER_ADMIN      Authorizations for XS controller admin
XS_MONITOR_ADMIN         Authorizations for XS monitoring management
XS_USER_ADMIN            Admin authorizations for XS user management
AUDITLOG_VIEWER
```

In the example output above, the default admin user `XSA_ADMIN` has a role collection `XS_CONTROLLER_ADMIN` that enables the administrator to perform **any** task in **any** XS advanced organization and space. In addition, `XSA_ADMIN` has the role collections `XS_AUTHORIZATION_ADMIN` and `XS_USER_ADMIN`, which enable the administrator user to create other users and maintain and assign role collections. `XSA_ADMIN` has an additional application role collection that provides the permissions required to view audit logs.

Assigning Role Collections

The following sections describe how to create new administrators and restricted users for the XS advanced platform by assigning the corresponding role collection and setting permissions that apply for particular organizations and spaces.

Creating XS Advanced Administrator Users

To create a new XS advanced administrator user, it is necessary to have an existing "non-restricted" SAP HANA user in the database where XS advanced keeps its persistence. The following example shows how to use the `hdbsql` utility to create an SAP HANA user. Log on to the SAP HANA system with `hdbsql` as a user with the `User.Admin` role and execute the following SQL command:

```
=> CREATE USER XSA_ADMIN2 PASSWORD "Welcome1"
0 rows affected (overall time 25.359 msec; server time 23.375 msec)
```

If you are logged to the XS advanced platform as an XS advanced administrator (for example, `XSA_ADMIN`), you can grant the new XS advanced administrator user (`XSA_ADMIN2`) the necessary controller roles `XS_CONTROLLER_ADMIN`, `XS_USER_ADMIN`, and `XS_AUTHORIZATION_ADMIN`, for example, using the command `xs assign-role-collection`:

Sample Code

```
$ xs assign-role-collection XS_CONTROLLER_ADMIN XSA_ADMIN2

Assigning role collection "XS_CONTROLLER_ADMIN" to user "XSA_ADMIN2"...
OK
```

```
$ xs assign-role-collection XS_USER_ADMIN XSA_ADMIN2

Assigning role collection "XS_USER_ADMIN" to user "XSA_ADMIN2"...
OK
```

```
$ xs assign-role-collection XS_AUTHORIZATION_ADMIN XSA_ADMIN2

Assigning role collection "XS_AUTHORIZATION_ADMIN" to user "XSA_ADMIN2"...
OK
```

The `XSA_ADMIN2` can be now used to perform administrator tasks on the XS Advanced platform.

Restriction

To log on to the new XS advanced administrator account (`XSA_ADMIN2`) an initial password change is required. Once logged in, you are prompted by the xs CLI or the UAA login page to change your initial password. Alternatively, you can use the command `hdbsql` to set the new password.

Creating XS Advanced Restricted Users

To create a new XS advanced restricted user, it is necessary to have an existing SAP HANA user in the database where XS advanced keeps its persistence. The following example shows how to use the `hdbsql` utility to create an SAP HANA user. Log on to the SAP HANA system with `hdbsql` as a user with the `User.Admin` role and execute the following SQL command:

```
=> CREATE RESTRICTED USER NEW_XSA_USER PASSWORD "Welcome1"
0 rows affected (overall time 26.582 msec; server time 24.367 msec)
```

Note, that for XS advanced restricted users, restricted HANA users (without `PUBLIC`role assignment) are sufficient.

As an alternative, you can also execute `xs create-user` as XS advanced administrator:

```
=> xs create-user NEW_XSA_USER --platform
PASSWORD>
Checking if user "NEW_XSA_USER" already exists (as "XSA_ADMIN")...
Creating user "NEW_XSA_USER" as "XSA_ADMIN"...
Password for user NEW_XSA_USER>
Confirm the password>
OK
```

Note, that by specifying the option --platform the role collection `XS_CONTROLLER_USER` is automatically assigned. For more information about `xs create-user`, see *XS CLI: User Administration* in the *Related Links*.

If you are logged on to the XS advanced platform as an XS advanced administrator (for example, XSA_ADMIN), you can grant the new XS advanced user (NEW_XSA_USER) the necessary controller roles XS_CONTROLLER_USER manually, for example, using the command `xs assign-role-collection`:

```
$ xs assign-role-collection XS_CONTROLLER_USER NEW_XSA_USER

Assigning role collection "XS_CONTROLLER_USER" to user "NEW_XSA_USER"...
OK
```

The assignment of the role collection XS_CONTROLLER_USER enables the new user to log on to the XS advanced platform. Since the created user is restricted, the user still does not have any XS Controller privileges. For this reason, it is necessary to assign XS Controller roles to the restricted XS advanced user at both the organization and space level, as described in the following sections.

Assigning Controller Roles

XS administrators can set roles and permissions at the organization level or at the space level.

Setting Roles and Permissions at the Organization Level

The `xs` command-line client includes the command `xs set-org-role`, which you can use to grant organization-specific roles to users. For example, to grant the user XSA_USER the `OrgManager` role in the organization "acme", run the following command:

```
$ xs set-org-role XSA_USER acme OrgManager

OrgManager Adding role 'OrgManager' to user NEW_XSA_USER in org "acme" ...
OK
```

To set user permissions at the organization level, you must log on to XS advanced with either administrator privileges or as a restricted user with the role `OrgManager` in the organization where you want to grant user roles and privileges. The following table lists the roles you can grant to an organization user:

XS Advanced Organization Roles

Organization Role	Privileges
`OrgManager`	Create and modify spaces in an organization Assign roles and privileges to other organization users
`OrgAuditor`	Browse through spaces inside an organization

For users maintained in an external IdP, the `--origin` option enables you to specify the name of the IdP where the external user is maintained while setting the organization role, for example, with the `xs set-org-role` command. The following example shows how to specify that the user XSA_USER comes from the IdP origin "accounts":

```
$ xs set-org-role XSA_USER acme OrgManager --origin accounts
```

To view the users available for an organization, use the command `xs org-users`. The command displays user name, the assigned role, origin and the deactivation state, as illustrated in the example below:

```
$xs org-users acme
Listing users in org "acme" by role ...
role         user        origin   deactivated
-----------------------------------------
OrgManager   TESTUSER1           yes
```

To remove a role already assigned to an organization user, use the `xs unset-org-role` command, as illustrated in the following example:

```
$ xs unset-org-role XSA_USER acme OrgManager
```

Setting Roles and Permissions at the Space Level

The `xs` command-line client includes the command `xs set-space-role`, which you can use to grant space-specific roles to users. For example, to grant the user `NEW_XSA_USER` the `SpaceManager` role for a space called "DEV" in an organization named "acme", run the following command:

```
$ xs set-space-role NEW_XSA_USER acme DEV SpaceManager

Adding role 'SpaceManager' to user NEW_XSA_USER in space "DEV" in org "acme" ...
OK
```

To set user permissions at the space level, you must log on to XS advanced with either administrator privileges or as a restricted user with the role `SpaceManager` in the space where you want to grant user roles and privileges. The following table lists the roles you can grant to a space user:

XS Advanced Organization Roles

Space Role	Privileges
`SpaceManager`	Modify a space Assign roles and privileges to other space users
`SpaceDeveloper`	Modify a space
`SpaceAuditor`	Browse through any space where you have the `SpaceAuditor` role

For users maintained in an external IdP, the `--origin` option enables you to specify the name of the IdP where the external user is maintained while setting the space role, for example, with the `xs set-space-role` command. The following example shows how to specify that the user `XSA_USER` comes from the IdP origin "accounts":

```
$ xs set-space-role XSA_USER acme SpaceManager --origin accounts
```

To view the users in a space, use the `xs space-users` command. The command displays user name, the assigned role, origin and the deactivation state of the space user, as illustrated in the example below:

```
$ xs space-users acme SAP
Listing users in space "SAP" of org "acme" by role ...
role           user        origin   deactivated
-----------------------------------------------
SpaceManager   TESTUSER2
```

To remove a role already assigned to a space user, use the `unset-space-role` command, as illustrated in the following example:

```
$ xs unset-space-role XSA_USER acme DEV SpaceManager
```

To view all the users on the platform that have been assigned a role, use the `xs users` command. The command shows the name, origin and the deactivation state of the users, as illustrated in the example below:

```
$ xs users
Getting users...
user        origin   deactivated
-------------------------------
TESTUSER1            yes
TESTUSER2
```

> ⓘ Note
>
> Users marked as '`orphaned`' are SAP HANA database users that have been deleted from the database but are still known to the XS Controller. To clean these users up and remove their connection to the XS Controller, run the command `xs purge-users`.

Related Information

Building Roles for XS Advanced Applications [page 1346]
Maintaining Organizations and Spaces in XS Advanced [page 1273]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
SAP HANA Security Guide
XS CLI: User Administration
Map Role Collections to an SAML Identity Provider [page 1428]

14.2.2.1.7 Displaying Application Information in XS Advanced

Use the `xs` command-line interface to deploy and maintain XS advanced applications.

After navigating to a particular organization and space, you can explore the resources contained in a space. A space contains different types of resources, for example: applications, service instances, and routes. The information in the following sections focuses on application resources and aims to help you perform the following application-related tasks in an XS advanced space:

- Displaying Deployed XS Advanced Applications in a Space [page 1289]
- Displaying Details of an Individual XS Advanced Application [page 1290]
- Displaying XS Advanced Application Logs [page 1292]
- Displaying XS Advanced Application Events [page 1294]
- Displaying XS Advanced Application Files [page 1295]

Displaying Deployed XS Advanced Applications in a Space

When you set a space as a particular "target"`xs` CLI to display information about the XS advanced applications running in the target space, as shown in the following example:

Output Code

```
$ xs apps

Getting apps in org "acme" / space "SAP" as XSA_ADMIN...
Found apps:
                    requested
name                state      instances memory   disk        alerts  urls
-------------------------------------------------------------------------------
auditlog-db         STOPPED    0/1       16.0 MB  <unlimited>         <none>
auditlog-server     STARTED    1/1       256 MB   <unlimited>         https:
auditlog-broker     STARTED    1/1       64.0 MB  <unlimited>         https:
deploy-service      STARTED    1/1       280 MB   <unlimited>         https:
component-registry-db STOPPED  0/1       16.0 MB  <unlimited>         <none>
product-installer   STARTED    1/1       256 MB   <unlimited>         https:
xsa-admin-backend   STARTED    1/1       128 MB   <unlimited>         https:
xsa-admin           STARTED    1/1       128 MB   <unlimited>         https:
demo-app            STARTED    2/3       256 MB   <unlimited> 1/1/1   https:
demo-app2           STARTED    0/1       256 MB   <unlimited> DOWN!   https:
```

The information displayed in the example above includes the following details:

XS Advanced Application Details

Property	Description
name	The names of the XS advanced applications deployed in the current organization/space
requested state	The **targeted** state of the application, for example, STARTED or STOPPED. This state results from a request by an administrator to start or stop an application. → Tip The `requested state` does not indicate the **actual** state of any application instances. The actual status of an application is displayed in the `instances` property.
instances	The actual status of an application, described by two numbers separated by a slash, for example. 0/1: • The number of application instances that are currently running • The targeted number of running instances → Tip An application is fully started if the number of running instances matches the targeted number of running instances, for example, `1/1`.
memory	The amount of memory assigned to the application
disk	The disk quota allocated for use by the application

Property	Description
alerts	Information about problems with the application. If there were crashes of application instances, three values are displayed, for example, “`1/1/1`”, which represent the number of crashes within different periods of time (short term: 5 minutes, mid-term: 1 hour, long-term: 1 day). If XS advanced cannot start an application, '`DOWN!`' is displayed.
urls	The Universal Resource Location (URL), defined in a “route” where the application is accessible

When reading the information displayed by the `xs apps` command, bear in mind the following details:

- For most applications, the expected number of application instances are running.
- Some applications are `STOPPED` either automatically or at the request of an XS advanced administrator. Any application that performs a single task stops automatically as soon as it completes the assigned task. For example, the applications "`*-db`" in the example above (`auditlog-db` and `component-registry-db`) perform a one-off task such as initial database deployment. After the deployment tasks finishes, the database application is stopped to save system resources.
- The application `demo-app2` is down and no further application instances will be started by XS advanced.
- Only 2 of the possible 3 instances of the application "`demo-app`" are started. There was a crash of an instance within the last 5 minutes. This could indicate a problem that needs further investigation.

Displaying Details of an XS Advanced Application Instance

To display detailed information about a specific instance of an XS advanced application, use the `xs app <appName>` command, as shown in the following example:

Output Code

```
$ xs app demo-app

Showing status and information about "demo-app"
  name:                       demo-app
  requested state:            STARTED
  instances:                  1
  memory:                     128 MB
  disk:                       <unlimited>
  buildpack:                  sap_nodejs_buildpack
  urls:                       https://acme.org:63055
  created:                    Oct 17, 2022 2:46:40 PM
  updated:                    Nov 28, 2022 11:14:04 AM
  weighted instance uptime:   99.6 %
  ALERTS:                     There were crashes!
    short term (last 5 m)     1
    mid term   (last 1 h)     1
    long term  (last 1 d)     1
    total                     2
Instances of droplet 1 created at Oct 17, 2022 2:46:45 PM
index  created                  state    host      port   os user
-------------------------------------------------------------------
0      Oct 19, 2023 1:34:08 PM  CRASHED  acme.org  50012  xsaxsa

Instances of droplet 2 created at Nov 28, 2022 11:14:10 AM
index  created                   state     host   port    os user
-------------------------------------------------------------------
```

```
0      Nov 28, 2022 10:13:44 AM  STOPPED   acme01 50030   xsaxsa
1      Nov 28, 2022 11:14:48 AM  RUNNING   acme01 50030   xsaxsa
2      Nov 28, 2022 11:14:48 AM  RUNNING   acme01 50031   xsaxsa
3      Nov 28, 2022 11:14:48 AM  CRASHED   acme01 50033   xsaxsa
4      Nov 28, 2022 11:15:53 AM  STARTING  acme01 50032   xsaxsa, you can use
the
```

If alerts for an application are present, the `xs app` command shows detailed information about those alerts. For example, it shows the number of application crashes within short-, mid-, and long-term periods as well as the total number of crashes registered for an application.

For each application instance, the `xs app` command displays the following information:

- `index`
  , you can use theThe index of the application instance used to reference it with further commands
- `created`
  The time at which the application instance is created
- `state`
  The current state of the application instance:
  - `STARTING`
    The application instance has started, for example, at the request of an administrator, but is not yet accessible.
  - `RUNNING`
    The application instance started successfully and is accessible.
  - `STOPPED`
    The application instance stopped, for example, at the request of an administrator, and is not accessible.
  - `CRASHED`
    The application instance terminated unexpectedly.

    > → Tip
    >
    > You can use the command `xs delete-app-instances` to clean up instances of an application in the state `CRASHED`.
- `host`
  The host on which the application instance is started
- `internal port`
  The internal port of the application instance

  > → Tip
  >
  > The internal port is not the port on which the application is reachable by a business user; it is the number of the **internal** system port to which the XS advanced Platform Router forwards requests. The external end point for an application instance is exposed by the Platform Router.
- `os user`
  The operating system user in whose account the processes of the specified application instances are started.

The output returned by the `xs app` command also includes information about the selected application's “technical” version and “droplet” version. Each time you stage an application as part of the deployment push to the XS advanced run-time platform, a so-called droplet is created which can be referenced by an index.

Although it is often the case that only application instances of the most recent droplet are running, it is also important to understand that information about a certain number of previous droplets and application instances is kept for supportability reasons. In the example output above, two droplets of the same application are running, and in the command output, the droplets are indicated by "`droplet 1`" and "`droplet 2`", where `droplet 2` is the most recent version of the application.

- `droplet 2`
  The most recent version of the application, as indicated by the creation time stamp, and the higher the droplet index the more recent the version. You can also see that two application instances are currently running and one application instance is still starting, which explains why the expected number of application instances has not yet been reached. One application instance crashed recently which caused the alert. One application instance of droplet 2 is `STOPPED`, which means that this instance of the application was started but the application processes are no longer running.

  > ⓘ Note
  >
  > The contents of the `STOPPED` application instance's file system sandbox are retained so that support teams can analyze log and applications files at a later point in time, if necessary.

- `droplet q`
  The previous version of the application. The application instance with index 0 of droplet 1 is in state `CRASHED'`, which means that the application was started before but terminated unexpectedly. Similarly to stopped applications, the file-system sandbox of crashed application instances is kept for the most recent instances so that support teams can analyze log and application files to find out the cause of the error.

> → Tip
>
> Only the five most recently stopped or crashed instances are kept; any older application instances are deleted automatically.

Displaying XS Advanced Application Logs

To query the logs written by an XS advanced application (for example, output written to `stdout` and `stderr` as well as any access logs), run the `xs logs` command, as illustrated in the following example:

```
$ xs logs myAPP
```

When called without parameters, the `xs logs` command switches to "tailing" mode, where only newly written application logs are displayed in the system output, and the command prompt is returned when the user presses the keyboard combination: CTRL + C. To display the most recent entries of an application's logs, use the option "`--recent`" as shown in the following example:

```
$ xs logs myAPP --recent
```

To display all logs written by an application, use the option "`--all`" as shown in the following example:

```
$ xs logs myAPP -all
```

By default, the `xs logs` command displays the logs of **all** application instances. If you are only interested in the logs of a particular instance of an application, you can reference the application instance by including the

command options `--droplet` and `--instance`. For example, if you want to display the logs of the crashed instance of the `demo-app` application in the example above, run the following command:

Output Code

```
$ xs logs demo-app --all --droplet 1 --instance 0

11/27/22 4:04:57.122 PM [API] OUT Created app 'demo-app' [Org 'myorg', ...
11/27/22 4:04:57.360 PM [API] OUT Updated files for app 'demo-app' [Org...
11/27/22 4:04:57.390 PM [API] OUT Created droplet with id 1 of app 'demo-...
11/27/22 4:04:57.398 PM [API] OUT Staging Droplet with id 1 of app demo-a...
11/27/22 4:04:59.875 PM [STG/1] OUT Node.js
11/27/22 4:05:00.555 PM [STG/1] OUT Node.js buildpack version 3.3.2 [...]
11/27/22 4:05:02.066 PM [STG/1] OUT Copying SSL CA certificates...
...
```

→ Tip

Each log entry includes a time stamp, followed by the log source in brackets, followed by the log type and the actual log message. For more information about log sources and log types as well as how to filter for a specific type or source, see the following sections.

XS Advanced Application Log Sources

The following tables lists the sources used by XS advanced applications when writing log files:

XS Advanced Application Logfile sources

Log Source	Description
`[API]`	A log entry created by the XS advanced platform regarding this application
`[STG/<Droplet index>]`	A log entry created by the staging process regarding this application. The entry displays the droplet index as a suffix. For example, "`STG/1`" means that this staging process produced the droplet with index 1
`[APP/<Droplet index>-<Instance index>]`	The output of an application instance, either on `stdout` (prefixed with '`OUT`') or on `stderr` (prefixed with '`ERR`'). In the log type tag, the droplet and application instance is encoded in this case. In the example output above, "`APP/1-0`" means that the application instance with index 0 of droplet 1 dumped a log line to `stdout`.
`[RTR]`	The access log for this application created by the Platform Router (`RTR`)

It is possible to filter for certain log sources by specifying a log source (or multiple sources in a comma-separated list) in the option "`--source`" with the `xs logs` command, as shown in the following example:

Output Code

```
$ xs logs demo-app --all --source RTR

11/28/22 6:28:58.000 PM [RTR] OUT ##.##.209.139 - to acme.corp:63018 "GET...
11/28/22 6:28:58.000 PM [RTR] OUT ##.##.209.139 - to acme.corp:63018 "GET...
```

XS Advanced Application Log Types

A log source may contain log lines of different **types**. The following table lists the available log types:

XS Advanced Application Log File Types

Log Type	Description
OUT	A log entry was written to Standard Out (`stdout`)
ERR	A log entry was written to Standard Error (`stderr`)
ACC	The log line was written to the application instance's access log file
LOG	The log line was written in list log format to `stdout` and the logger starts with "`/Application`"
SYS	The log line was written in list-log format to `stdout`
TRC	Application-specific tracing

It is possible to filter the output to display only content from particular log **types**, for example, by specifying a log type (or multiple types in a comma-separated list) in the option "`--source`" with the `xs logs` command, as shown in the following example:

```
$ xs logs demo-app --all --type ERR

11/27/22 6:29:58.123 PM [APP/1-0] ERR    Segmentation fault
```

You can combine options and log files as shown in the following example:

```
xs logs demo-app --recent --source APP --type ERR
11/27/22 6:29:58.123 PM [APP/1-0] ERR    Segmentation fault
```

Displaying XS Advanced Application Events

The XS Controller saves important events that occur during the life time of an application. Viewing the events provide an overview of the changes made to an application from the point of view of the XS advanced platform:

Output Code

```
$ xs events deploy-service

Showing events for app "deploy-service" and user XSA_ADMIN

time         level   component     message
-------------------------------------------------------------------------------
Dec 1, 2022
11:26:00 AM  INFO    APPLICATION   Created app 'deploy-service' [Org ...
11:26:00 AM  INFO    SERVICE       Created service binding between app...
[…]
11:26:04 AM  INFO    APPLICATION   Updated files for app 'deploy-service'...
11:26:04 AM  INFO    DROPLET       Created droplet with id 1 of app 'deploy-..
11:26:04 AM  INFO    DROPLET       Staging Droplet with id 1 of app deploy-...
11:26:21 AM  INFO    DROPLET       Staged app 'deploy-service' [Org...
```

```
[…]
```

The application-specific information displayed in the output is grouped by day (for example, `Dec 1. 2017`) and typically covers the high-level areas listed and described in the following table:

XS Advanced Application Event Output

Event Information	Description
`time`	The exact time when the event occurred
`level`	The severity level assigned to the event, for example, `INFO`, `ERROR`, etc.
`component`	The name of the component that caused the event, for example: `APPLICATION`, `SERVICE`, `DROPLET`, etc.
`message`	The message text describing the event

Displaying XS Advanced Application Files

If an application is uploaded to the XS advanced run-time platform and staged, the application files are stored in a so called droplet. If an application instances are subsequently started, the files contained in the droplet are extracted into the file system for the instance to be executed. That means, application files exist on different layers, and the `xs files` command enables you to view the files by choosing the layer you are particularly interested in.

Application Instance Files

To view the files of the currently running application instance, use the `xs files` command with the application name and no further parameters.

Output Code

```
$ xs files deploy-service

Getting files of app "deploy-service" ...

dxwr - META-INF/
dxwr - WEB-INF/
  wr 154 B index.html
dxwr - logs/
```

For each file or directory, `xs files` displays information about the following components:

- The file type and system permissions (d=directory; x=executable; w=writeable; r=readable)
- The size of the file
- The name of the file or directory

To view detailed information about an specific directory tree, add the name of the directory to the command, as shown in the following example.

Output Code

```
$ xs files deploy-service META-INF/

Getting files of app "deploy-service" ...

dxwr - META-INF/
dxwr - WEB-INF/
  wr 154 B index.html
dxwr - logs/
```

To view detailed information about an individual file, specify the full path for the file, as shown in the following example.

Output Code

```
$ xs files deploy-service index.html

Getting files of app "deploy-service" ...

== "index.html" ====================================
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>XS2 ALM Service</title>
</head>
<body>Welcome to XS2 ALM Service!
</body>
</html>
```

Application Droplet Files

To view detailed information about the contents of application droplets, use `xs files` command with the `--droplet-index` option, as shown in the following example:

```
$ xs <myAPP> --droplet-files [--droplet-index <index>]
```

The command shows the contents of the root directory of an application's droplet. By default, the latest droplet is displayed. However, you can choose to display details of a specific droplet by specifying its index, for example, with the option `--droplet-index`. For more information about how to display an application's droplet index, use the `xs app <myApp>` command as described in *Displaying Application Details* above.

To download all files associated with an application instance (or droplet) to a target directory on your local disk, use the option `--download <target directory>`, as illustrated in the following example:

```
$ xs <myAPP> --download <target directory>
```

Related Information

Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]

14.2.2.1.8 Maintaining Applications in XS Advanced

Deploy applications to the XS advanced run-time environment

The most common task performed when maintaining applications in XS advanced is the deployment of XS advanced applications to the run-time environment. Although XS advanced provides tools to maintain entire business applications consisting of multiple micro-services, the XS command-line interface includes commands that enable administrators to maintain micro-services individually. In addition, the XS CLI also provides commands that help support the maintenance of XS advanced applications during their life cycle:

- Maintaining Multi-Target Applications in XS Advanced [page 1297]
- Maintaining Individual Microservices in XS Advanced [page 1298]
- Starting and restarting applications [page 1300]
- Restaging applications [page 1301]
- Configuring Application Health Checks [page 1302]

Maintaining SAP Products

The commands `xs install` and `xs uninstall` provided by the Product Installer can be used to maintain SAP products. For more information about using these commands to perform application life-cycle management tasks in XS advanced, see *Related Information*.

Maintaining Multi-Target Applications in XS Advanced

Multi-Target Applications (MTA) are applications composed of one or more microservices. An MTA is delivered in an archive with the file extension `.mtar` (multi-target application archive). The MTAR contains a meta-data description (MTAD, multi-target-application description) as well as the application code itself in a single deployment package. The deployment is performed by the so called Deploy Service, which deploys the individual micro-services specified by an MTAR, creates the corresponding service instances and takes care of connecting the deployed micro-services. To deploy an application in MTAR format, use the command `xs deploy`, as illustrated in the following example:

```
$ xs deploy mybusinessapp.mtar
```

To display a list of all currently deployed MTAs, use the command `xs mtas`, as illustrated in the following example:

Output Code

Listing the Deployed MTAs

```
$ xs mtas

Getting multi-target apps in org "mymorg" / space "SAP" as XSA_ADMIN...
Found multi-target apps:

mta id                   version
--------------------------------
```

```
com.sap.xsa.admin        1.5.5
com.sap.xs.auditlog.ui   1.0.0
com.sap.xs.jobscheduler  1.6.2
com.sap.core.account     1.1.2
alm-product-installer    1.13.7
```

To display a overview of the resources and microservices associated with a specific MTA, use the `xs mta` command, as illustrated in the following example:

Output Code

Listing the Deployed MTAs

```
$ xs mta com.sap.xs.jobscheduler

Getting information for multi-target app "com.sap.xs.jobscheduler"
in org "orgname" / space "SAP" as XSA_ADMIN...
Showing information about "com.sap.xs.jobscheduler" version: 1.6.2

Apps:
name                    state    inst  mem    disk         urls
------------------------------------------------------------------------------
jobscheduler-backend    STARTED  1/1   256MB  <unlimited>  https://host1:51022
jobscheduler-broker     STARTED  1/1   256MB  <unlimited>  https://host1.51021
jobscheduler-dashboard  STARTED  1/1   256MB  <unlimited>  https://host1:51023
jobscheduler-db         STOPPED  0/1   256MB  <unlimited>  <none>
jobscheduler-rest       STARTED  1/1   256MB  <unlimited>  https://host1:51017
jobscheduler-service    STARTED  1/1   512MB  <unlimited>  https://host1:51020

Services:
name                        service   plan
---------------------------------------------
jobscheduler-sbss           hana      sbss
jobscheduler-db-container   hana      hdi-shared
jobscheduler-uaa            xsuaa     default
jobscheduler-securestore    hana      securestore
```

To undeploy all micro-services belonging to a deployed MTA, use the command `xs undeploy`, as illustrated in the following example:

```
$ xs undeploy com.sap.xs.jobscheduler
```

Maintaining Individual Microservices in XS Advanced

To deploy a single XS advanced application, log on the XS advanced run-time instance and use the command `xs push` to deploy the application to the organization and space where you want the application to run, as illustrated in the following example:

```
xs push MY-APP
```

During the deployment, the XS advanced runtime performs the following tasks:

1. Uploads and stores the new application files
2. Creates or updates application meta data
3. Creates and binds routes

4. Binds any required service instances
5. Chooses the appropriate application run-time environment
6. Creates a droplet of the application
7. Selects the available execution agents to run the droplet
8. Starts the application instances

Output Code

Deploying a Single Application in XS Advanced

```
$ xs push

MY-APP Creating app "MY-APP" in org "orgname" / space "SAP" as XSA_ADMIN...
Creating HTTP route "MY-APP.acme.org" in org "myorg"/space "SAP" as
XSA_ADMIN..

Binding route "https://MY-APP.acme.org" to app "MY-APP"...
Uploading "MY-APP" ...
 Checking which files to upload from /tmp/xs2TestApp ...
 -> "MY-APP" consists of 10 files....
 Uploading 10 new or modified files ...
 Uploading "MY-APP" finished in 265 ms.

Staging app "MY-APP"...
   OUT Detected Java application
   OUT Compiling Java application...
   OUT Java XS Buildpack Version: 1.#.##
   OUT Downloaded 'SAP JVM JRE', version '8.#.##' in 0.559 s.
   OUT Downloaded 'Tomcat Runtime', version '8.#.##' in 0.285 s.
   OUT Downloaded 'XS Authenticator', version '1.#.#' in 0.0 s.
   OUT Downloaded 'SAPJWT', version '1.#.#' in 0.0 s.
   OUT Downloaded 'SAP JVM Memory Calculator', version '1.#.#' in 0.0 s.
Starting app "MY-APP"...
 Starting instances as OS user "sapxsaxsa"
 0 of 1 instances running, 1 starting ...
Showing status and information about "MY-APP"
 1 of 1 instances running
  name: MY-APP
  requested state: STARTED
  instances: 1
  memory: 1.00 GB
  disk: <unlimited>
  buildpack: <default>
  urls: https://MY-APP.acme.org

Instances of droplet 1 created at Feb 13, 2023 10:17:36 AM
index  created                    state      os user
-----------------------------------------------------
0      Feb 13, 2023 10:17:49 AM   RUNNING    sapxsaxsa
```

Note

XS advanced applications that use services (for example, database or file-system) are not completely functional the services are provisioned and bound to the application. For more information about maintaining services in XS advanced, see *Related Information*.

Before you push the application to XS advanced, bear in mind the following additional options that might be useful or required when performing a custom deployment:

- Custom host name, port or domain:

You can specify a custom route during deployment. The route is a combination of the domain and subdomain and must be globally unique. This is true whether you specify a portion of the route or allow XS advanced to use defaults.

- Custom start command:
  You can specify a custom command to start instances of the application; this custom command replace the automatically detected start command.
- Custom memory limit:
  You can specify the maximum amount of memory that each instance of the application can use.

> ### Note
>
> For more information about setting memory limits, see *Scaling Applications in XS Advanced* in *Related Information* below.

Starting and Restarting XS Advanced Applications

After pushing the application "MY-APP", the XS advanced run-time environment starts the application with the **default** start command of the appropriate build pack. If you want to use a custom start command, use the "-c" option with the `xs push` command, as shown in the following example.

```
$ xs push MY-APP -c "node MY-APP.js"
```

The custom command that you provide with the -c option becomes the default start command and is used for subsequent updates (`xs push MY-APP`) of the application. To reset to the start command used by the default build-pack, use the special start parameter value "`null`", as illustrated in the following example:

```
$ xs push MY-APP -c "null"
```

The commands `xs start` and `xs stop` are used to start or stop an application. This always affects all application instances, but it does not change the number of instances configured for scaling the application. When an application is started, the most recent droplet and the current settings for the application environment variables are used. To restart your application, run the following command:

```
$ xs restart MY-APP
```

Restarting an application stops and restarts it with the most recent droplet and environment variable settings. This is required for "refreshing" the application after any operation that modifies the application's environment, for example, binding a new service or directly setting environment variable values.

> ### Note
>
> If an environment variable is also consumed by the build pack, then restarting an application might not be enough to effect the desired change. In this case, the application must first be restaged for the change to take effect.

Restaging XS Advanced Applications

Restaging an application compiles a new droplet of the staged application. Staging an application can be useful in the following cases:

- You changed the application environment
- The application must consume an updated application runtime
- The application must use an updated set of trust certificates

> ### Note
>
> A restaged application continues to run during staging; the application only restarts, if you call the `xs restart` command.

To restage your application, use the `xs restage` command, as shown in the following example:

```
$ xs restage MY-APP
```

> ### Note
>
> Restaging an application compiles a new droplet from the application without updating the application source. To update the application source, push (deploy) the application again.

Restaging Applications on Outdated Runtimes

The `xs restage` command provides an option that enables the automatic restaging and restarting of applications that are currently using outdated application runtimes. In the context of restaging, an application runtime is considered to be outdated if there is a newer version of the same runtime type installed in XS Advanced. For example, if two application runtimes of type "node20" are installed (version 20.10.0 and version 20.11.0), then the older of the two versions is considered outdated. Any applications that have the outdated runtime in their active droplet can automatically be restaged.

To find all such applications and automatically restage and restart them, use the `xs restage` command with the --outdated option, as shown in the following example:

```
$ xs restage --outdated
```

This will display a list of all updatable applications that are accessible for the logged in user. After confirmation, the command begins restaging and restarting the listed applications. Depending on the number of applications, this can take some time to finish. It is possible to restrict the restaging operation to specific spaces, by providing additional arguments to the command. The following example uses the -s option to restrict the restaging operation to the applications using outdated rutn times in "space1" and "space2":

```
$ xs restage --outdated -s space1 -s space2
```

Even after successful restaging and restarting, an application could still be listed as being bound to an outdated runtime. As long as the XS advanced runtime still keeps the execution root directories of stopped application instances with the old droplet, a reference to the old runtime version will remain. To clean up any old execution root directories, use the option --clear-instances, as shown in the following example:

```
$ xs restage --outdated --clear-instances
```

The --clear-instances option ensures that old execution root directories are deleted. Since this removes any references to the old application droplet, the droplet can safely be removed using garbage-collection so that the application is no longer listed as bound to an outdated application runtime version.

Configuring XS Advanced Application Health Checks

XS advanced determines the state of application instances by means of so-called application health checks. An application health check is a monitoring process that periodically checks the status of an application instance. If one check fails, the application instance is marked as `CRASHED` and automatically restarted. For more information about the possible status of an XS advanced application instance, for example, `STOPPED`, `STARTING`, or `RUNNING`, etc., see *Displaying Application Information in XS Advanced* in *Related Information* below.

> → Tip
>
> If tracing is enabled, the health-check status reports are written to the log file of the XS advanced execution agent.

XS advanced supports the following types of application health check:

- Process:
  The health check examines that the application instance process is running.
- Port:
  The health check opens a TCP connection on the application instance port and checks that the port is connectable. This is the default health check type.
- HTTP:
  The health check makes a HTTP GET request to a specified application end point and checks that the response code is 200.

It is possible to set the type of application health check either during application deployment (for example, with the "`xs push`" command or in the application's `manifest.yml` file) or if the application is already running with the "`xs set-health-check`" command.

Health Check Configuration Options

You can use the `xs push` command to set the health check for a specific application, using the options listed in the following table:

Application Health-Check Configuration Options

Health-Check Option	Description
`-u <HEALTH_CHECK_TYPE>`	The application health check type. Valid values are port, process or http. The default is port.
`--health-check-timeout <TIMEOUT>`	The XS advanced runtime will wait at most the specified time for a successful health check after the initial startup of an instance. If no health check was successful after the timeout was reached, the instance will be marked as crashed and the instance will be terminated. The default is to wait indefinitely.

Health-Check Option	Description
`--health-check-http-endpoint <HEALTH_CHECK_ENDPOINT>`	The end point for the HTTP request provided by an application. This option is only relevant for the health-check type http. The default value is "/".
`--invocation-timeout <HTTP_INVOCATION_TIMEOUT>`	The timeout for each HTTP GET request. This option is only relevant for the health-check type http. The default value is 1 second.

In the following example, the command pushes an application and configures the health check type 'http' with a health check timeout of 180 seconds using the HTTP endpoint "/healthcheck" and a GET request timeout of 5 seconds:

```
$ xs push demo-app -u http --health-check-timeout 180 --health-check-http-
endpoint /healthcheck --invocation-timeout 5
```

The following example shows how to specify the same configuration in the application manifest:

```
---
  ...
  health-check-type: http
  health-check-http-endpoint: /healthcheck
  invocation-timeout: 5
```

If the application has already been pushed, you can use the `xs set-health-check` command to configure health checks, as illustrated in the following example:

```
xs set-health-check <application> http --endpoint /healthcheck --invocation-
timeout 5
```

> **Note**
>
> You must restart the application to enable the configured health checks.

Viewing the Health Check Configuration for an Application

You can use the `xs get-health-check` display the current health-check configuration, as illustrated in the following example:

```
$ xs get-health-check demo-app

  name :                              demo-app
  health check type :                 http
  endpoint :                          /
  invocation timeout (in seconds) :   1

OK
```

Related Information

Displaying Application Information in XS Advanced [page 1288]
Installing and Updating Products and Software Components in SAP HANA XS Advanced Model [page 626]

Maintaining Services in XS Advanced [page 1305]
Maintaining the XS Advanced Application Environment [page 1316]
Maintaining Trust Certificates in XS Advanced [page 1321]
Scaling Applications in XS Advanced [page 1304]

14.2.2.1.9 Scaling Applications in XS Advanced

Configure applications to handle increased traffic on demand.

Scaling an application can enable it to handle increased traffic on demand, for example, to meet the needs associated with increased user load. The command `xs scale` can be used to increase or decrease the number of running application instances (also known as horizontal scaling) or to change the resource limit of existing application instances (vertical scaling).

Horizontal Scaling

You can horizontally scale your application by specifying the number of application instances that should be allowed, as illustrated in the following example, which sets the maximum number of instances to five (5):

```
$ xs scale MY-APP -i 5
```

This configures the XS advanced run-time environment to increase or decrease the number of instances of the specified application to match the number of instances specified.

> ⓘ Note
>
> The consumption of file-system and memory resources increases with the number of application instances. In addition, the application itself determines whether it supports horizontal scaling, that is, whether it can manage several application instances running in parallel.

A round-robin policy is used to distribute newly created application instances among available hosts bearing the role xs_worker. You can use host pinning to refine the set of `xs_worker` hosts where instances of specific applications will be scheduled to run. For more information about host pinning, see *Related Information*. To decrease the number of application instances, the same round-robin policy is used to remove application instances from affected `xs_worker` hosts. The load generated by requests sent to a scaled application is automatically balanced across all application instances by the XS advanced Platform Router. Although, by default, a weighted round-robin policy is used to balance the load associated with requests among application instances, it is nontheless possible to choose from the other load-balancing algorithms provided by the SAP WebDispatcher.

Vertical Scaling

To change the memory limits for existing applications, use the command "xs scale" with the option "-m", for example:

```
$ xs scale MY-APP -m 1G
```

In this example, the memory limit of the application `MY-APP` is changed to 1 GB. To define the unit of size, you can either use "`M`" (for megabytes) or "`G`" for gigabytes. For the new settings to be effective, restage and restart the corresponding application, for example, with the commands `xs restage` and `xs restart`, respectively. For more information about restaging and restarting application in XS advanced, see *Related Information*.

> ⚠ Restriction
>
> By default, memory limits currently are only valid for Java applications. However, you can use the `OPTIMIZE_MEMORY` option to configure a memory limit for Node.js applications, too.

To limit the memory for Node.js applications globally, use the following command:

```
xs update-staging-environment-variable-group --add OPTIMIZE_MEMORY true
```

Alternatively, you can use the following command to configure the memory setting for individual Node.js applications:

```
xs set-env <app> OPTIMIZE_MEMORY true
```

In both cases, the memory limit for an application becomes active, when the application is restaged.

Related Information

Maintaining Host Pinning [page 1338]
Maintaining Applications in XS Advanced [page 1297]

14.2.2.1.10 Maintaining Services in XS Advanced

Use the `xs` command-line interface to maintain so-called “Backing Services” for applications running in the XS advanced run-time platform.

Backing Services are the medium by which applications running in the XS advanced run-time platform can access resources, for example: a database, an offering for audit logging, or user authentication.

Backing Services are connected with applications by creating so called service instances within XS advanced. During the creation of a service instance, XS advanced creates the backing-service resource by means of a so-called “service broker”. By adding custom service brokers, arbitrary backing services can be connected to XS advanced. At some point, the service instance is bound to an application. In this way, all the information required for access to the backing service is injected into the process environment of an application instance.

- Listing Backing Services in the Market Place [page 1306]
- Maintaining Service Instances [page 1306]
- Creating User Provided Services [page 1308]
- Binding Service Instances to Applications [page 1309]
- Maintaining Service Keys [page 1310]
- Maintaining Service Brokers [page 1310]
- Maintaining Syslog Drain Services [page 1312]

Listing Backing Services in the Market Place

Typically the service broker provides credentials to the application to access the service. In XS advanced, the list of available backing services provided by service brokers is displayed in the service "market place", as shown in the following example:

Output Code

```
$ xs marketplace

Getting services from marketplace...

service       plans                                    description
------------------------------------------------------------------------------
fs-storage    free                                     xs file service provides.
xsuaa         default, devuser, space                  Manage app authorizati...
hana          hdi-shared, sbss, schema, securestore    SAP HANA database
managed-hana  hdi-shared, schema, securestore          Creates service
instan...
auditlog      free                                     Audit log broker on
XSA..
```

The output from the `xs marketplace` command displays the following information:

- `service`
  The name of the backing service
- `plan`
  The available service plans a particular backing service provides. A service plan determines the type or category of service that an individual backing service provides. For example the `hana` service can be used to get a single container (`schema`) or provide access to the SAP HANA Secure Store (`securestore`).
- `description`
  A short summary of the backing service

Maintaining Service Instances

To make use of a service, an instance of the service must be created and an XS advanced application must be bound to the specified service instance. When a service instance is created, resources are allocated within the

respective backing service. To create a service instance, use the command `xs create-service` and choose a service name and a service plan from the marketplace, as illustrated in the following example:

```
$ xs create-service hana hdi-shared myservice
Creating service "myservice"...
  create in progress
  create succeeded
OK
```

You can configure service instances by passing a set of parameters with the option `-c` in JSON format. You can also add tags to the service instance with the option `-t`; the information is passed to the bound applications. Service instances are created within a particular space and cannot be used from other spaces.

The `xs update-service` command enables you to update a service instance, for example, by modifying its service plan as well as any parameters or tags:

```
$ xs update-service myservice -t mytag
Updating service instance "myservice"...
OK
```

> ⚠ Caution
>
> The corresponding service broker determines what happens during a service update. For example, changing the plan of an existing service could result in a new allocation of service resources within the backing service, and this could lead to the deletion of all previous data represented by this service instance.

The `xs rename-service` command enables you to rename a service instance, as illustrated in the following example:

```
$ xs rename-service myservice newservice
OK
```

> ⓘ Note
>
> Renaming a service instance has no effect on the backing resource represented by the service instance.

The `xs delete-service` command enables you to remove a service instance along with all associated data, as illustrated in the following example:

```
$ xs delete-service myservice
Really delete service instance "myservice"? (y/n) > y
Deleting service instance "myservice"...
OK
```

Since a service instance is maintained and persisted not only at the service broker but also at the XS Controller, the service-related information should be consistent in both places. If the service information is lost at the service broker, for example, because it has been manually deleted, you can use the purge option (`xs delete-service --purge`) to forcibly remove the service information from the XS Controller as well.

To display a list of all currently available service instances in a space, use the command `xs services` as illustrated in the following example:

> {·} Output Code
>
> Viewing Available Service Instances
>
> ```
> $ xs services
> ```

```
Getting services from marketplace...

name                      service     plan          last operation     bound
apps
-------------------------------------------------------------------------------
------------------
auditlog-db-container     hana        hdi-shared    create succeeded
auditlog-db, auditlog-...
auditlog-sbss             hana        sbss          create succeeded
auditlog-server, auditlog
deploy-service-auditlog   auditlog    free          create succeeded   deploy-
service
deploy-service-fss        fs-storage  free          create succeeded   deploy-
service
deploy-service-ss         hana        securestore   create succeeded   deploy-
service
deploy-service-database   hana        schema        create succeeded   deploy-
service
deploy-service-uaa        xsuaa       default       create succeeded   deploy-
service
product-installer-dbase   hana        schema        create succeeded
product-installer
component-registry-dbase  hana        hdi-shared    create succeeded
component-registry-db,...
```

The output from the `xs services` command displays the following information:

- `name`
  The name of the service instance
- `service`
  The name of the backing service
- `plan`
  The available service plans a particular backing service provides. A service plan determines the type or category of service that an individual backing service provides. For example the `hana` service can be used to get a single container (`schema`) or provide access to the SAP HANA Secure Store (`securestore`).
- `last operation`
  Shows the last performed operation (create, update, delete) and its state (in progress, succeeded, failed).
- `bound apps`
  Shows all applications bound to the service.

Creating User Provided Services

Since user-provided services do not require a service broker they are not chosen from the service catalog. Instead, the credentials for a user-provided services are provided when creating the service instance, as illustrated in the following example:

{·} Output Code

User Credentials in JSON for a User-Provided Service

```
$ xs create-user-provided-service my-up-service -p
'{"host":"example.org","username":"admin","password":"pa55woRD"}'

Created environment (excerpt):
{
```

```
  "name" : "my-up-service",
  "credentials" : {
    "password" : "pa55woRD",
    "host" : "example.org",
    "username" : "admin"
  }
}
```

→ Tip

The combination of operating system, shell, or terminal type determines the “quote” or “escape” characters required when providing the parameter string in JSON format. For more information about the correct quoting, see `xs help create-user-provided-service`.

The `xs` CLI also has an interactive mode, which you can use to specify the fields of the service credentials as parameters, for example, with the `-p` option and a comma-separated list.

{·} Output Code

Interactive User Credentials for a User-Provided Service

```
$ xs create-user-provided-service my-up-service -p host,username,password
host> example.org
username> admin
password> pa55woRD

Created environment (excerpt):
{
  "name" : "my-up-service",
  "credentials" : {
    "password" : "pa55woRD",
    "host" : "example.org",
    "username" : "admin"
  }
}
```

Binding Service Instances to Applications

To connect an application with a particular backing service resource, you must bind the corresponding service instance to the application using the `xs bind-service` command, as illustrated in the following example:

```
$ xs bind-service myapp myservice
OK
TIP: Use 'xs restart' to ensure your env variable changes take effect
```

It is also possible to pass binding parameters in JSON format by using the option "`-c`". The effect of binding parameters depends very much on the corresponding backing service. After restarting the bound application, credentials required for to access to the backing service can be found in the environment variable `<VCAP_SERVICES>` in the application environment. To remove a service-binding, call the command `xs unbind-service`, as illustrated in the following example:

```
$ xs unbind-service myapp myservice
OK
```

Maintaining Service Keys

Service keys provide a way to query the credentials required for a service without having to bind the service instance to an application. To create a service key use the `xs create-service-key` command, as shown in the following example:

```
$ xs create-service-key myservice myservicekey

Creating service key "myservicekey" for service instance "myservice" ...
OK
```

Service keys are created within a particular space and cannot be used from other spaces. After creating a service key, you can view the service credentials by calling the command `xs service-key`, as illustrated in the following example:

Output Code

```
$ xs service-key my-service my-service-key

Getting service key "my-service-key" for service instance "my-service" ...
{
  "host" : "host1.acme.org",
  "user" : "1EBB56E88DYS76",
  "password" : "Dd2f8RVCP8gr1[...]v"
}
OK
```

To display a list of all existing service keys, use the command `xs service-keys`:

Output Code

```
$ xs service-keys

Getting service keys in org "orgname" / space "SAP" as XSA_ADMIN...

service instance   name
------------------------------
myservice          myservicekey
```

To remove a service key, use the command `xs delete-service-key`:

Output Code

```
$ xs delete-service-key myservice myservicekey

Really delete service key "myservicekey" for service instance "myservice"?
(y/n) > y
Deleting service key "myservicekey" for service instance "myservice" ...
OK
```

Maintaining Service Brokers

The service broker interface provides a way to connect arbitrary backing service with the XS advanced platform. XS advanced implements the open service broker API (version 12.2). To view the service brokers

currently connected to the XS advanced platform, use the `xs service-brokers` command, as shown in the following example:

Output Code

```
$ xs service-brokers

Getting service brokers...
Found service brokers:

name              url
------------------------------------------------------------------
fs-storage        https://acme.org:30033/v2/fs-service
uaa-security      https://acme.org:30033/uaa-security
hdi-broker        https://acme.org:30033/hdi-broker
instance-manager  https://acme.org:30033/instance-manager
auditlog          https://acme.org:30033
```

You can add your own service broker with the command `xs create-service-broker`, as illustrated in the following example:

Output Code

```
$ xs create-service-broker mybroker user passwd https://acme.org:8080/activemq

OK
```

After a service broker has been created, the backing services provided by this service broker immediately show up in the service catalog, also known as the marketplace.

To rename an existing service broker, use the following command:

Output Code

```
$ xs rename-service-broker mybroker newbroker

OK
```

If a service broker is moved to a different URL or the credentials for a service broker change, you can update an existing service broker to reflect the new URL and new credentials by calling the command `xs update-service-broker`, as illustrated in the following example:

Output Code

```
$ xs update-service-broker mybroker user passwd https://acme.org:1080/
activemq

OK
```

To remove an existing service broker from the XS advanced platform, use the following command:

Output Code

```
$ xs delete-service-broker mybroker

Really delete service broker "mybroker"? (y/n) > y
Deleting service broker "mybroker"...
```

```
OK
```

Maintaining Syslog Drain Services

In order to stream application logs to an external log management service, for example an Elastic (ELK) stack, you can make use of so called “syslog-drain services”. These services define a URL to the `syslog`-compatible end point of the log-management service: the so-called syslog drain. The same application logs that can be streamed to a `syslog` drain can be accessed with the command `xs logs`, as described in *Displaying Application Information* in *Related Information*. XS advanced supports syslog drains which are able to parse log messages according to the standards described in RFC 5424.

To start streaming logs from your application to the syslog drain, you need to create a user-provided service and bind it to your application. The following example shows how to use the `xs` command-line interface to create the corresponding user-provided service; you need to specify a service name and, by means of the parameter -l, the URL of the syslog drain:

```
$ xs create-user-provided-service my-drain -l syslog://syslog-
drain.example.org:1234
OK
```

The log messages can be delivered via TCP, TCP over TLS, or HTTPs, as shown in the following table, which provides an overview of the format of the syslog-drain URL that XS advanced expects for the supported protocols:

URL Formats for Syslog Drain in XS Advanced

Protocol	Syslog-drain URL Format
TCP	`syslog://syslog-drain.example.org:1234`
TCP over TLS	`syslog-tls://syslog-drain.example.org:1234`
HTTPs	`https://syslog-drain.example.org:1234`

Next, you need to bind the created service to your XS advanced application, as illustrated in the following example:

```
$ xs bind-service my-app my-drain
OK
TIP: Use 'xs restart' to ensure your env variable changes take effect
```

After a short delay all new logs generated by the bound application will be streamed to the syslog drain. A restart of the application is not required, as the syslog-drain service binding does not affect the environment variables of the application.

If you are streaming logs to a syslog drain that uses TLS, make sure that the required trust certificates are available to the XS Controller. For more information about how to configure trust certificates please see *Maintaining Trust Certificates* in *Related Information* below.

Bear in mind that additional configuration might be required within the log-management service of your choice. For more information, see *The Syslog Protocol* in *Related Information*.

Related Information

XS Advanced Platform Components [page 1257]
Displaying Application Information in XS Advanced [page 1288]
The Syslog Protocol (ietf.org)
Maintaining Trust Certificates in XS Advanced [page 1321]

14.2.2.1.11 Maintaining Application Routes in XS Advanced

Use the `xs` command-line interface to maintain routes for XS advanced applications.

Routes provide applications and their instances with a public HTTP endpoint. A route represents a URL that can be mapped to one or more applications. HTTP traffic that is sent to a route is forwarded by the Platform Router to one of the instances of the mapped applications. The selection of the application instance is decided by a round-robin algorithm. The Platform Router acts as the central routing and load-balancing component within XS advanced.

Usually a route is mapped to a single application. When scaling the application to multiple instances, traffic sent to the route is load-balanced between all instances. Mapping a route to multiple applications might for example make sense in cases where a blue-green deployment is performed.

Routing Modes

During installation of SAP HANA XS advanced model you can select between two routing modes: **ports** and **host names**. If the administrator configures the routing mode "ports", the XS advanced platform creates routes based on distinct **ports**, as illustrated in the following URLs representing different port-based routes:

- `https://acme.org:50000`
- `https://acme.org:50001`

If the administrator configures the routing mode "host names", the XS advanced platform creates routes based on distinct sub-domains, as illustrated in the following URLs representing different host-name-based routes:

- `https://app1.acme.org`
- `https://app2.acme.org`

> → Tip
>
> For more information about technical prerequisites and security implications of the two routing modes see SAP Note 2245631, *Domains and Routing Configuration for SAP HANA Extended Application Services, Advanced Model*.

In the "hostnames" routing mode, the XS advanced platform can also create routes based on ports, which is especially useful if you also want to use TCP routes, which always require a dedicated port.

To simplify the creation of port-based routes, you can configure a default port range for the platform, for example, by setting XS advanced platform parameters. In this way, a free port from the declared port range can

be used if no port is explicitly defined when creating a route. The port range should be treated as **reserved**, and SAP HANA XS advanced expects that no other processes on the SAP HANA hosts use ports within the declared port range. If you are running multiple SAP HANA XS advanced systems on the same host, it is important to separate the default port range for each XS advanced system from each other.

→ Tip

For more information about setting a default port range for XS advanced, see SAP Note 2507070, *Multiple XS Advanced Systems on the Same Host*.

Viewing Routes

To see which routes have been mapped to XS advanced applications in the current space, use the `xs routes` command, as illustrated in the following example:

{·} Output Code

Route List in "hostnames" Routing Mode

```
$ xs routes

Getting routes in org "XSA" / space "SAP" as XSA_ADMIN...

host                        domain    port   path  type   apps
-------------------------------------------------------------------------------
-
auditlog-server             acme.org         /     HTTP   auditlog-server
auditlog-broker             acme.org         /     HTTP   auditlog-broker
deploy-service              acme.org         /     HTTP   deploy-service
xsa-sap-product-installer   acme.org         /     HTTP   product-
installer
                            acme.org  50510  /     TCP    hello-world-tcp
                            acme.org  50500  /     TCPS   hello-world-tcps
```

The information displayed in the example above includes the following details:

XS Advanced Application Details

Property	Description
host	The host of the specified route. The host can only be set where hostname-based routing is used.
domain	The domain name used by the specified route. → Tip For more information about domains, see *Maintaining Domains* in *Related Information*.

Property	Description
port	The port that is dedicated to the specified route. The port can only be set if the route is a port-based route. → Tip Host-name-based routes use the default router port.
path	The URL path of the specified route. Using different URL paths you can create multiple routes with the same host or port.
type	The protocol used for the specified route, which can be one of the following: HTTP, TCP, or TCPS.
apps	The applications bound to the specified route

Creating and Mapping Routes

You can use the `xs map-route` and `xs unmap-route` commands to map routes to applications and remove the mappings between applications and routes. Similarly, the commands `xs create-route` and `xs delete-route` can be used to create new routes and delete them.

When creating a route, you can define a custom URL path instead of using the default URL path, ("`/`"). When forwarding the request to the application instance, the request is sent via the same URL path as the one that is defined in the route. As a result, applications must serve requests on the defined URL path. Using different URL paths enables you to create multiple routes that use the same sub-domain or port, as illustrated in the following examples:

- `https://app.acme.org/frontend`
- `https://app.acme.org/backend`

The default type of a route is HTTP. This means that the traffic on this route must adhere to the HTTPs (HTTP over SSL) protocol, which is used within XS advanced by default. It is also possible to change this configuration globally, in order to use plain HTTP routes. If a route is created with type HTTP, the XS advanced Platform Router can enable certain HTTP features for this route, for example: Sticky Sessions, HTTP access logs, or certain timeouts. The XS advanced platform is also able to provide additional information to the application instances through HTTP headers (for example, `X-Forwarded-Proto` and `X-Forwarded-Port`). Routes, which are based on hosts, can only be used with the HTTP type, as the Platform Router routes these requests to the correct application instances based on the `HTTP Host` header.

It is also possible to create routes of type “TCP” or “TCPs”. It is important to remember that the XS advanced Platform Router does not expect a specific protocol on TCP routes; it simply forwards the TCP traffic to the application instances using the mapped route. TCP routes must, however, use their own distinct port and are not allowed to use hosts. All HTTP features are disabled for TCP routes, so it only makes sense to use these routes if the application requires a protocol other than HTTP. In addition, since the XS advanced Platform Router does not terminate SSL, for TCP routes you can also terminate SSL traffic at the application instance.

You can also use the route type “TCPs”, which expects at least SSL-encrypted TCP traffic and allows the XS advanced Platform Router to terminate the SSL connection. The application instance receives plain TCP traffic.

The advantage of using “TCPs” is that the XS advanced application does not require any SSL configuration, and you can continue to rely on the domain certificates you have configured within SAP HANA XS advanced.

Related Information

Maintaining Domains and Certificates in XS Advanced [page 1324]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]

14.2.2.1.12 Maintaining the XS Advanced Application Environment

Use environment variables to configure an XS advanced application's behavior.

When an application instance is started in XS advanced, several configuration parameters are injected into the application's process environment; the parameters specify information about the following elements:

- Application properties
- Information about backing services
- Custom environment values

> → Tip
>
> You can use environment variables to transport custom configuration parameters to application instances.

Viewing the Application Environment

The command `xs env` displays an overview of an application's environment. The command output is split into two sections: `System-Provided` and `User-Provided`, as illustrated in the following example:

```
$ xs env demo-app

Getting env variables for app "demo-app"...
OK

System-Provided:
{
  "VCAP_APPLICATION" : {
    "start" : "2023-01-02 16:51:41 +0100",
    "application_id" : "927c0822-3734-4c91-9796-432ddd1a9ed4",
    "instance_id" : "927c0822-3734-4c91-9796-432ddd1a9ed4",
    "space_id" : "eccc5bd5-934b-4472-b32c-009d6497e93b",
    "application_name" : "demo-app",
    "organization_name" : "myorg",
    "space_name" : "PROD",
    "started_at_timestamp" : "1514908301002",
    "started_at" : "2023-01-02 16:51:41 +0100",
    "state_timestamp" : "1513239094881",
```

```
    "full_application_uris" : [ "https://demo-app.example.org" ],
    "application_uris" : [ "demo-app.example.org" ],
    "uris" : [ "demo-app.example.org" ],
    "version" : "3d397398-0f1a-4d07-b9e6-88f33d8387b0",
    "application_version" : "3d397398-0f1a-4d07-b9e6-88f33d8387b0"
  },
  "VCAP_SERVICES" : {
   "hana" : [ {
      "name" : "demo-app-database",
      "label" : "hana",
      "tags" : [ "hana", "database", "relational" ],
      "plan" : "schema",
      "credentials" : {
        "schema" : "USR_EHLDI8B567MXSPX1BLGSJJW7J",
        "password" : "1h0FHlR9o7WvC5AHmI_522DU8kLPIYKM",
        "driver" : "com.sap.db.jdbc.Driver",
        "port" : "31313",
        "host" : "962.acme.com",
        "db_hosts" : [ {
          "port" : 31313,
          "host" : "962.acme.com"
        } ],
        "user" : "USR_EHLDI8B567MXSPX1BLGSJJW7J",
        "url" : "jdbc:sap://962.acme.com:31313/?
currentschema=USR_EHLDI8B567MXSPX1BLGSJJW7J"
      }
  }
User-Provided:
  custom-value: 1234
```

System-Provided Environment Variables

System-provided environment variables are calculated by XS advanced and cannot be changed the user. The following environment variables contain application and service-binding properties:

- `VCAP_APPLICATION`
  Contains application properties in JSON format, for example, the time stamp of the last application start time, unique application identifiers, organization and space coordinates, and application URLs.
- `VCAP_SERVICES`
  Contains information about the bound backing services in JSON format. For each backing service, the information includes the credentials required for access to the specified backing service.

For more details about *Maintaining Services*, see *Related Information* below.

User-Provided Environment Variables

Custom user-provided environment variables can be set by the user or are set by the deployer of the application.

Modifying the Application Environment

To add a new environment variable or alter the value of an existing environment variable, use the command `xs set-env`, as shown in the following example:

> **Note**
>
> How an application consumes the value you set with the `xs set-env` command depends very much on the application itself.

Output Code

Setting Environment Variables in XS Advanced

```
$ xs set-env demo-app "http_proxy" "proxy:8080"

Setting env variable "http_proxy" to "proxy:8080" for app "demo-app"...
OK
TIP: Use 'xs restage' followed by 'xs restart' to ensure your env variable
changes take effect
```

To remove a new environment variable from the existing application environment, use the command `xs unset-env`, as shown in the following example:

Output Code

Setting Environment Variables in XS Advanced

```
$ xs unset-env demo-app http_proxy

Removing variable "http_proxy" from environment of app "demo-app"...
OK
TIP: Use 'xs restage' followed by 'xs restart' to ensure your env variable
changes take effect
```

Any change to the application environment is visible in the output of the `xs env` command immediately. However, the effect of the change will only be applied to the application after the application has been restarted.

→ Tip

If the buildpack that compiles the application consumes the altered environment variables, you must restage the application, too.

Global Environment Variables

It is possible to set global environment values for all applications; this avoids having to set the environment for each application individually. A common use case for such a scenario would be when applying proxy settings.

Running Environment Variable Groups

To set a group of environment variables for all newly started applications, use the command `xs set-running-environment-variable-group`. You can pass a group of variables by specifying a JSON structure as a command-line option, as illustrated in the following example:

ⓘ Note

This command overwrites any existing running environment variable group you have configured.

Output Code

Setting Running Environment Variables in XS Advanced

```
$ xs set-running-environment-variable-group '{"no_proxy":"localhost"}'

Setting the contents of the running environment variable group as
XSA_ADMIN...

Variable Name     Assigned Value
--------------------------------
no_proxy          localhost
```

To edit single variables for the currently running environment, it is more convenient to use the command "`xs update-running-environment-variable-group`" (or "`xs urevg`" for short). This command enables you not only to add single new environment variables and change the value of existing variables, but also to remove variables from the running environment. In the following example, the `xs urevg` command sets the value of the variable `my_db_host` to "`dbhost.acme.org`" by adding the variable to the running environment (if it is not yet part of the running environment-variable group) or by updating the existing variable to the new value:

```
$ xs update-running-environment-variable-group --add my_db_host dbhost.acme.org

Updating the contents of the running environment variable group as XSA_ADMIN...
Variable "my_db_host" updated.
Updated environment variables:

Variable Name     Assigned Value
--------------------------------
no_proxy          localhost
my_db_host        dbhost.acme.org
```

To remove any variable from the running environment-variable group, use the option `--remove` as illustrated in the following example:

```
$ xs update-running-environment-variable-group --remove my_db_host

Updating the contents of the running environment variable group as XSA_ADMIN...
Variable "my_db_host" removed.
Updated environment variables:

Variable Name     Assigned Value
--------------------------------
no_proxy          localhost
```

To view the currently configured running environment-variable group, use the command `xs running-environment-variable-group`, as shown in the following example:

Note

Environment variables in the running environment-variable group have no effect on the staging processes.

Output Code

Displaying the Running Environment Variables Group in XS Advanced

```
$ xs running-environment-variable-group

Retrieving the contents of the running environment variable group as
XSA_ADMIN...
```

```
Variable Name     Assigned Value
-------------------------------
no_proxy          localhost
```

Staging Environment Variable Groups

To set a group of environment variables that is only exposed to build packs during the application staging process, use the command `xs set-staging-environment-variable-group`, as illustrated in the following example:

> **Note**
>
> This command overwrites any existing **staging** environment variable group you have configured.

> **Output Code**
>
> Setting Staging Environment Variables in XS Advanced

```
$ xs set-staging-environment-variable-group '{"no_proxy":"localhost"}'

Setting the contents of the staging environment variable group as
XSA_ADMIN...

Variable Name     Assigned Value
-------------------------------
no_proxy          localhost
```

To edit single variables for the currently staging environment, you can use the command "`xs update-staging-environment-variable-group`" (or "`xs usevg`" for short). The `xs usevg` command enables you not only to add single new environment variables and change the value of existing variables, but also to remove variables from the staging environment. In the following example, the `xs usevg` command sets the value of the variable `my_db_host` to "`dbhost.acme.org`" by adding the variable to the staging environment (if it is not yet part of the staging environment-variable group) or by updating the existing variable to the new value:

```
$ xs update-staging-environment-variable-group --add my_db_host dbhost.acme.org

Updating the contents of the staging environment variable group as XSA_ADMIN...
Variable "my_db_host" updated.
Updated environment variables:

Variable Name     Assigned Value
-------------------------------
no_proxy          localhost
my_db_host        dbhost.acme.org
```

To remove any variable from the staging environment-variable group, use the option `--remove` as illustrated in the following example:

```
$ xs update-staging-environment-variable-group --remove my_db_host

Updating the contents of the staging environment variable group as XSA_ADMIN...
Variable "my_db_host" removed.
Updated environment variables:

Variable Name     Assigned Value
-------------------------------
no_proxy          localhost
```

To view the currently configured staging environment variable group, use the command `xs staging-environment-variable-group`, as shown in the following example:

Output Code

Displaying the Staging Environment Variables Group in XS Advanced

```
$ xs staging-environment-variable-group

Retrieving the contents of the staging environment variable group as
XSA_ADMIN...

Variable Name     Assigned Value
----------------------------------
no_proxy          localhost
```

Related Information

Maintaining Services in XS Advanced [page 1305]

14.2.2.1.13 Maintaining Trust Certificates in XS Advanced

Use the `xs` command-line interface to maintaining trust certificates for the XS advanced run-time platform.

The XS advanced run-time environment and the applications running in XS advanced use SSL connections wherever possible. Applications connect to each other using HTTPs and can also be configured to use SSL when connecting to the SAP HANA database. Although the XS advanced platform components (for example, the XS Controller) use the SAP HANA system PKI to securely communicate with each other and with the SAP HANA database, nonetheless it might be necessary to configure additional SSL trust certificates, especially when connecting securely to end points that are located outside the XS advanced run-time environment. Since application run times (for example, Java or Node.js) usually already include their own certificate trust stores, configuring additional SSL trust certificates might only be necessary when using certificates, which are not signed by a well known certificate authority (CA). This usually is the case for certificates signed by a corporate CA or self-signed certificates.

The XS advanced run-time platform includes tools that enable administrators to maintain and manage the certificates used to set up trusted relationships; you can use the tools to upload custom trust certificates to the platform. The trust certificates are propagated to the applications by the build packs. For example, the propagation of the certificates is performed by default when using either the SAP Node.js or the SAP Java build pack. However, you can use **custom** trust certificates, for example, if you have configured a custom “certificate authority” within your corporation.

By adding the root certificate of your certificate authority, you enable any application to establish fully trusted SSL sessions with corporate servers, even those configured outside the SAP HANA system. In addition to custom trust certificates, the XS advanced run-time platform can be used to manage **domain** certificates. For more information, see *Maintaining Domains* in *Related Information* below.

→ Tip

Like domain certificates, trust certificates are available to all applications.

Viewing Trust Certificates

To display a list of custom trust certificates, use the `xs trusted-certificates`, as illustrated in the following example:

Output Code

```
$ xs trusted-certificates

Retrieving the list of trusted certificates as XSA_ADMIN...

Alias: CORPORATE_CA
------------------------------------------------------------------------
Subject:              CN=CORPORATE Root CA,O=SAP SE,L=Walldorf,C=DE
Issuer:               CN=CORPORATE Root CA,O=SAP SE,L=Walldorf,C=DE
Valid from:           Sun Jan 01 12:00:00 UTC 2023
Valid until:          Thu Jan 01 11:59:59 UTC 2032
Signature algorithm:  SHA256withRSA

Alias: HANA_SSL
------------------------------------------------------------------------
Used within HANA Broker service bindings
Subject:              CN=hanahost,O=SAP SE,L=Walldorf,C=DE
Issuer:               CN=hanahost,O=SAP SE,L=Walldorf,C=DE
Valid from:           Sun Jan 01 12:00:00 UTC 2023
Valid until:          Mon Jan 01 11:59:59 UTC 2024
Signature algorithm:  SHA256withRSA
Alias: SSO_CA
------------------------------------------------------------------------
Used for validation of certificate-based client authentication
Subject:              CN=CORPORATE SSO CA,O=SAP SE,L=Walldorf,C=DE
Issuer:               CN=CORPORATE SSO CA,O=SAP SE,L=Walldorf,C=DE
Valid from:           Sun Jan 01 12:00:00 UTC 2023
Valid until:          Thu Jan 01 11:59:59 UTC 2032
Signature algorithm:  SHA256withRSA
```

The `xs trusted-certificates` command displays a list of all trust certificates (including the alias for a certificate, if defined) and some additional information that is extracted from the certificate itself, for example, the subject, the issuer, and the period for which the certificate is valid.

Adding Trust Certificates

To upload a custom trust certificate, use the command `xs trust-certificate`, as illustrated in the following example:

Output Code

```
$ xs trust-certificate <ALIAS> -c <PATH>
```

In the example command output above, `<ALIAS>` is an arbitrary name and has no special meaning. `<PATH>`, however, must point to a valid certificate file, which must be a X.509 certificate in the `PEM` format.

Caution

All trust certificates are passed to all applications and can be used to validate SSL sessions to any server.

SAP HANA SSL Trust Certificates

If you want to use SSL to encrypt connections between the SAP HANA database and SAP HANA XS advanced, you must upload to the XS advanced certificate trust store the root certificate of the server certificate, which is used by your SAP HANA database. For more information about how to set up SSL encryption between SAP HANA XS advanced model and the SAP HANA database it is running on, see SAP Note 2300943 (*Enabling SSL encryption for database connections for SAP HANA extended application services, advanced model*).

From SAP HANA SPS 06 and from XS advanced 1.2.0, there is no need to manually configure individual, database, trust certificates for XS advanced applications; these trust certificates are configured automatically for XS advanced applications during startup of the XS advanced run time, when SSL is enforced at the database. If during startup of XS advanced, a manually configured database certificate is found, the configuration still applies; auto-configuration of an XS advanced application's database certificate only occurs in those cases where no manually configured database certificate exists for an XS advanced application.

> → Tip
>
> You can display a list of all the automatically trusted certificates using the `xs trusted-certificates --all` command.

Client Authentication Trust Certificates

The XS advanced platform router supports the validation of client certificates to allow certificate-based authentication. The platform router also has the option to request client certificates if at least one trust certificate for client authentication has been configured in XS advanced. The validated client certificate is passed on to platform services and applications in the HTTP header "`X-Forwarded-Client-Cert`".

The certificates used by the platform router to validate a client certificate during the SSL handshake must be uploaded to XS advanced using the `xs trust-certificate` command together with the `--client-auth` option. Certificates that are uploaded using the "`--client-auth`" option are not propagated to applications. For more information, see *Maintaining Single Sign-On for XS Advanced Applications* in *Related Information* below.

> ⓘ Note
>
> It is strongly recommended to use root CA certificates as client-authentication trust certificates; it is not recommended to upload user-specific client-authentication trust certificates.

Related Information

Maintaining Domains and Certificates in XS Advanced [page 1324]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]
Maintaining Single Sign-On for XS Advanced Applications [page 1509]

14.2.2.1.14 Maintaining Domains and Certificates in XS Advanced

Use the `xs` command-line interface to maintain domains and domain certificates for the XS advanced run-time platform.

In XS Advanced, domains are used as building blocks for application routes. Application routes are used to route business-user requests to the respective applications. You can configure multiple domains in your XS advanced system. The following example shows how the domains `acme.org` and `myotherdomain.org` are used in routes for applications:

- `https://app.acme.org`
- `https://app.myotherdomain.com`

During installation you need to specify a default domain. This domain is used when creating URLs for the platform components and when creating the routes of the applications created by the initial installation. The default domain is a shared domain, which means it can be used from within any organization. When creating additional domains one can limit the usage of a domain to a specific organization. Such domains are called "private domains" and can then only be used by routes created in that organization. You can also create external domains to indicate that a domain references an external host, for example a reverse proxy.

Maintaining Domains in XS Advanced

To display a list of all domains, run the following command:

Sample Code

```
xs domains

Getting domains...

name                  type               org    port
-------------------------------------------------------
acme.org              shared (default)
myotherdomain.com     shared
privatedomain.com     private            XSA
externaldomain.com    shared (external)         443
```

The information displayed in the example above includes the following details:

XS Advanced Application Details

Property	Description
shared (default)	The default domain specified in `xscontroller.ini` which is available to all organizations
shared	An additional domain that is available to all organizations
private	An additional domain that is only available to a single organization XSA.

Property	Description
external	A shared domain which points to a reverse proxy or is a virtual hostname

If XS advanced is configured to use the host-name-based routing mode, the XS advanced platform automatically creates subdomains for a domain when a new route is created. To ensure that this is possible, DNS entries for the domains need to include a wild-card DNS entry, which needs to point to the XS advanced Platform Router. For more information about the prerequisites for configuring and using domains in XS advanced, see SAP Note 2245631, *Domains and routing configuration for SAP HANA extended application services, advanced model*.

Creating a Shared Domain

You can use the command `xs create-domain` to create additional shared domains in XS advanced, as shown in the following example:

```
xs create-domain myotherdomain.com
```

You can then create individual routes using this domain. To access applications by the new new domain, the domain needs to be resolvable by the browser of the busines-user as well as by the application backend running on the SAP HANA system.

If XS advanced is configured to use the host-name-based routing mode, the XS advanced platform automatically creates subdomains for a domain when a new route is created. To ensure that this is possible, DNS entries for the domains need to include a wild-card DNS entry, which needs to point to the XS advanced platform router. For more information about the prerequisites for configuring and using domains in XS advanced, see SAP Note 2245631 (*Domains and routing configuration for SAP HANA extended application services, advanced model.*)

Creating an External Domain

You can use the xs create-domain command to create an external domain in XS Advanced, as shown in the following example:

```
xs create-domain externaldomain.com --external
```

For every route on the default domain, a virtual route on the external domain is created automatically (external routes). It is not necessary to create external routes, manually. The external domain may point to a different host outside XS Advanced. This means that with an external domain you can setup up XS Advanced behind a reverse proxy, with the following requirements:

- Business user requests on the external domain have to be routed to the default domain, e.g. by a reverse proxy or a virtual hostname.
- Business user requests are redirected to the respective authentication endpoint dynamically, depending on which domain the request was sent (default domain or external domain).

- Application backends use the default domain for internal communication.

With this approach, the external domain and the default domain can be in different DNS zones, assuming the following requirements:

- Business users must be able to resolve the external domain. However, the external domain does not need to be resolvable from within the SAP HANA system.
- Similarly, XS Advanced applications must be able to resolve the default domain. However, the default domain does not need to be resolvable outside the SAP HANA system, for example, by business users.

This method also enables users to log into and administrate XS Advanced via the default domain while exposing individual applications to business users via the external domain.

> ⚠ Restriction
>
> If you created an external domain, business users that log in via an external IdP can only access the platform via the external domain. They cannot access the platform via the default domain.

For more information about setting up XS Advanced behind a reverse proxy, see *Installing XS Advanced behind a reverse proxy* in *Related Information* below.

Using a Custom Port

In hostname-based routing mode, it is possible to use a custom port for external routes, which can be different from default domain routes. For example, you can use the standard port 30033 for default domain routes internally but expose applications on port 443 to business users. The following example shows how to use the `-p` (port) option to ensure that the reverse proxy forwards all traffic on port 443 to the default domain and to port 30033:

```
xs create-domain externaldomain.com --external -p 443
```

> ⓘ Note
>
> If you do not specify the `-p` (port) option, the `router_port` property in `xscontroller.ini` is used to configure the port for the external routes.

Using external domains with SAP HANA System Replication

Configuring an external domain for the failover router or virtual hostname in a SAP HANA System Replication setup also has effects on the system replication takeover:

- If no external domain is configured, the `default_domain` setting in the `xscontroller.ini` file is automatically replicated to the secondary system. For this reason, the default domain certificate that you have setup on the primary system is automatically valid on the secondary systema as well.
- When configuring an external domain, the `default_domain` setting in the `xscontroller.ini` file is not replicated to the secondary system. This means that the secondary system will start using a different default domain than the primary system, with the advantage that XS Advanced is able to start independently, even if the reverse proxy is not available. However, this requires that the default domain certificate you configure at the primary system contains the host names of all hosts of all primary and secondary systems, so that the certificate is valid even after a failover to a new host.

For more information about setting up XS advanced with system replication, see *SAP HANA System Replication Setup for XS Advanced Runtime* in *Related Information* below.

Removing Domains

To remove shared, private or external domains, run the following command in a terminal:

> ⓘ Note
>
> It is not possible to remove the default domain that is defined in `xscontroller.ini`.

```
xs delete-domain <domain-name>
```

Managing Domain Certificates

You can configure SSL certificates for every domain: the SSL certificate for a domain is used for every application route that uses this domain. By default, the XS advanced platform automatically generates a self-signed SSL certificate for every domain; the XS advanced administrator can replace this self-signed SSL certificate with a certificate signed by a well-known certificate authority. If the “host names” routing mode is enabled, it is necessary to deploy a wild-card certificate to each domain, since routes for a domain are created as subdomains. If you are using the “ports” routing mode, wild-card certificates are not needed.

> ⓘ Note
>
> Wild-card certificates include the domain name (for example, "`example.org`") and an additional wildcard entry such as "`*.example.org`" in the certificate, as illustrated in the following example.

To display a list of domain certificates for each configured domain, use the `xs domain-certificates`, as illustrated in the following example:

> {·} Output Code

```
$ xs domain-certificates

Retrieving the list of domain certificates as XSA_ADMIN...

Domain: acme.org
------------------------------------------------------------
Last updated at:            Sun Jan 01 12:00:00 UTC 2023
Created by:                 Platform (self-signed)
Subject:                    CN=acme.org,OU=XS,O=ACME SE,C=GH
Subject alternative names:  [[2, .acme.org], [1, *.acme.org]]
Issuer:                     CN=acme.org,OU=XS,O=ACME SE,C=GH
Valid from:                 Sun Jan 01 12:00:00 UTC 2023
Valid until:                Sun Jan 01 12:00:00 UTC 2024
Signature algorithm:        SHA256withRSA
```

In addition to domain certificates, the XS advanced run-time platform can be used to manage custom trust certificates. For more information, see *Maintaining Trust Certificates* in *Related Information* below.

> → Tip
>
> Domain certificates are treated in the same way as custom trust certificates; both types of certificates are available to **all** XS advanced applications.

To configure a customer certificate for a domain, use the `xs set-certificate` command. You can use certificates either im PEM or in PSE format. The following sections describe how to set a customer certificate in the supported formats. If you are using certificates in other formats, you need to create new ones using the supported formats or convert the certificates to the supported formats using the CA guidelines.

For more information about how to set certificates for the XS advanced run-time platform, see SAP Note 2243019.

Certificates for External Domains

If the external domain points to a reverse proxy that does not terminate SSL or is a virtual hostname, you can set up a custom certificate for this domain which is presented to clients.

If the reverse proxy terminates SSL, the reverse proxy is responsible for issuing the certificate for the external domain. In this case, however, the reverse proxy opens a new SSL session to forward a request to the default domain. For this reason, the reverse proxy must trust the XS Advanced default domain root certificate.

Setting Domain Certificates in PEM Format

To configure your own certificate for a domain you can use the "xs set-certificate" command. The command requires a private key in PKCS8 PEM format and the full certificate chain in X.509 PEM format. The following example walks you through the process of creating a CA-signed certificate and setting it as the domain certificate. It uses the command-line tool `openssl`.

Creating a Certificate-Signing Request with openssl

> → Tip
>
> For more information about creating CA-signed certificates, see SAP Note 2243019 (*Providing SSL certificates for domains defined in SAP HANA extended application services, advanced model*).

As part of the process of creating a domain certificate, you must first create a certificate-signing request, which you would normally define in the `openssl` configuration files, but can also do on the command line, in a bash script, as illustrated in the following very simple example, in which the contents of the `subjectAltName` and `Subject` fields must be adapted to fit to the corresponding domain name:

```
openssl req -new -sha256 -newkey rsa:4096 -nodes \
  -keyout domain.key \
  -subj "/CN=acme.org/C=GH/ST=ACC/O=ACME/OU=XS" \
  -reqexts SAN \
  -config <(cat /etc/ssl/openssl.cnf \
    <(printf "\n[SAN]\nsubjectAltName=DNS:acme.org,DNS:*.acme.org")) \
  -out domain.csr
```

> ⚠ Caution
>
> Make sure that the file generated by the process is only readable by the user running the script.

After you have successfully created a certificate signing request in the file `domain.csr` and a new private key in the file `domain.key`, you need to provide the certificate signing request to the certificate authority

of your choice. You must follow the CA-specific process to have the certificate signed and then download the signed certificate as `domain.crt`. Note that you also need all the intermediate certificates and the root certificate which were used by the certificate authority to sign your domain certificate. In the following example, we assume the certificate authority used an intermediate CA certificate "`intermediate.crt`" for signing purposes, and the intermediate certificate was signed by the root CA certificate ("`root.crt`").

Since the private key must be in PKCS8 format, use the `openssl` command to perform the conversion, as illustrated in the following example:

```
openssl pkcs8 -topk8 -nocrypt -in domain.key -out domain.pk8
```

Configuring the Certificate in XS Advanced

You can now configure the certificate in XS advanced by running the `xs set-certificate` command, as illustrated in the following example:

```
xs set-certificate example.org -c root.crt intermediate.crt domain.crt -k
domain.pk8
```

> → Tip
>
> The platform performs a connectivity check using the new certificate and returns an error message if it detects that there is something wrong with the certificate. You can remove the files "`domain.pk8`", "`domain.key`", and "`domain.csr`" after completing this step.

If you received a `.p12` or `.pfx` file from the Certificate Authority rather than separate certificate and key files, you can use it directly with the `-pkcs12` option in the `xs set-certificate` command, as follows:

```
xs set-certificate example.org -pkcs12 example.pfx
```

To ensure the settings take full effect, restart the XS advanced system by running the XSA command as `<sid>adm`, as illustrated in the following example:

```
XSA restart
```

Setting Domain Certificates in PSE Format

It is also possible to use the `set-certificates` command to configure CA-signed domain certificates in the PSE-format, as shown in the following example. The first step is to create a new PSE container together with

a certificate signing request. To create the PSE container, use the `sapgenpse` command, as illustrated in the following example:

Creating a Certificate-Signing Request with sapgenpse

> ⓘ Note
>
> The subject and subject alternative names parameters need to be adapted to suit your use case. You should also remember to protect the PSE container with a suitable PIN and ensure that the generated files are only readable by trusted users who really need access to the sensitive data stored in the file.

```
sapgenpse gen_pse -p domain.pse -r domain.csr -k "GN-dNSName:example.org" -k "GN-dNSName:*.example.org" "CN=example.org, C=DE, ST=BW, O=SAP, OU=XS"
```

If the command completes successfully, the certificate signing request will be available in the file "`domain.csr`", and a new PSE container with the name "`domain.pse`" is created. Next, you need to provide the certificate signing request to the certificate authority of your choice. You must follow the CA-specific process to have the certificate signed. Request the certificate from the CA as a PKCS7 container; this container includes the signed certificate, all intermediate certificates, and the root certificate.

The next step is to import the signed certificate and its certificate chain into the PSE container. Put the PKCS7 response you receive from your CA in a file called "`domain.p7`" and import it into the PSE container using the following command:

```
sapgenpse import_own_cert -p domain.pse -c domain.p7
```

Configuring the Certificate in XS Advanced

You can now configure the certificate using the following command:

```
xs set-certificate example.org --pse domain.pse
```

> → Tip
>
> The platform performs a connectivity check using the new certificate and returns an error message if it detects that there is something wrong with the certificate. You can remove the files "`domain.pse`", "`domain.p7`", and "`domain.csr`" after completing this step.

To ensure the settings take full effect, restart the XS advanced system by running the `XSA` command as `<sid>adm`, as illustrated in the following example:

```
XSA restart
```

Recovering from Expired or Corrupted Certificates

If the XS Advanced default certificate is invalid or has expired, it is not possible to set a new certificate with remote tools such as the `xs` command-line interface or the XS Advanced Admin Cockpit. To recover from a situation where the default certificate for XS Advanced has expired or is no longer valid, log in at the affected SAP HANA system as `<sid>adm` user and run the following command:

```
XSA reset-certificate
```

The `reset-certificate` command generates a new, self-signed, default certificate for XS Advanced and then restarts XS Advanced. Once the new certificate is available, you can use the commands described in the sections above to set a new custom certificate with the `xs` cli or in the XS Advanced Admin Cockpit.

Related Information

Maintaining Trust Certificates in XS Advanced [page 1321]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]
The XSA Command Reference [page 1370]
Setting Up XS Advanced Behind a Reverse Proxy (SAP HANA Installation Guide)
SAP HANA System Replication Setup for XS Advanced Runtime [page 787]

14.2.2.1.15 Maintaining Tenant Databases in XS Advanced

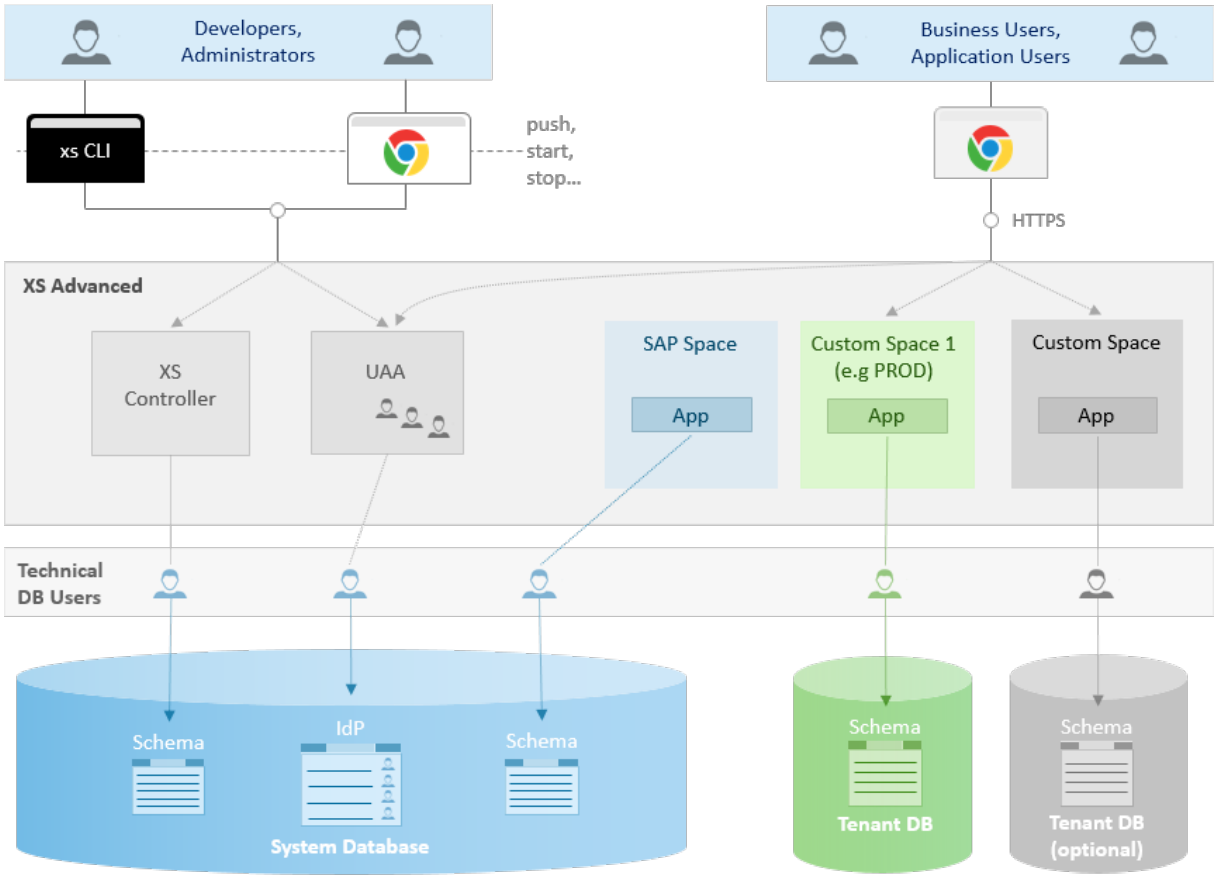
Use the `xs` command-line interface to maintain tenant database for the XS advanced run-time platform.

This section describes how XS advanced integrates into a multiple-database containers (MDC) setup in SAP HANA.

Although XS advanced is a system-wide service, it uses a dedicated database for the persistence of XS advanced system data, for example, meta-data about applications, spaces, and organizations. An XS advanced space, however, can be mapped to an arbitrary tenant databases in an SAP HANA system. In this way, custom applications within an XS advanced space can use the mapped tenant database to persist application data. Although there is already a data isolation concept based on SAP HDI containers within a single database, using different tenant databases for different XS advanced spaces enables even stronger isolation of application data.

XS Advanced in the System Database

The following figure shows the setup when XS advanced is installed in the system database, that is, uses the system database for system data:



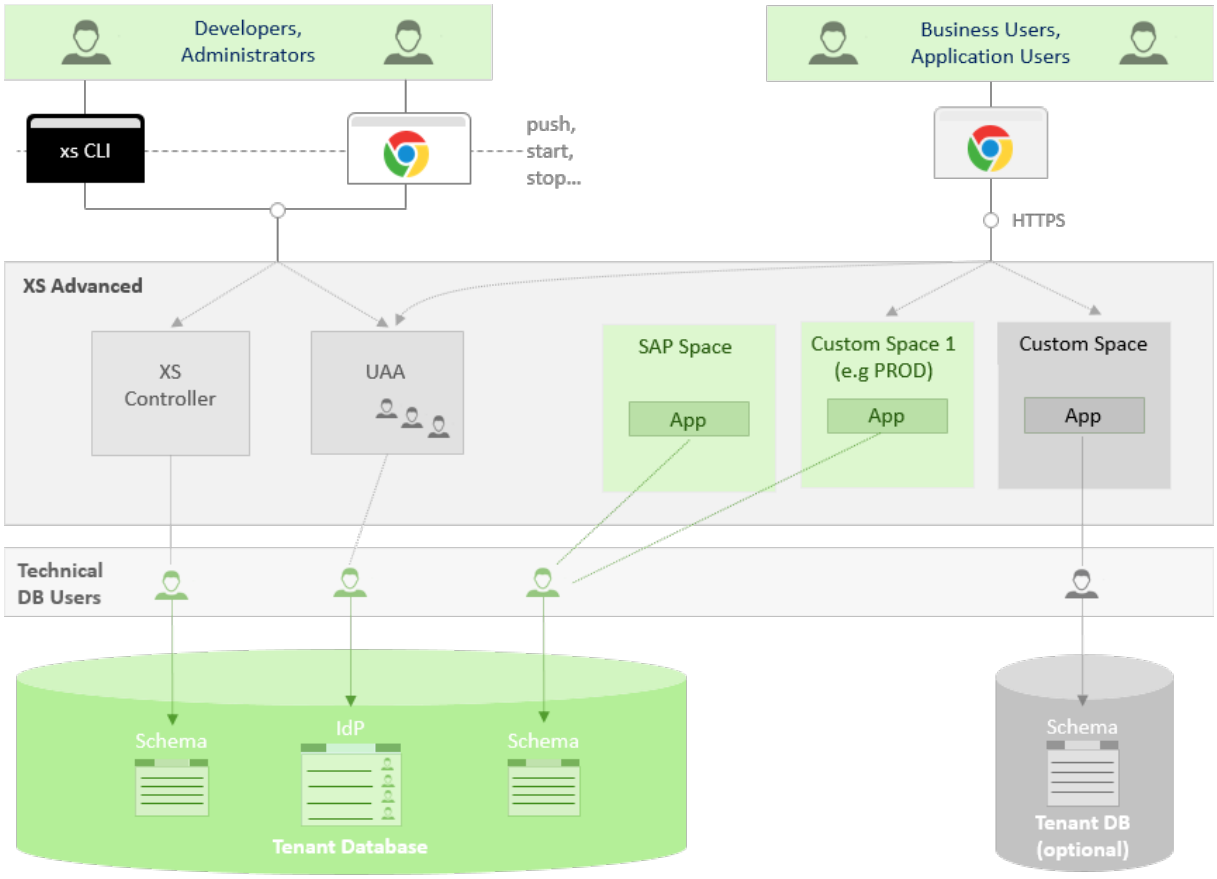
XS advanced in the System Database (Fresh Installation)

- XS Advanced platform services and the User Account and Authentication (UAA) service use the System database as persistence
- UAA uses the System database as an Identity Provider (IdP). This means that database users created in the System database can be used as XS advanced platform users or business users.
- The SAP space is mapped to the System database by default. This means that XS advanced core applications use the System database as persistence.
- A custom space (for example 'PROD') is mapped to the default tenant database. This means that applications bound to SAP HANA service instances in the custom space use the default tenant database as persistence.
- Newly created spaces are mapped by default to the default tenant database.
- It is possible to map additional tenant databases if required.

Although there is already a data-isolation concept based on HDI Containers within a single database, using different tenant databases for different spaces enables even stronger isolation of application-specific data.

XS Advanced in a Tenant Database

The following figure shows the setup when XS advanced is installed in a tenant database, that is, uses the tenant database for system data:



XS advanced in a Tenant Database (Fresh Installation)

- XS Advanced platform services and the User Account and Authentication (UAA) service use a tenant database as persistence
- UAA uses the a tenant database as an Identity Provider (IdP). This means that database users created in the tenant database can be used as XS advanced platform users or business users.
- The SAP space is mapped to the same tenant database where XS advanced in installed. This means that XS advanced system applications use this tenant database as persistence.
- A custom space (for example 'PROD') is mapped to the same tenant database where XS advanced in installed. This means that applications bound to SAP HANA service instances in the custom space use this tenant database as persistence.
- Newly created spaces are mapped by default to the same tenant database.
- It is possible to map additional tenant databases if required.

Viewing Tenant Databases in XS Advanced

As an XS advanced administrator, you can use the `xs tenant-databases` command to display a list of the current mappings between spaces and tenant databases, as illustrated in the following example:

Output Code

XS advanced persistence installed in tenant "XSA":

```
$ xs tenant-databases

Getting tenant databases as XSA_ADMIN...

name      XSA      status  host         sql-port  mappings
-----------------------------------------------------------------
MYTENANT1 -        active  h1.acme.org  30041
MYTENANT2 enabled  active  h2.acme.org  30044
XSA       main     active  h3.acme.org  30015     <default>, orgname:PROD
```

Output Code

XS advanced persistence installed in `SYSTEMDB`:

```
$ xs tenant-databases

Getting tenant databases as XSA_ADMIN...

name      XSA      status  host         sql-port  mappings
-----------------------------------------------------------------
MYTENANT1 -        active  h1.acme.org  30041
MYTENANT2 enabled  active  h2.acme.org  30044
XSA       enabled  active  h3.acme.org  30015     <default>, orgname:PROD
SYSTEMDB  main     active  h3.acme.org  30013     orgname:SAP
```

The information displayed in the example above includes the following details:

XS Advanced Application Details

Property	Description
name	The name of the database tenant
XSA	The listed tenant database is either enabled ("enabled") or not yet enabled ("-") for use with XS advanced. The database with XSA persistence is labelled as "main".
status	An `active` tenant database is up, running, and available; a tenant database that is `inactive` is either stopped or in the process of being created, or otherwise unavailable.
host	The name of the master host where the tenant database is running
sql-port	The port to use for SQL connections to this tenant database's master host
mappings	A comma-separated list of the XS advanced spaces (`org:space`) that are mapped to the tenant database

In the example describes the following mapping configuration:

- The tenant databases `MYTENANT1` and `MYTENANT2` are active, but not yet enabled for use with XS advanced.
- The tenant database `XSA` is enabled for use in XS advanced and the tenant is mapped by default when a new XS advanced space is created. In addition, the tenant database `XSA` is mapped explicitly to the space `PROD` in the organization `orgname`.
- The `systemDB` is used by applications in space `SAP` within the organization `orgname`.

Enabling Tenant Databases for XS Advanced

To register a new tenant database, for example, `MYTENANT1`, for use in XS advanced use the following command:

```
$ xs enable-tenant-database MYTENANT1
```

If you use the `xs` CLI to create a new tenant database, the new database is immediately **enabled** for use with XS advanced, as shown in the following example:

Output Code

```
$ xs tenant-databases

Getting tenant databases as XSA_ADMIN...

name      XSA  status  host         sql-port  mappings
-------------------------------------------------------------------
MYTENANT1 yes  active  h1.acme.org  30041
MYTENANT2 no   active  h2.acme.org  30044
XSA       yes  active  h3.acme.org  30015     <default>, orgname:PROD
SYSTEMDB  yes  active  h4.acme.org  30013     orgname:SAP
```

→ Tip

The `XSA` column indicates if a tenant database is either enabled (`yes`) or disabled (`no`) for use in XS advanced. A **disabled** database also does not have either a host or an SQL port assigned.

Alternatively, you can use the xs command `xs create-tenant-database`, which creates a tenant DB and registers it for use with XS advanced automatically:

```
$ xs create-tenant-database MYTENANT3
```

Mapping Tenant Databases in XS Advanced

After a tenant database is enabled for use with XS advanced, you can map organizations and spaces to the newly enabled tenant database with the `xs map-tenant-database` command, as shown in the following example:

```
$ xs map-tenant-database -o orgname -s myspace MYTENANT1
```

```
$ xs map-tenant-database -o myorg2 MYTENANT2
```

→ Tip

If you map a tenant database to an organization, the mapping is inherited by **all** spaces in the mapped organization, as long as no tenant is set explicitly.

To display an overview of the mappings between tenant databases and organizations and spaces, use the `xs tenant-database-mappings` command, as shown in the following example:

{·} Output Code

```
$ xs tenant-database-mappings

Getting tenant database mappings as XSA_ADMIN...

organizations and spaces    tenant databases
------------------------------------------------------------
myorg2                      MYTENANT2
myorg2:dev                  MYTENANT2 (inherited from orgname)
orgname                     XSA (inherited from global)
orgname:PROD                XSA
orgname:SAP                 SYSTEMDB
orgname:myspace             MYTENANT1
```

If the same space or organization is mapped to multiple tenant databases, you can use the `--default` option to specify one of the tenant databases as the default database, as shown in the following command:

```
$ xs map-tenant-database -o orgname -s myspace MYTENANT2 --default
```

The selected **default** tenant database is flagged in the output of the `xs tenant-database-mappings` command as shown in the following example:

{·} Output Code

```
$ xs tenant-database-mappings

Getting tenant database mappings as XSA_ADMIN...

organizations and spaces    tenant databases
------------------------------------------------------------
[...]
orgname:myspace             MYTENANT1, MYTENANT2 (default)
[...]
```

If no default tenant database is declared or you want to use a particular mapped tenant database for a certain SAP HANA service instance, you must specify the GUID of the target database whenever you create an SAP HANA service in the target space, as illustrated in the following example:

```
$ xs create-service hana hdi-shared myservice -c '{"database_id":
"0e140209-5e3a-42b3-8c7a-21260eba4287"}'
```

To display a list of know tenant databases and the corresponding database GUIDs use the `--guids` option, as shown in the following example:

{·} Output Code

```
$ xs tenant-databases --guids
```

```
Getting tenant databases as XSA_ADMIN...

name       XSA  status  host         sql-port  id             mappings
-------------------------------------------------------------------------
MYTENANT1  yes  active  h1.acme.com  30041     0e140209-...   orgname:myspace
```

Unmapping and Deleting Tenant Databases in XS Advanced

To unmap a tenant database from a space or organization, use the command `xs unmap-tenant-database`, as shown in the following example:

```
$ xs unmap-tenant-database -o orgname -s myspace MYTENANT1
```

To disable a tenant DB for use with XS advanced, use the command `xs disable-tenant-database`, as shown in the following example:

```
$ xs disable-tenant-database MYTENANT1
```

To delete a tenant database, use the command `xs delete-tenant-database`. You need to specify the credentials for the system users of both the System database (`SYSTEMDB`) and the specified tenant database (`MYTENANT2`), as shown in the following example:

```
$ xs delete-tenant-database MYTENANT2 -n <TENANTDB_SYSTEM_USER> -t <PASSWD> -u
<SYSTEMDB_SYSTEM_USER> -p <PASSWD>
```

> ⚠ Caution
>
> Deleting a tenant database destroys all data within the specified tenant database. In addition, if you use the SQL command-line interface (CLI) to delete a tenant database that is mapped to an organization or space in XS advanced, the database mapping in XS advanced is not removed, and XS advanced is not made aware of the deletion of the tenant DB, which can lead to problems for XS advanced.

Renaming Tenant Databases in XS Advanced

You can rename not only the tenant database where XS advanced is installed but also a tenant database that is registered for use with XS advanced, as follows:

1. Stop the XS advanced platform.

   ```
   $ XSA disable
   ```

2. Rename the tenant database.
   For more information about how to rename a tenant database, see *Related Information* below.
3. Restart the XS advanced platform:
   After renaming a tenant database, the command you use to restart XS advanced depends on whether the renamed database is registered for use with XS advanced and whether XS advanced is installed in the database that is renamed.

1. A tenant database in which XS advanced is installed.
   If you have renamed the tenant database where XS advanced is installed, use the following command to restart XS advanced:

   ```
   $ XSA select-xsa-runtime-db <new tenant database name> --skip-fss
   ```

   This activates XS advanced for the renamed tenant database and then starts XS advanced, too.
2. A tenant database that is registered with XS advanced.
   If you have renamed a tenant database that is registered for use with XS advanced, use the following command to restart XS advanced:

   ```
   $ XSA enable
   ```

   This ensures that, during startup, XS advanced notices the new name for the registered tenant database and adapts all existing service instances that use the renamed database.

Related Information

Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]
Rename a Tenant Database

14.2.2.1.16 Maintaining Host Pinning

Use the `xs` command-line interface to maintain services for applications running in the XS advanced run-time platform.

Host pinning enables you to “pin” an XS advanced application to a particular set of hosts that have been assigned the role xs_worker. In this way, you can control where applications run and ensure they can only be run on a particular host or in a specific space. Pinning applications to a particular host provides the following benefits:

- Isolation
  Ensure that applications run inside the same SAP HANA system but completely isolated from one another on specific (different) hosts.
- Sizing
  Exploit application-specific hardware and sizing benefits which are only available on particular hosts

XS advanced distinguishes between the following host pinning modes:

- `strict`
  The pinned application will only start if one of the pinned hosts is available. (Default)
- `relaxed`
  The pinned application will start even if none of the pinned hosts is available.

Viewing Currently Pinned Hosts

You can display a list of the applications that are currently pinned in the target space. The following example shows that no applications are pinned to any host or space, and no **default** pinning configuration has been set up yet:

Output Code

```
$ xs pinned-hosts

Getting apps in org "orgname" / space "SAP" as XSA_ADMIN...

Found apps:

app                    pinned to    mode
---------------------------------------
auditlog-db            <not set>
auditlog-server        <not set>
auditlog-broker        <not set>
deploy-service         <not set>
product-installer      <not set>

Default host pinning for target space "SAP": <no pinning>
```

Viewing Hosts Available for Pinning

To pin applications to certain hosts, you first need to find out which hosts are available for pinning duties, for example, by using the command `xs system-info`, which displays a list of all hosts running an execution agent (assigned the xs_worker role) along with some additional details, as illustrated in the following example:

Output Code

```
$ xs system-info

Getting system infrastructure information...

Execution agents:
-----------------------------------------------------
index          1
host           host1.example.org:29881
created at     13-Feb-2023 10:18:51
port range     50000-50999
used ports     66 of 1000 (26 by XS advanced apps, 40 reserved for SAP
services)
os.arch        amd64
os.name        Linux
os.version     5.3.18-59.24-default
version        v1.1.0
index          2
host           host2.example.org:23456
created at     13-Feb-2023 10:18:51
port range     50000-50999
used ports     63 of 1000 (24 by XS advanced apps, 40 reserved for SAP
services)
os.arch        amd64
os.name        Linux
os.version     5.3.18-59.24-default
```

```
version        v1.1.0
[...]
```

The system output shown in the example above shows two running Execution Agents (index: 1 and 2), which can be used to pin applications to the hosts where the agents are registered.

Pinning Applications to Hosts

It is possible to pin either specific applications or entire spaces to a particular host or hosts. To pin an application to specific hosts, use the command `xs pin-hosts <application> <host>`. There are two ways to specify the respective host when using the `xs pin-hosts` command:

- Specify the Execution Agent index
  The index is the number associated with the hosts listed by the `xs system-info` command - “1” or “2” in the example above.
- Specify the host name
  The host name is listed along side the “index” in the output displayed by the `xs system-info` command.

The following two commands are equivalent and pin the application `product-installer` to the host `host2.acme.org`:

```
$ xs pin-hosts product-installer --eas 2
```

```
$ xs pin-hosts product-installer --hosts host2.acme.org
```

Assuming you are logged on to the same XS advanced system where you ran the `xs system-info` command in the example, above, the following command pins the `deploy-service` application to the same host where Execution Agent 2 is running, but in `relaxed` pinning mode:

```
$ xs pin-hosts deploy-service --eas 2 --relaxed
```

→ Tip

To change an existing host-pinning setting, use the "`--unset-pinning`" option.

To pin a space to a host, you can use the `xs update-space` command, which works in a similar way to the `xs pin-hosts` command. For example, to pin the space "SAP" to the host `host2.acme.org`, run the following command:

```
$ xs update-space SAP --hosts host2.acme.org
```

To display details of all currently pinned hosts, use the `xs pinned-hosts` command, as illustrated in the following example:

{·} Output Code

```
$ xs pinned-hosts

Getting apps in org "orgname" / space "SAP" as XSA_ADMIN...
Found apps:

app                    pinned to       mode
```

```
-------------------------------------------
auditlog-db            <not set>
auditlog-server        <not set>
auditlog-broker        <not set>
deploy-service         host2.acme.org  relaxed
component-registry-db  <not set>
product-installer      host2.acme.org  strict

Default host pinning for target space "SAP":
  [host2.acme.org] (strict)
```

> **Note**
>
> To apply and enable the new (or modified) host-pinning configuration, you must restart the pinned applications. Explicit host-pinning settings for an application takes precedence over the host-pinning settings for the space in which the application is deployed and running.

Troubleshooting Host-Pinning Problems

Stopping the Execution Agent on a host (for example, `host2.example.org`) by removing the `xs_worker` role from a host illustrates the difference between “strict” and “relaxed” pinning. A short time after the Execution Agent stops, the state of the pinned applications look like in the following example:

Output Code

```
$ xs apps

Getting apps in org "acme" / space "SAP" as XSA_ADMIN...
Found apps:

name              requested  instances  memory   urls
                  state
------------------------------------------------------------------------------
[...]
deploy-service    STARTED    1/1        280 MB   https://deploy-
service.ex[...]
[...]
product-installer STARTED    0/1        64.0 MB  https://product-install[...]
```

The output from the `xs apps` command shows in the `instances` column that the `product-installer` application could not start; this is because the pinning mode was set to “strict”. The `deploy-service` command, however, was able to start because the pinning mode was “relaxed”. The logs for the `product-installer` application contain the information illustrated in the following example:

Output Code

```
$ xs logs product-installer --last 2

------------------------------------------------------------------------------
Connected, dumping recent logs for app "product-installer"
12/15/22 [...PM] [API] ERR  Number of running instances for app 'product-
installer'
12/15/22 [...PM] [API] ERR  Failed to start an instance of app 'product-
installer'
```

Related Information

Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
The XS Command-Line Interface [page 1262]
The XSA Command Reference [page 1370]

14.2.2.1.17 Maintaining Build Packs in XS Advanced

Build packs transform application code into an application droplet that can be deployed to the appropriate run-time environment.

A build pack is the component that XS advanced uses to transform application code into an application droplet, which can be run as an application instance on the XS advanced platform. This transformation process is called "staging". When pushing an application to the run-time environment, the appropriate build pack is automatically detected for the pushed application, and the application is staged using the selected build pack. For example, if you push a WAR file, the Java build pack detects that it is the appropriate build pack to create a usable Tomcat application from that WAR file. Usually a build pack adds the corresponding run time environment (for example, Node.js or Tomcat) along with any specified dependencies (for example, Node modules) to the application being pushed, and it may even add additional libraries (for examples, JDBC drivers) and configure the application to connect to bound services.

Viewing Installed Build Packs

By default XS advanced comes with a set of default build packs that are supported by SAP. The command `xs buildpacks` shows a list of all the build packs installed on the XS advanced platform. The following example shows a list of all default build packs:

```
$ xs buildpacks
Getting buildpacks...
buildpacks             version   position   enabled   locked
-------------------------------------------------------------
sap_java_buildpack     1.8.37    1          true
sap_nodejs_buildpack   4.5.0     2          true      true
sap_python_buildpack   0.5.3     4          true
```

The names of the build packs are generally self-explanatory; in the example above, the build packs names indicate support for Java, Node.js, and Python applications.

The value for the position attribute displayed in the output of the `xs buildpacks` command is important if the build pack needs to be detected automatically. This is because the detection procedures for the various build packs are executed in the order of their positions; in this example: position 1, 2, or 4. The first build pack to report that it has detected an application it thinks it is able to stage is used. Build packs that are `enabled` can be used for staging an application; disabled build packs are not able to detect any uploaded files or indicate that they are responsible for staging a particular type of application. The default setting for `enabled` is "`true`". Locking a build pack ensures that the locked build pack stays at the current version; it cannot be updated with a new version. The default setting for `locked` is "`false`".

Creating a Build Pack

You can use the `xs create-buildpack` command to create your own custom build pack. Creating your own build pack is useful if you want to run applications in XS advanced, which are written in a programming language that is not supported by the default build packs. For example, if you want to run a Go application in XS advanced, you will need to write your own custom build pack for Go. For more information about creating custom build packs, see the section "*Custom Build Packs in XS Advanced*" in the *SAP HANA Developer Guide for XS Advanced* (in *Related Information* below). After you have implemented your custom build pack, you can upload it to the XS advanced platform by running the following command, which would place it at position 5 in the build pack detection process:

```
xs create-buildpack my_custom_buildpack /path/to/my_custom_buildpack 5
```

Updating a Build Pack

You can update a build pack by using the "xs update-buildpack" command, for example:

```
xs update-buildpack my_custom_buildpack -p /path/to/version2/my_custom_buildpack
```

The default build packs provided by SAP are automatically updated with new features and security patches, when you update the XS advanced platform.

Updating a build pack does not directly affect any applications that were staged with the old version of the build pack. If a security patch or a new feature of the build pack is applied to an application, the application needs to be restaged. Restaging happens when the application is deployed or pushed. A restage, however, can also be triggered individually. After restaging an application, it is necessary to restart the application, too, in order to ensure the execution of the new droplet produced by the new version of the build pack, as illustrated in the following example:

```
xs restage <app_name>
xs restart <app_name>
```

It is also possible to enforce a restage of all applications. However the procedure requires a restart of XS advanced. To trigger a restart of all application, log on to XS Advanced as the `<sid>adm` user and run the following (`XSA`) commands:

```
XSA restage-at-startup
XSA restart
```

Using Git Build Packs

Creating and updating a build pack as described in this section, requires administrator privileges within the XS advanced platform. However, when pushing an application, it is also possible to specify a git repository that contains the source code of a build pack, as illustrated in the following example

```
xs push my-application -f my-manifest.yml -b https://my-git-server.example.com/
my_custom_buildpack.git
```

In this scenario, the build pack is downloaded from the specified git repository and used when staging the pushed application. Note that it is only possible to specify a Git repository URI that is also valid for the `git clone` command. The URI can use either the `git` or the `https` protocol and needs to be reachable from the XS Controller host, and the source code of the build pack must to be available in the default branch of the repository. Note that if you are calling a Git server via HTTPs, it might be necessary to configure the correct trust certificates first. For more information about how to configure trust certificates within XS advanced, see *Maintaining Trust Certificates* in *Related Information* below.

Related Information

Maintaining Trust Certificates in XS Advanced [page 1321]
SAP HANA Developer Guide for XS Advanced Model (SAP Web IDE)

14.2.2.1.18 Maintaining Service URLs

The SAP HANA XS advanced platform supports the creation and maintenance of a list of service URLs, which consist of a service name and the corresponding URL. Some service URL mappings are created during installation of XS advanced, but additional URLs can be registered by an administrator at any point in time. Usually, the service URL points to an application running on the XS advanced platform, but the URL could also point to an external service.

This list of registered service URLs can be read by client tools, for example, the `xs` command-line tool without requiring any logon credentials. If the XS Controller API URL is set, the known service URLs are displayed in the output of the `xs version` command, as illustrated in the following example:

Output Code

Registered Service URLs

```
$ xs version
...
------------------------------------------------------------------------------
Registered service URLs:
  deploy-service          = https://deploy-service.xsa.example.com:35033
  product-installer       = https://orgname-sap-product-
installer.xsa.examp...
  job-scheduler-dashboard = https://jobscheduler-
dashboard.xsa.example.com:350
  product-installer-ui    = https://orgname-sap-product-installer-ui.xsa.ex...
  xsa-cockpit             = https://xsa-cockpit.xsa.example.com:35033
```

Links to the services are also created on the status page that is delivered to a Web browser that is pointed to the XS advanced API URL. In addition, the platform provides a redirection mechanism for the registered services. Pointing a Web browser to `<API-URL>/go/xsa-cockpit` redirects the browser to the URL registered for the service `xsa-cockpit`. This is particularly useful if you want to provide stable and predictable links to front-end applications even in **port**-routing mode without having to create routes with static ports.

Registering Service URLs

Service URLs for applications deployed on the XS advanced platform can be registered automatically using the corresponding feature in the application's `mtad.yaml` deployment descriptor. XS advanced administrators can, however, also register a service URL manually, using the `xs register-service-url` command, as illustrated in the following example:

Output Code

Register a Service URL

```
$ xs register-service-url myservice https://myservice.xsa.example.com
```

To display a list of all currently registered service URLs, run the following command:

Output Code

List all Registered Service URLs

```
$ xs service-urls
...
------------------------------------------------------------------------------
Registered service URLs:
 deploy-service          = https://deploy-service.xsa.example.com:35033
 product-installer       = https://orgname-sap-product-installer.xsa.exampl...
 job-scheduler-dashboard = https://jobscheduler-dashboard.xsa.example.com:3503
 product-installer-ui    = https://orgname-sap-product-installer-ui.xsa.exa...
 xsa-cockpit             = https://xsa-cockpit.xsa.example.com:35033
```

Unregistering Service URLs

Service URLs can be manually removed from this list of registered service URLs by using the `xs unregister-service-url` command, as illustrated in the following example:

Output Code

Unregister a Service URL

```
$ xs unregister-service-url myservice
...
Removed registration of service name "myservice" with URL "https://
myservice.xsa.example.com".
OK
```

Related Information

Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]

14.2.2.1.19 Building Roles for XS Advanced Applications

Use authorization artifacts to control access to XS advanced applications.

In XS advanced, application developers create and deploy application-based authorization artifacts for business users. Administrators use this information to build roles, define sets of roles called role collections, and assign these collections to business users or user groups. In this way, they control the users' permissions and, as a result, access to the applications.

Role Templates

Applications define role templates acting as a blueprint for real-life role instances. Based on these role templates, administrators create role instances by filling in concrete values for attributes defined by role templates. For example a role template for editing actions on HR data could have an attribute to configure the region or country to restrict the editing permission to. To display the role templates provided by applications in the current target space or made available globally, use the command `xs role-templates`, as illustrated in the following example:

Output Code

Role Templates for XS Advanced Applications

```
$ xs role-templates

Getting role templates in space "SAP" as user "XSA_ADMIN"...

app               role template   attributes           description
-------------------------------------------------------------------------
alm               ControllerAdmin                      Template for Ctrl Admin
xs_role_admin     XS_ROLE_ADMIN                        Auths for XS role build
fileprocessor     Admin                                Role for Admin UI
                  API                                  Role for API access
                  Auditor                              Role for read-only acce
java-hello-world  Viewer          Country              View all books
                  Editor          Country, CostCenter  Edit, delete the books
```

In the example above, the role templates “Viewer” and “Editor” of the application `java-hello-world` are examples for role templates with attributes (`Country` and `CostCenter`. It is not possible to assign role templates directly to users; a specific instance of the role template must be created first, in the form of a role, which is described in more detail in the next section.

Roles in XS Advanced Applications

If you are basing a role on a role template that does not specify any attributes, creating the role is a straight-forward process, for example, using the `xs create-role` command, as illustrated in the following example:

```
$ xs create-role fileprocessor Admin FP-Admin "Role for fileprocessor Admin UI
access"
Creating role "FP-Admin" for app "fileprocessor" using role template "Admin" as
user "XSA_ADMIN"...
```

```
OK
```

The example command above creates a new role called `FP-Admin` from the `Admin` template of the `fileprocessor` application. No attributes are specified in this example, but it is always a good idea to provide a description for the new role; this information is used when the role instance is created and appear in the list of roles that can be displayed with the `xs roles` command, as illustrated in the following example:

Output Code

Displaying XS Advanced Application Roles

```
$ xs roles

Getting application roles in space "SAP" as user "XSA_ADMIN"...
Found roles:

name                              app
template           ...
------------------------------------------------------------------------------
XS_CONTROLLER_ADMIN               <cloud_controller>
XS_CONTROLLER_ADMIN
XS_CONTROLLER_ADMIN_READ_ONLY     <cloud_controller>
XS_CONTROLLER_ADMIN_READ_ONLY
XS_CONTROLLER_AUDITOR             <cloud_controller>
XS_CONTROLLER_AUDITOR
XS_CONTROLLER_CREDENTIALS_VIEWER  <cloud_controller>
XS_CONTROLLER_CREDENTIALS_VIEWER
XS_CONTROLLER_GLOBAL_AUDITOR      <cloud_controller>
XS_CONTROLLER_GLOBAL_AUDITOR
XS_CONTROLLER_USER                <cloud_controller>
XS_CONTROLLER_USER
XS_AUTHORIZATION_ADMIN            <xs_authorization>
XS_AUTHORIZATION_ADMIN
XS_AUTHORIZATION_DISPLAY          <xs_authorization>
XS_AUTHORIZATION_DISPLAY
XS_MONITOR_ADMIN                  <xs_monitor>
XS_MONITOR_ADMIN
XS_MONITOR_DISPLAY                <xs_monitor>
XS_MONITOR_DISPLAY
XS_USER_ADMIN                     <xs_user>
XS_USER_ADMIN
XS_USER_DISPLAY                   <xs_user>           XS_USER_DISPLAY
XS_USER_PUBLIC                    <xs_user>
XS_USER_PUBLIC
ControllerAdmin          alm                ControllerAdmin          Default
XS_ROLE_ADMIN            xs_role_admin      XS_ROLE_ADMIN            Default
API                      fileprocessor      API                      Default
Admin                    fileprocessor      Admin                    Default
Auditor                  fileprocessor      Auditor                  Default
FP-Admin                 fileprocessor      Admin                    Admin UI
```

The `xs roles` command shows all application roles scoped for the current target space and any global roles. In addition to roles created from application-defined role templates, there are also some roles defined by the platform itself. Platform-defined roles typically have names starting with "`XS_`" and the origin of the role is shown in angle brackets `<...>` to distinguish them from normal application roles.

When creating a role from a role template with attributes, the `xs create-role` command prompts the user to specify the values to use for all the attributes. Alternatively, the attribute values can be provided on the command line with the option -a ATTRIBUTES where ATTRIBUTES can either be a JSON format structure with

all attribute values, or the path to a text file that contains the JSON structure. A convenient way to create such a file is by letting the xs client create it for a given role template, as illustrated in the following example:

Output Code

Creating a Sample `attributes.json` File for a Role Template

```
$ xs create-role java-hello-world Editor --create-sample attributes.json
Creating attribute values sample file for role template "Editor" for app
"java-hello-world"...
OK
```

The resulting file `attributes.json` can then be edited in any text editor to fill in the desired values. To create a new role with these attributes and the edited values, use the option -a attributes.json as illustrated in the following example:

Output Code

Creating an Application Role Using Attribute Values Stored in a JSON File

```
$ xs create-role java-hello-world Editor Editor-Germany -a attributes.json
"Editor role for Germany"
Creating role "Editor-Germany" for app "java-hello-world" using role template
"Editor" as user "XSA_ADMIN"...
OK
```

The details of the newly created (or any other) role can be displayed using the `xs role` command. The details include a description of the role, the provided scopes, and any attribute values, as illustrated in the following example:

Output Code

Displaying Details of an Application Role

```
$ xs role Editor-Germany

Getting role "Editor-Germany" in space "SAP" as user "XSA_ADMIN"...

name:        Editor-Germany
description: Editor role for Germany
app:         java-hello-world
template:    Editor (Edit and Delete the books)

scope                        description
-----------------------------------------
java-hello-world!i1.Create   create
java-hello-world!i1.Delete   delete

attribute   value     source   description
-----------------------------------------
Country     Germany   static   Country
CostCenter  de-01     static   CostCenter
```

Role Collections in XS Advanced

In XS advanced, roles are not directly assigned to users; they are managed in reusable role collections that can, for example, list all the roles required to perform a specific task. New role collections can be created using the `xs create-role-collection` command, as illustrated in the following example:

Output Code

Creating a Role Collection

```
$ xs create-role-collection Example-Collection "An example role collection"
Creating role collection "Example-Collection" as user "XSA_ADMIN"...
OK
```

To display existing role collections, use the command `xs role-collections`, as illustrated in the following example:

Output Code

Displaying a List of Role Collections

```
$ xs role-collections

Getting role collections as user "XSA_ADMIN"...

role collection                    description
-------------------------------------------------------------------------
XS_AUTHORIZATION_ADMIN             Authorizations for XS role builder
XS_AUTHORIZATION_DISPLAY           Authorizations for XS role viewer
XS_USER_ADMIN                      Admin authorizations for XS user management
XS_USER_DISPLAY                    Display authorizations for XS user
management
XS_USER_PUBLIC                     Default authorizations for XS user
XS_MONITOR_ADMIN                   Authorizations for XS monitoring management
XS_MONITOR_DISPLAY                 Authorizations for XS monitoring display
XS_CONTROLLER_ADMIN                Authorizations for XS controller admin
XS_CONTROLLER_ADMIN_READ_ONLY      Authorizations for XS controller read-only
admin
XS_CONTROLLER_GLOBAL_AUDITOR       Authorizations for XS controller global
auditor
XS_CONTROLLER_USER                 Authorizations for XS controller user
XS_CONTROLLER_AUDITOR              Authorizations for XS controller auditor
XS_CONTROLLER_CREDENTIALS_VIEWER   Authorizations for XS controller
credentials viewer
Example-Collection             An example role collection
```

You can add individual roles to (or remove them from) a role collection using the command `xs update-role-collection`, as illustrated in the following example:

Output Code

Adding a Role to a Role Collection

```
$ xs update-role-collection Example-Collection --add-role FP-Admin
Getting role "FP-Admin" in space "SAP" as user "XSA_ADMIN"...
Updating role collection "Example-Collection" as user "XSA_ADMIN"...
OK
```

To show the contents of a role collection, use the command `xs role-collection`, as shown in the following example:

Output Code

Displaying Details of a Role Collection

```
$ xs role-collection Example-Collection

Getting role collection "Example-Collection" as user "XSA_ADMIN"...
Roles of role collection "Example-Collection" (An example role collection):

name            app               space        template  description
-------------------------------------------------------------------------------
FP-Admin        fileprocessor     orgname:SAP  Admin     Role Admin UI access
Editor-Germany  java-hello-world  orgname:SAP  Editor    Edit role for Germany
```

Assigning Role Collections

The final step in role management is assigning the role collection to users. The XS command-line client provides the `assign-role-collection` command for this purpose, as shown in the following example:

Output Code

Assigning a Role Collection to an XS Advanced User

```
$ xs assign-role-collection Example-Collection MYUSER

Assigning role collection "Example-Collection" to user "MYUSER"...
OK
```

Use the `xs unassign-role-collection` command to remove any role collections assigned to users. To display the role collections currently assigned to a user, use the `xs assigned-role-collections` command, as illustrated in the following example:

Output Code

Displaying Details of Assigned Role Collections

```
$ xs assigned-role-collections MYUSER

Getting role collections assigned to user "MYUSER"...

role collection       description
-------------------------------------------------------
XS_USER_PUBLIC        Default authorizations for XS user
Example-Collection    An example role collection
```

Related Information

Maintaining Platform Users in XS Advanced [page 1281]

14.2.2.1.20 Maintaining XS Advanced Application Runtimes

Keep XS advanced application run-time versions up to date.

Applications runtimes are used by buildpacks to compile an application during the staging process. XS advanced keeps a store for application runtimes such as virtual machines for Java, Node.js, application containers, and so on. The application run-time store is automatically updated with most recent version of the application runtime when XS advanced is updated to a new version. However, the XS advanced administrator can also maintain application runtimes manually.

Viewing Installed Application Runtimes

To display a list of the installed run-time environments in XS advanced, use the `xs runtimes` command, as illustrated in the following example:

Output Code

Listing Installed Run-Time Environments

```
$ xs runtimes

Getting runtimes...

type               version    id  resolved  active  maintenance  bound
descr
                                                                 apps
-------------------------------------------------------------------------------
hanajdbc2          2.19.17    28  true      true                 2      SAP
node16.20          16.20.1.0  6   true      true    outdated     3      Node.
node18             18.16.1    10  true      true                 2      Node.
sapjvm8_jre        8.1.96     11  true      true    outdated     0      SAP JV
sapjvm8_jre        8.1.97     31  true      true                 4      SAP JV
sapmachine11_jre   11.0.22    34  true      true                 0      SapMac
sapmachine17_jre   17.0.10    36  true      true                 0      SapMac
tomcat10           10.1.18    2   true      true                 0      Apache
tomcat9            9.0.85     3   true      true                 4      Apache
tomee7.0_webprof   7.0.9.39   29  true      true                 0      Apache
tomee9.1_plume     9.1.2      4   true      true                 0      Apache
```

For each application run-time environment, the following information is displayed:

Application Run-Time Environment Details

Information	Description
type	The name of the application run-time environment
version	The version of the application run-time environment
id	The index of the application runtime by which it is referenced in other xs CLI commands (for example, `xs runtime`)

Information	Description
resolved	The indicated application runtime is correctly detected by the XS advanced run-time environment [true | false]
active	The application runtime is enabled for use by build packs [true | false]
maintenance	Flags a runtime as "outdated" when it is out of maintenance or a newer version of the same type of runtime exists in the system
bound apps	The number of applications currently using the application runtime
description	A short summary of the application run-time details

Finding Bound Applications

To display a list of all applications that use a particular application run-time environment, use the `xs runtime` command and specify a particular runtime with the run-time index, as illustrated in the following example:

→ Tip

The run-time index is displayed in the `id` column of the output of the `xs runtimes` command.

{·} Output Code

Listing Applications Bound to a Specific Runtime

```
$ xs runtime -i 2

Showing information about "node18"
  type:        node18
  version:     18.14.0
  id:          2
  description: Node.js 18.14.0 for Linux x64
  resolved:    true
  active:      true
  bound apps:
  org   space   apps              requested state   currently used   used
after restart
-------------------------------------------------------------------------------
----------
  org   SAP     auditlog-db       STOPPED           no               yes
         --     auditlog-broker   STARTED           yes              yes
```

In addition to the organization, space and application name, you also display the information if the app is currently started, The last two columns provide information about how the application is bound to the specified runtime:

- `currently used`:
  Indicates if there are currently running instances of the application that are using the specified runtime.
- `used after restart`:
  Indicates if the specified runtime will be used when the application is restarted.

You can use the information displayed by the `xs runtime` command to check which applications need to be restaged if a new application runtime has been installed and, in addition, to ensure that all relevant applications are using the specified application run-time environment.

Creating and Uploading a New Runtime

You can upload a new runtime to XS advanced with the command `xs create-runtime`; you need to specify the directory or .zip file containing the application runtime that you want to upload, as illustrated in the following example:

```
$ xs create-runtime -p Python-2.7.6
```

> ⚠ Restriction
>
> Only application runtimes supported by XS advanced can be uploaded, for example: SAP JVM, SAP Node.js, Python, Tomcat, TomEE, and HANA JDBC driver.

Updating Runtimes

When updating XS advanced, the default application runtimes are automatically maintained and updated as well, for example, to provide security patches.

> ⓘ Note
>
> Manually uploaded application runtimes are not included in the automatic update.

To ensure that all applications use the updated application runtimes, all applications must be restaged and restarted. You can use the `xs restage` command with the `--outdated` option, as illustrated in the following example:

```
$ xs restage --outdated
```

For more information about the `xs restage` command with the `--outdated` option, see the section *Restaging Applications on Outdated Runtimes* in *Maintaining Applications in XS Advanced* in *Related Information* below.

Alternatively, you can use the commands `xs restage` and `xs restart` to restage and restart individual applications manually.

Removing Outdated Runtimes

Although an update of XS advanced may install new versions of runtimes, the update process does not automatically remove old run-time versions. Existing applications could still use the old run-time version and need to be restaged to update to the latest application runtime. See section Updating Runtimes [page 1353] above.

All outdated application runtimes that are no longer bound to any applications can be removed by using the xs delete-runtimes command with the --outdated option, as illustrated in the following example:

Removing Outdated Runtimes with Bulk Operations

To remove outdated application runtimes as part of a bulk operation, perform the following steps:

1. Make sure that existing applications are updated to the most recent application runtime versions:

   ```
   $ xs restage --outdated
   ```

2. Remove outdated runtimes by running the following command:

   ```
   $ xs delete-runtime --outdated
   ```

3. If an outdated application runtime cannot be deleted because it still has applications that are bound to it, then restage the applications again with the option --delete-app-instances, as illustrated in the following example:

   ```
   $ xs restage --outdated --delete-app-instances
   ```

   Then repeat step 2 above.

> → Tip
>
> If after performing these steps, an outdated application runtime has still applications bound to it, it means that the application does not support newer versions of the application runtime, and you must updated the application before restaging the application as described in the steps above.

Removing Outdated Runtimes without Bulk Operations

Make sure that the runtime selected for removal is no longer used by any applications, for example, by using the `xs runtime` command with the -i option to display details about the runtime you want to remove, as shown in the following example:

```
$ xs runtime -i 7

Showing information about "node10.23"
  type:        node10.23
  version:     1.0
  id:          7
  description: Node.js 10.23.1 for Linux x86-64
  resolved:    true
  active:      true
  maintenance: outdated
  bound apps:
  org   space   apps              requested state   currently used   used after
restart
-------------------------------------------------------------------------------
-------
  org   SAP     auditlog-db       STOPPED           no               yes
         --     auditlog-broker   STARTED           yes              yes
```

To make sure that a runtime is no longer in use and can safely be removed, perform the following steps:

1. Restage any bound application that shows a "`yes`" in the "`used after restart`" column.
2. If, after restaging, there is still a "`yes`" in the "`used after restart`" column, then the application is not compatible with a newer version of the runtime. In this case, the application itself has to be updated.
3. When all bound applications display "`no`" in the "`used after restart`" column, restart all applications that display "`yes`" in the "`currently used`" column.

4. At this point, there should be no application with "`yes`" in either the `currently used` or `used after restart` columns, which confirms that the outdated runtime is no longer being actively used by any application. However, applications could still be bound to this runtime, for example, due to stopped application instances that are kept for the purposes of post-mortem analysis. For this reason, it is recommended to use the `xs update-runtime` command to disable the outdated runtime and prevent any future use by any application, as shown in the following example:

   ```
   xs update-runtime -i <ID> --disable
   ```

5. To completely remove the runtime, delete any existing stopped application instances that are still referencing the outdated runtime. You can do this by executing the following command for each bound application:

   ```
   xs delete-app-instances <APP> --stopped --crashed
   ```

6. Now, you can safely remove the runtime, as follows:

   ```
   xs delete-runtime -i <ID>
   ```

Pinning Run-time Environments to Applications

Normally, the buildpack chooses the latest version of an application runtime in the version range specified by the application during application staging. However, in support cases, and if the application is compatible, you can pin a specific version of a runtime to an application, forcing the build pack to use the specified run-time version during application staging. For example using the command `xs pin-runtime` and the options `<ALIAS>`, `<APP>`, and `-i <PINNED RUNTIME>`, you can ensure that the application runtime with index `<PINNED RUNTIME>` is always used when an application build pack requests a runtime with the name `<ALIAS>`.

The `<ALIAS>` can either be an exact run-time type (for example, "`node14.21`" to pin a fixed run-time version if the application build pack requires the Node.js version 14.21) or a prefix of a run-time type (for example, "`node14`" to pin a fixed run-time version if the application build pack requires any version of Node.js version 14.*).

> ⚠ Caution
>
> Support for run-time pinning is provided for Java and Node.js types only.

In the following example, a Node.js application is pinned to a Node.js “14” run-time **type** despite the fact that the buildpack would normally choose a Node.js runtime of type “16”:

> → Tip
>
> The run-time type is displayed in the `type` column of the output of the `xs runtimes` command.

> {·} Output Code
>
> Pinning Applications to a Run-Time Type
>
> ```
> $ xs pin-runtime node16 myapp -i 11
> Updating app "myapp" in org "myorg" / space "PROD" as XSA_ADMIN...
> ```

To ensure that the specified changes take effect, the target application must be re-staged and re-started. You can use the `xs pinned-runtimes` to display a list of the mappings between runtimes and applications, and filter the output for a specific application name, as illustrated in the following example:

Output Code

Displaying Pinned Runtimes

```
$ xs pinned-runtimes myapp

Showing pinned runtimes for app "myapp".

alias   type      version  id  resolved active description              bound
                                                                        apps
-------------------------------------------------------------------------------
node16  node14.21 14.21.2  11  true     true   Node.js 14.21.2 Linux x86...  0
```

To undo the changes, and revert to the previous run-time settings, use the `xs unpin-runtime` command, as illustrated in the following example:

Output Code

Removing Application-to-Run-Time Pinnings

```
$ xs unpin-runtime node14 myapp

Updating app "myapp" in org "myorg" / space "PROD" as XSA_ADMIN...
```

Related Information

Maintaining Applications in XS Advanced [page 1297]

14.2.2.1.21 Maintaining Tasks in XS Advanced

Schedule and manage one-off tasks with XS advanced applications.

XS advanced applications are Web applications that on startup open a port and then remain available to respond to user requests until the applications are stopped. This type of application life-cycle is not appropriate for all types of work. For example, it is not appropriate for initialization or setup tasks that need to be performed only once and do not require any user interaction; these types of problems are best handled by using so called one-off tasks. One-off tasks are special application instances that are started by specifying a particular custom command on application startup. No route is created for a one-off task, which means that the custom command can perform any type of activity that does not require user interaction. After the custom command exits, the one-off tasks finishes and reports the command's exit code to the user.

Running a One-Off Task

Before using an application to perform a one-off task, the corresponding application must first be deployed, as described in *Maintaining Applications* in *Related Information*. After the application has been successfully deployed, the command `xs run-task` can be used to create a new task instance to run it. The following arguments have to be used to specify the required information in the `run-task` command:

1. The name of the application (and therefore the droplet files for the task execution)
2. A name for the new task
3. The command to execute

The following example shows what a typical task start command looks like:

Output Code

```
$ xs run-task my-app my-task "echo Hello World" --wait-for-completion
Running task "my-task" on app "my-app"...
Task finished successfully.
```

The new task "`my-task`" is started based on the current droplet and environment variable settings of the application "`my-app`". Any additional environment variable values that are required can be provided by using the option `-e` with the command `run-task`. The option `--wait-for-completion` forces the client to wait for the task execution to finish. With this option, the success (or failure) of the task execution is reported at the end; without this option, the task execution is started asynchronously and the `run-task` command returns immediately.

→ Tip

You can display the current state of a task by using the `xs tasks` command.

Checking the Current State of Tasks

The command `xs tasks` is used to display a list of existing tasks for a given application and check their current state, as shown in the following example:

Output Code

```
$ xs tasks my-app

Listing tasks for app "my-app"...

index  name      created                  command              state
-------------------------------------------------------------------------------
7      bad-task  Apr 25, 2022 10:07:43 AM  non-existing command  FAILED
       Failure reason: Process terminated with exit code 1
8      my-task   Apr 25, 2022 10:08:22 AM  echo Hello World      SUCCEEDED
```

Although failed tasks have an additional short failure reason message, more useful information can be found in the task-execution output.

Displaying Task Output

The task execution is very similar to the execution of regular apps. Therefore, the output of the task can be found in the logs of the corresponding application. As documented in Displaying Application Logs, the command xs logs is used to display the application logs. For the example above, the log output could look like this:

Output Code

```
$ xs logs my-app --recent
Connected, dumping recent logs for app "my-app"...
------------------------------------------------------------------------------
4/25/22 10:07:43.385 AM [API] OUT Starting new task instance '39660f9d-6eed...
4/25/22 10:07:45.124 AM [APP/1-7/bad-task] OUT
4/25/22 10:07:45.124 AM [APP/1-7/bad-task] OUT /hana/shared/ABC/xs/app_work...
4/25/22 10:07:45.127 AM [APP/1-7/bad-task] ERR 'non-existing...
4/25/22 10:07:45.127 AM [APP/1-7/bad-task] ERR operable program or batch fi...
4/25/22 10:07:49.043 AM [API] ERR Crashed instance [state CRASHED, index 7]...
4/25/22 10:08:22.774 AM [API] OUT Starting new task instance '09d52632-d616...
4/25/22 10:08:24.014 AM [APP/1-8/my-task] OUT
4/25/22 10:08:24.014 AM [APP/1-8/my-task] OUT /hana/shared/ABC/app_working/...
4/25/22 10:08:24.018 AM [APP/1-8/my-task] OUT Hello World
```

The log output reveals details of the failed task, which in the example above concerns the use of an unrecognized command (`non-existing`). The log lines relevant for a given task can be found by looking for the task name and task index within the log line **source** string, which is enclosed in square brackets in the log output, for example, `[APP/1-7/bad-task]`. The name and index associated with a task are displayed in the output of the `xs tasks` command.

The successful `"Hello World"` task shown in the log output has the index “`8`” and the name “`my-task`” and it was executed using the first droplet of the `my-app` application. As a result, the output of this task is marked with the source string `[APP/1-8/my-task]`.

Related Information

Maintaining Applications in XS Advanced [page 1297]
The XS Command-Line Interface [page 1262]

14.2.2.1.22 Maintaining Identity Providers

This document describes how to maintain SAML identity providers (IdPs) in XS Advanced.

Maintaining Identity Providers

You can connect SAML identity providers (IdP), for example Active Directory (AD) or Keycloak Open Source Identity and Access Management, to XS Advanced. This way, you can log in with users maintained by such an IdP.

To configure a SAML IdP in XS Advanced, you need to perform the following tasks:

At the SAML IdP:

1. Generate the UAA trust XML metadata and configure it at the SAML IdP, see Generate the Metadata Required to Configure an Identity Provider.
2. Define SAML groups that you want to map to role collections, and assign users to these groups.

In XS Advanced:

1. Create an IdP trust configuration for the SAML identity provider.
2. Map SAML groups to role collections defined by applications or by the platform.

These steps are described in detail in the following sections.

> ⓘ Note
>
> To use the commands described on this page, make sure to use at least XS Advanced version 1.4.4.

For each SAML identity provider that shall be connected to XS Advanced, an IdP trust configuration -- also referred to as *'origin'* -- must be maintained.

Listing Available Identity Providers

To list existing identity providers that are currently connected to XS Advanced, use the command `xs idendity-providers`:

```
$ xs identity-providers
Getting registered identity providers as user "XSA_ADMIN"...
status   name       origin key   description
------------------------------------------------------------------------
Active   Keycloak   myidp        Keycloak Identity and Access Management
```

> ⓘ Note
>
> Please, note, that the `origin key` is a unique identifier of an IdP and must be used to reference an IdP when using other IdP related xs cli commands, e.g. to maintain role collection mappings, via the `--origin` option.

Creating a new Identity Provider Trust Configuration

To connect a new IdP to XS Advanced, use the command `xs create-identity-provider:`

```
$ xs create-identity-provider Keycloak --origin myidp --metadata saml-
metadata.xml --description "Keycloak Identity and Access Management"
Adding new identity provider "Keycloak" with origin key "myidp"...
OK
Created identity provider configuration:
------------------------------------------------------------------------------
----------------------
name                                Keycloak
origin key                          myidp
description                         Keycloak Identity and Access Management
active                              yes
entity ID                           0289f8fa-966a-4797-9f9d-b533776abf13
link text                           Keycloak
create shadow users during login    yes
name ID                             <unspecified>
attribute mappings
{"given_name":"first_name","family_name":"last_name","email":"mail"}
metadata                            <skipped>
```

This command has several further options for configuring the new IdP (refer to The SAP HANA XS Command-Line Interface Reference for a full list). Mandatory options are:

Option	Description
the first argument	a descriptive name for the new IdP
`--origin`	the origin key as a unique identifier for the IdP
`--metadata`	the name of an XML file with the SAML metadata downloaded from the IdP

Other options include:

Option	Description
`--description`	An informal description for the IdP.
`--show-link`	Enables a clickable link to this IdP on the UAA login page (The default is `true`, disable with `--show-link=false`).
`--link-text`	A custom text for the login link. By default, the name of the IdP will be used.
`--add-shadow-user-on-login`	Determines whether users should be allowed to authenticate without having a user pre-populated in the users database (if true, this is the default), or whether shadow users must be created before login by an administrator (if false).
`--attribute-mappings`	The name of a JSON file with mappings from UAA attribute names to external attributes in the provider assertion or token.
`--name-id`	The provider-specific name ID format that is used to uniquely identify a user.

Configuring SAML Assertion Attribute Mappings

By default, the UAA expects the following SAML assertion user attribute names:

UAA Attribute Name	Description
`email`	The email address of a user.
`given_name`	The first name of a user.
`family_name`	The family name of a user.
`phone_number`	The phone number of a user.
`email_verified`	A flag if the email address was verified.
`external_groups`	External group names.
`user.attribute.<attribute_name>`	A custom user attribute.

If your IdP uses different names for these attributes in the SAML assertion and it's not possible to adapt these names at the IdP, you can map the UAA attribute name to the attribute name of your IdP using the `--attribute-mappings` option described above. This way, XS Advanced applications will have access to user attributes nevertheless.

Example:

```
{ "given_name": "urn:oid:2.5.4.42", "family_name": "urn:oid:2.5.4.4", "email":
"urn:oid:1.2.840.113549.1.9.1" }
```

Configuring the NAME ID Format

This configuration determines the user ID by which the UAA references an IdP user. This setting influences how

- you reference an IdP user with the xs cli tool
- XS Advanced applications show the name of a user

The values you can provide for the `--name-id` option depend on your IdP. Prominent settings include:

Example Value	Description
`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`	This is the default setting. It depends on the IdP which user ID is used.
`urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`	Use the email address of a user to identify it in XS Advanced.

Updating Existing Identity Provider Configurations

All IdP configuration options of an existing origin can be updated using the command `xs update-identity-provider`, for example:

```
$ xs update-identity-provider --origin myidp --link-text "Login with Keycloak"
Updating identity provider "Keycloak" with origin key "myidp"...
OK
Updated identity provider configuration:
------------------------------------------------------------------------------
----------------------
name                            Keycloak
origin key                      myidp
description                     Keycloak Identity and Access Management
active                          yes
entity ID                       0289f8fa-966a-4797-9f9d-b533776abf13
```

```
link text                           Login with Keycloak
create shadow users during login    yes
name ID                             <unspecified>
attribute mappings
{"given_name":"first_name","family_name":"last_name","email":"mail"}
metadata                            <skipped>
```

The configuration to update is selected by the provided origin key. The command can be used to update configuration values using the same options as during creation above.

Getting the Current Configuration of an IdP

To display the current configuration of an identity provider, use the command `xs identity-provider`:

```
$ xs identity-provider --origin myidp
Identity provider configuration:
------------------------------------------------------------------------------
----------------------
name                                Keycloak
origin key                          myidp
description                         Keycloak Identity and Access Management
active                              yes
entity ID                           0289f8fa-966a-4797-9f9d-b533776abf13
link text                           Login with Keycloak
create shadow users during login    yes
name ID                             <unspecified>
attribute mappings
{"given_name":"first_name","family_name":"last_name","email":"mail"}
metadata                            <skipped>
```

The technical SAML metadata information of the provider is skipped from the output of the command by default, but can be enabled by providing the command line option `--metadata`.

Maintaining SAML Group to Role Collection Mappings

Unlike for SAP HANA database users, you don't assign role collections directly to users maintained in an IdP. Instead, you create mappings between SAML groups and XS Advanced role collections. Afterwards, in the IdP, you assign SAML groups to a user. The role collection mappings ensure that, when the user logs in at XS Advanced, the permissions of the mapped role collections are granted.

Listing SAML Group to Role Collection Mappings

The following command `xs role-collection-mappings` shows all available role collection mappings:

```
$ xs role-collection-mappings --origin myidp
Getting all role collection mappings for IdP "Keycloak" (origin: myidp) as user
"XSA_ADMIN"...
SAML group   maps to Role Collection
-----------------------------------
xsa-user     XS_CONTROLLER_USER
auth_admin   XS_AUTHORIZATION_ADMIN
user_admin   XS_USER_ADMIN
auditor      XS_CONTROLLER_AUDITOR
editor       MYAPP_EDITOR
```

Creating a SAML Group to Role Collection Mapping

To create a new SAML group to role collection mapping, specify the SAML group and the role collection name use the following command:

```
$ xs create-role-collection-mapping --origin myidp GROUP_CONTROLLER_ADMIN
XS_CONTROLLER_ADMIN
Creating role collection mapping for IdP "Keycloak" (origin:  myidp) as user
"XSA_ADMIN"...
Added mapping: GROUP_CONTROLLER_ADMIN -> XS_CONTROLLER_ADMIN (Authorizations for
XS controller admin)
OK
```

Deleting a SAML Group to Role Collection Mapping

You can delete an existing role collection mapping as follows:

```
$ xs delete-role-collection-mapping --origin myidp GROUP_CONTROLLER_ADMIN
XS_CONTROLLER_ADMIN
Do you want to delete the role collection mapping? (y/n) > y
Deleting role collection mapping for IdP "Keycloak" (origin: myidp) as user
"XSA_ADMIN"...
Deleted mapping: GROUP_CONTROLLER_ADMIN -> XS_CONTROLLER_ADMIN (Authorizations
for XS controller admin)
OK
```

Exporting and Importing SAML Group to Role Collection Mappings

Role collection mappings are attached to a specific IdP. In order to use the same mappings for another IdP, even in another system, you can export SAML group mappings to a file:

```
$ xs role-collection-mappings --origin myidp --export mymappings.json
Exporting all role collection mappings for IdP "Keycloak" (origin: myidp) to
file "mymappings.json" as user "XSA_ADMIN"...
Successfully exported 5 mappings.
```

Afterwards, you can use the generated file and import it, e.g. at another IdP:

```
$ xs role-collection-mappings --origin newidp --import mymappings.json
Importing role collection mappings for IdP "New SAML IdP" (origin: newidp) from
file "mymappings.json" as user "XSA_ADMIN"...
Found 5 mappings in import file.
Importing mappings from file...
  Skipping mapping for SAML group "xsa-user" to role collection
"XS_CONTROLLER_USER": this mapping is already present
Imported 4 new role collection mappings.
New effective mappings:
SAML group   maps to Role Collection
-------------------------------------
xsa-user     XS_CONTROLLER_USER
auth_admin   XS_AUTHORIZATION_ADMIN
user_admin   XS_USER_ADMIN
editor       MYAPP_EDITOR
auditor      XS_CONTROLLER_AUDITOR
```

14.2.2.1.23 Scanning and Reporting XS Advanced Application Artifacts

Generate and display reports with details of dependencies between application artifacts and resources in XS advanced.

Web applications typically include a large number of software components such as Java archives (`.jar`) or Node.js modules (`.js`). In most cases, these artifacts are third-party (probably open-source) components that need special attention. And these components are not only in the application bundle pushed to the server; a significant amount of additional artifacts can be added by the platform itself during application deployment, for example: UI-framework artifacts, middleware, or low-level modules such as database drivers or cryptographic libraries. Regardless of origin, all artifacts in a staged application potentially contribute to productive code. For this reason, it is crucial to be able to get a complete overview of all artifacts that an application references either directly or indirectly, in order to identify the following:

- Vulnerable (open-source) components
- Components with well-known functional flaws
- Outdated or redundant component versions
- Components with license restrictions

The xs CLI commands `find-app-artifacts`, `find-droplet-artifacts`, `find-runtime-artifacts`, and `find-buildpack-artifacts` are suitable for analyzing artifact dependencies of a single XS advanced controller resource such as an application, a droplet (a staged application), a deployed run time, or a build pack. When scanning a broader range of resources on the server, for example, a whole space or even an organization, it makes sense to start with the command `xs find-artifacts`, which provides the same basic features for analyzing and filtering artifacts, but operates on a larger set of resources. In their initial version, all artifact-reporting tools for XS advanced applications are limited to the detection of Java archives and Node.js modules.

> → Tip
>
> You can use operating-specific tools to `pipe` the report output generated by the `find-*-artifacts` commands to other commands such as `grep` or to a specific file. If the search completes successfully but nothing is found that matches the search query, the commands have process exit code "`0`". If the search query finds a match, "`-1`" is returned, and "`1`" if the commands finish unsuccessfully.

Listing and Updating Application Artifacts with XS Advanced 1.4.0 or Higher

With XS Advanced 1.4.0, xs find-artifacts was enhanced as follows:

- The output shows the name of the Multi Target Application (MTA) if applicable.
- The output shows the name of the SAP Software Component Version (SCV) if the artifact is part of an SAP product.
- Using the `-n` option, you can provide the full path name of an artifact (e.g. based on the output of a vulnerability scan), the full artifact name or parts of the artifact name.

This additional information gives direct hints what actions to take to update an outdated or vulnerable artifact:

Example 1

Use `xs find-artifacts` with a full path to an outdated or vulnerable artifact:

Sample Code

```
xs find-artifacts -n /hana/shared/SID/xs/app_working/myhost/executionroot/
03873ba2-fbc6-4ace-8fa5-fff3dd2d81c2/app/WEB-INF/lib/commons-lang3-3.14.0.jar
...
+ SCV "XSAC_XSA_COCKPIT" with version 1.1.47
  + MTA "com.sap.core.account" with version 1.1.47
    + DROPLET of application "xsa-cockpit" with index 1 in space "SAP" of org
"org" created at Aug 4, 2025 2:55:27 PM (1 RUNNING)
      + APP "xsa-cockpit" in space "SAP" of org "org" created at Aug 4, 2025
2:55:07 PM (STARTED)
          JAR commons-lang3 3.14.0
            path: /app/WEB-INF/lib/commons-lang3-3.14.0.jar
```

The output indicates, that the artifact `commons-lang3-3.14.0.jar` is contained in the droplet of the application `xsa-cockpit`, which is part of the MTA `com.sap.core.account`, which in turn is part of the SCV `XSAC_XSA_COCKPIT` . It also shows, that the artifact is part of the application (APP) itself. Hence, in order to update this artifact, you have to check for an updated version of the MTA or SCV and update it with `xs deploy` or `xs install`.

Example 2

Use `xs find-artifacts` to find artifacts that are not shipped by the application vendor, but that are added by the platform:

Sample Code

```
xs find-artifacts -n ecj-4.20.jar
...



                                    + SCV "XSAC_XSA_COCKPIT" with version
1.1.47
  + MTA "com.sap.core.account" with version 1.1.47
    + DROPLET of application "xsa-cockpit" with index 1 in space "SAP" of org
"org" created at Aug 4, 2025 2:55:27 PM (1 RUNNING)
      + RUNTIME "tomcat9" 9.0.107
          JAR ecj 4.20
            path: /app/META-INF/.sap_java_buildpack/tomcat/lib/ecj-4.20.jar
```

This output shows, that the artifact `ecj-4.20.jar` is contained the droplet of the application `xsa-cockpit` which is part of the MTA `com.sap.core.account` which in turn is part of the SCV `XSAC_XSA_COCKPIT` . However, it is not shipped by the application vendor. Instead, it's brought in by the application runtime (RUNTIME) `tomcat9` .

As a solution, you can check, if a more recent application runtime version is contained in the system with the command `xs runtimes` . In this case, you can restage and restart the application `xsa-cockpit` to make use of the newer runtime.

See also: Maintaining Application Runtimes

Example 3

The following example shows a dependency that's brought into an application droplet by the SAP Java buildpack version 2.0.5.1:

Sample Code

```
xs find-artifacts -n caffeine-2.9.3.jar
...
+ SCV "XSAC_XSA_COCKPIT" with version 1.1.47
  + MTA "com.sap.core.account" with version 1.1.47
    + DROPLET of application "xsa-cockpit" with index 1 in space "SAP" of org
"org" created at Aug 4, 2025 2:55:27 PM (1 RUNNING)
      + BUILDPACK "sap_java_buildpack" 2.0.5.1
          JAR caffeine 2.9.3
            path: /app/META-INF/.sap_java_buildpack/tomcat/lib/
caffeine-2.9.3.jar
```

In this case, check the currently installed buildpack version with `xs buildpacks`. If this application was staged by a previous version of the buildpack, restage the application with the latest version of the buildpack. Restart the application afterwards. If the buildpack version shows the same version as the output, you have to update XS Advanced to receive a new version of the buildpack.

Displaying the Usage of Application Modules

After deploying an application `my-web-app` to the XS advanced platform, you can find out which version of a particular Node.js module (for example, `express`) was uploaded to the server during application deployment. To scan the uploaded application's artifacts, run the command `xs find-app-artifacts` as a user with the privileges required to read application artifacts:

Note

The current xs target must be the space where the application is running.

Output Code

```
$ xs find-app-artifacts my-web-app -n express*

Finding artifacts of app "my-web-app" with name wild card "express*"...

+ APP "my-web-app" in space "SAP" of org "test" created at Jan 18, 2023
  12:33:03 PM (STARTED)
    NPM express 4.18.2
      path: /node_modules/express
    NPM express-session 1.17.3
      path: /node_modules/express-session

Found Artifacts   Affected Apps
------------------------------
2                 1
```

The report returned by the `xs find-app-artifacts` command shows that the deployed application `my-web-app` uses the modules `express` version 4.17.3 and `express-session` version 1.17.3, and that both modules are located in the `node-modules` folder.

→ Tip

The command `xs find-app-artifacts <APPNAME>` displays *all* components uploaded with the application.

You can use the following options with **all** find-artifact commands:

- --artifact-info
  Displays meta information about the artifacts, if available, for example, vendor, license, etc.
- --artifact-tree
  Displays the logical dependency tree of the components, if available. This helps understand which artifacts are referenced transitively by a direct dependency.

Showing Which Artifacts the Platform Adds During Application Staging

If you push a Spring-based Java application (for example, containing `.jar` files) to the XS advanced run-time platform, it is important to understand that, in addition to the Java archives required to implement the application logic, the Java build pack also adds a number of Java archives from the build pack content itself, as well as from several run times. You can use the command `xs find-droplet-artifacts` to display a report with details of all the artifacts in the target application's current droplet. The report displayed by the `xs find-droplet-artifacts` command also shows higher-level resources such as the application itself and the build pack and run times used, as illustrated in the following example:

{·} Output Code

```
$ xs find-droplet-artifacts my-spring-app

Finding artifacts of app "my-spring-app"...

+ DROPLET of application "my-spring-app" with index 3 in space "PROD" of
  org "test" created at Jan 19, 2018 1:47:39 PM (1 RUNNING, 1 STOPPED)
  + APP "my-spring-app" in space "PROD" of org "test" created at Sep 17, 2018
    3:11:07 PM (STARTED)
     JAR spring-security-core 4.2.3.RELEASE
       path: /app/WEB-INF/lib/spring-security-core-4.2.3.RELEASE.jar
     JAR spring-security-oauth2 2.3.3.RELEASE
       path: /app/WEB-INF/lib/spring-security-oauth2-2.3.3.RELEASE.jar
     JAR spring-security-web 4.2.3.RELEASE
       path: /app/WEB-INF/lib/spring-security-web-4.2.3.RELEASE.jar
     JAR spring-web 4.3.15.RELEASE
       path: /app/WEB-INF/lib/spring-web-4.3.15.RELEASE.jar
     [...]
  + BUILDPACK "sap_java_buildpack" 1.7.8
     JAR httpclient 4.5.3
      path: /app/META-INF/.sap_java_buildpack/tomcat/impl/httpclient-4.5.3...
     JAR httpcore 4.4.6
      path: /app/META-INF/.sap_java_buildpack/tomcat/impl/httpcore-4.4.6.jar
     JAR java-container-security 0.30.1
      path: /app/META-INF/.sap_java_buildpack/tomcat/impl/java-container-sec
     [...]
  + RUNTIME "hanajdbc2" 2.3.53
     JAR ngdbc 2.3.53
      path: /app/META-INF/.sap_java_buildpack/hana_jdbc/ngdbc-2.3.53.jar
  + RUNTIME "sapjvm8_jre" 8.1.43
     JAR rt 1.8.0_181
      path: /app/META-INF/.sap_java_buildpack/sapjvm/lib/rt.jar
     [...]
```

```
  + RUNTIME "tomcat8" 8.5.32
     JAR tomcat-api 8.5.32
      path: /app/META-INF/.sap_java_buildpack/tomcat/lib/tomcat-api.jar
     JAR tomcat-coyote 8.5.32
      path: /app/META-INF/.sap_java_buildpack/tomcat/lib/tomcat-coyote.jar
     JAR tomcat-websocket 8.5.32
      path: /app/META-INF/.sap_java_buildpack/tomcat/lib/tomcat-websocket.jar
     [...]

Found Artifacts   Affected Droplets
----------------------------------
226               1
```

The example above shows that the Spring-based application droplet contains the following artifacts:

- Application "`my-spring-app`"
- Build pack "`sap-java-buildpack`" (version 1.7.8)
- Run time "`hanajdbc2`" (version 2.3.53)
- Run time "`sapjvm8_jre`" (version 8.1.43)
- Run time "`tomcat8`" (version 8.5.32)

In total, the report found 226 artifacts matching the search criteria `my-spring-app`.

Listing the Artifact Versions Used in Productive Code

You can use the `xs` command-line tools to check if specific versions of an application component are still in use. This type of report is useful if a particular version of an application component is known to have a vulnerability or a proven functional flaw. For example, the `xs find-artifacts` command enables you to scan a large set of application resources on the server within a customizable search range: applications, droplets, run times, or build packs.

Since, generally speaking, the scope of the scan is determined by the privileges granted to the XS advanced platform user who runs the scan, it is recommended that application-artifact scans are run by a platform administrator user, in order to ensure that all resources available on the server are analyzed. You can limit the scope of a scan, for example, to search within a specific organization or space, by using the options "-o <ORG>" and "-s <SPACE>".

The following example shows how to run a search with the scope set to the Java archive `security-commons` with major version “0” and minor version “26”, which is outdated and needs to be replaced:

Output Code

```
$ xs find-artifacts -n security-commons* -v 0.26.*
Finding all artifacts of droplets with name wild card "security-commons*"
and version wild card "0.26.*"...
+ DROPLET of application "my-java-app" with index 1 in space "PROD"
  of org "test" created at Jan 20, 2018 9:02:42 AM (1 RUNNING, 1 STOPPED)
  + BUILDPACK "sap_java_buildpack" <outdated version, latest version:1.7.8>...
    JAR security-commons 0.26.13 (OUTDATED)
     path: /app/META-INF/.sap_java_buildpack/tomcat/impl/security-commons-0.26
Found Artifacts   Outdated Artifacts   Affected Droplets
-------------------------------------------------------------------------------
2                 2                   1
There are outdated artifact versions in search scope.
Restage and restart affected apps to replace them with current versions
```

```
available on the platform.
```

The command output in the example above shows that a droplet of a running instance of the application "`my-java-app`" is using an outdated version (`0.26.13`); the dependency was apparently introduced by "`sap_java_buildpack`". The first "`OUTDATED`" tag in the example above indicates that the build pack "`sap_java_buildpack`" on the platform has been an updated to version `1.7.8` after the application "`my-java-app`" was staged; restaging the application should update this dependency.

> **Note**
>
> After staging the `my-java-app` application with the current build pack, it is necessary to restart the application to activate the updates.

Related Information

The XS Command-Line Interface [page 1262]

14.2.2.2 Offline Maintenance and Diagnostics of XS Advanced with the XSA CLI

Diagnose and repair XS advanced services using command-line tools.

The XSA command is an offline diagnostic and repair tool. Use the XSA command if maintenance is required when XS advanced services like the controller (`xscontroller`), the execution agent (`xsexecutionagent`), and the `xsuaaserver` that provides services for User Accounts and Authentication (UAA) are offline.

With the XSA command, you can perform the following actions on an instance of the XS advanced run time:

- Enable
- Disable
- Restart
- Set a domain certificate
- Reset the default domain certificate for an XS advanced run-time instance
- Restage all applications that are running when the XS advanced run-time instance is restarted
- Backup and restore the Secure Store file system
- Delete user data from log files

> **Note**
>
> To log-in to XS the advanced platform,for example, to maintain run-time componentns, use the `xs` command-line client or the XS Advanced Cockpit, as described in *Related Information* below.

Related Information

The XSA Command Reference [page 1370]
Maintaining XS Advanced Run-Time Components with the XS CLI [page 1262]
Maintaining the XS Advanced Run-time Environment with SAP HANA XS Advanced Cockpit [page 1387]

14.2.2.2.1 The XSA Command Reference

Commands to help maintain XS advanced run-time instances, for example, enable, disable, or restart.

```
XSA {COMMAND} [--OPTIONS]
```

The following table lists the `XSA` commands that are most commonly used:

> **⚠ Restriction**
>
> To use the `XSA` command, you must log on as the operating-system user `<SID>adm`.

XSA Commands Overview

Command	Description
`help`	Display an overview of all command help parameters and options
`enable`	Enable a disabled XS advanced run-time instance
`disable`	Disable a running instance of the XS advanced run time
`restart`	Disable and re-enable an XS advanced run-time instance
`set-certificate`	Set the default server certificate after the XS Controller is shut down
`reset-certificate`	Reset the default server certificate and restart the XS advanced run-time instance
`trust-certificate`	Add a trusted certificate
`restage-at-startup`	Restage all applications that are running when the XS advanced instance (controller) is restarted
`backup-ssfs`	Store the current XS advanced Secure Store File System (SSFS) configuration in an encrypted database table
`recover-ssfs`	Restore the XS advanced Secure Store File System (SSFS) configuration from an encrypted database table
`diagnose`	Run tests that help diagnose problems with the XS advanced system
`backup-fss`	Create a backup copy of the file-system services in the database

Command	Description
`recover-fss`	Restore the file-system service instances from the database
`collect-traces`	Creates a zip containing the trace files of the XS advanced services
`delete-personal-data`	Erase user data from the log files
`du`	Provides the disk usage for application instances and file-system service
`grant-privileges-to-support-user`	Grants an SAP HANA user access to XS advanced database schemas
`list-tenants`	Lists all (tenant) databases registered for XS advanced → Tip You can also verify the credentials of the technical XS advanced users within tenants used by XS advanced.
`select-xsa-runtime-db`	If a database already contains an XS advanced installation, this command activates XS advanced for the specified tenant database
`unlock-technical-users`	Ensures the connectivity of all technical XSA database users
`renew-passwords-of-technical-users`	Set a new password for all technical users required by XS advanced system services.
`update-tenants`	Allow the SAP HANA broker to maintain databases regarding rights and roles needed for operation.
`renew-uaa-signing-keys`	Rotates the UAA signing keys. By default, if you don't specify additional options, it rotates all available UAA signing keys. You can use this command, e.g. after a system copy, to separate UAA signing keys from the source system.

enable

Enable a disabled XS advanced run-time instance.

Usage

```
XSA enable [--localhost] [--verbose] [--async]
```

Options

Command Options Overview

Option	Description
`--localhost`	Restrict the operation to the local host
`--verbose`	Display all available information during the selected operation
`--async`	Return to the command prompt without waiting for the command to complete

disable

Disable a running instance of the XS advanced run time.

Usage

```
XSA disable [--localhost] [--verbose] [--async]
```

Options

Command Options Overview

Option	Description
`--localhost`	Restrict the operation to the local host
`--verbose`	Display all available information during the selected operation
`--async`	Return to the command prompt without waiting for the command to complete

restart

Disable and re-enable an XS advanced run-time instance.

Usage

```
XSA restart [--localhost] [--verbose] [--async]
```

Options

Command Options Overview

Option	Description
`--localhost`	Restrict the operation to the local host

Option	Description
`--verbose`	Display all available information during the selected operation
`--async`	Return to the command prompt without waiting for the command to complete

set-certificate

Set the default server certificate of an XS advanced run-time instance.

> ⓘ Note
>
> The `set-certificate` command enables you to set a valid domain certificate without the need to start the XS advanced controller first.

By default, all applications and the XS advanced controller are reached by means of the default domain that is configured during the installation operation. To enable HTTPS communication, it is necessary to configure an SSL certificate for the default domain or, alternatively, use the self-signed certificate provided by XS advanced. If the certificate provided for the default domain expires, neither the applications nor the XS advanced controller can be reached. In addition, the XS Controller will not be reachable after a restart because the initial availability checks will fail. The `set-certificate` command enables you to resolve the problem by setting the default server certificate.

Usage

```
XSA set-certificate [--cert <CERTIFICATE_FILE>] [--key <KEY_FILE>] [--verbose]
[--async]
```

Options

Command Options Overview

Option	Description
`--cert <CERTIFICATE_FILE>`	Specify the file containing the public certificate to use.
`--key <KEY_FILE>`	Specify the file containing the private key to use.
`--verbose`	Display all available information during the selected operation.
`--async`	Return to the command prompt without waiting for the command to complete.
`-n, --no-restart`	Do not restart XS-advanced services after setting the certificate.
`-p, --pse <PSE_FILE>`	Specify the path to the PSE file containing the RSA key and the X.509 certificate chain.

Option	Description
`--pse-pin <PIN>`	Specify the PIN for the PSE file. If no PIN is provided, the user is prompted to supply it interactively.
`--stdin`	Retrieve the PSE PIN from `stdin` pipe.

reset-certificate

Reset the default server certificate of an XS advanced run-time instance and restart the XS advanced run-time instance.

> → Tip
>
> The `reset-certificate` command can only be used to generate a new **self-signed certificate**, which is helpful in cases when you just want to restart the XS advanced controller after a certificate becomes outdated.

By default, all applications and the XS advanced controller are reached by means of the default domain that is configured during the installation operation. To enable HTTPS communication, it is necessary to configure an SSL certificate for the default domain or, alternatively, use the self-signed certificate provided by XS advanced. If the certificate provided for the default domain expires, neither the applications nor the XS advanced controller can be reached. In addition, the XS Controller will not be reachable after a restart because the initial availability checks will fail. The `reset-certificate` command enables you to reset the certificate for the default domain to resolve the problem.

Usage

```
XSA reset-certificate [--verbose] [--async]
```

Options

Command Options Overview

Option	Description
`--verbose`	Display all available information during the selected operation
`--async`	Return to the command prompt without waiting for the command to complete

trust-certificate

Adds a trusted certificate, even if the XS advanced controller is not running.

Usage

```
XSA trust-certificate [--alias <ALIAS>] [--cert <CERTIFICATE_FILE>] [--hana-
broker] [--client-auth] [--verbose]
```

Options

Command Options Overview

Option	Description
`-a, --alias <ALIAS>`	The alias of the trusted certificate
`-c, --cert <CERTIFICATE_FILE>`	The path to the X.509 certificate file in PEM format.
`--client-auth`	Use the certificate for validation of certificate-based client authentication
`--hana-broker`	Add the certificate to the service bindings issued by the SAP HANA service broker.
`--verbose`	Display all available information during the selected operation

restage-at-startup

On restart of the XS advanced instance (controller), restage all applications that were running in the XS advanced instance before the restart. Restaging applications ensures that they have access to any security fixes for deployment build packs and run-time environments that are available.

Usage

```
XSA restage-at-startup [--verbose]
```

> → Tip
>
> The XS advanced instance is not restarted automatically; you must restart the XS advanced instance yourself manually.

Options

Command Options Overview

Option	Description
`--verbose`	Display all available information during the selected operation

backup-ssfs

Store the current XS advanced Secure Store File System (SSFS) configuration in an encrypted database table.

Usage

```
XSA backup-ssfs [--verbose]
```

> **Note**
>
> The `backup-ssfs` command replaces the command `save-ssfs-to-dbss`.

Options

Command Options Overview

Option	Description
`--verbose`	Display all available information during the selected operation

recover-ssfs

Restore the XS advanced Secure Store File System (SSFS) configuration from an encrypted database table.

Usage

```
XSA recover-ssfs [-u <system_db_user>] [-p <system_db_password>] [-n
<db_host_port>] [--verbose]
```

> **Note**
>
> The `recover-ssfs` command replaces the command `restore-dbss-to-ssfs`.

Options

Command Options Overview

Option	Description
`-u <system_db_user>`	The name of the system database user
`-p <system_db_password>`	The password of the system database user. **Note** If you do not provide the password with the command, you are prompted to provide it interactively.

Option	Description
`-n, --database-name <database_name>`	The database name to use to establish a connection to the tenant database where XS advanced keeps its persistence, for example, in a multi-tenant database container (MDC) system that was migrated from a single database container system. **Note** This information is not required if XS advanced keeps its persistence in the System database (default).
`--verbose`	Display all available information during the selected operation

diagnose

Run tests that help diagnose problems with the XS advanced system. The command writes its findings to `stdout` as well as to a dedicated trace file. The path to the location of the trace file is added to the end of the output written to `stdout`.

Usage

```
XSA diagnose [--verbose]
```

backup-fss

Creates a backup of the file-system services (FSS) in the database. The command backs up all existing FSS instances, unless the `--services` option is specified. If an FSS instance is deleted, then the corresponding backup is also removed from the database.

Usage

```
XSA backup-fss [--services <service>] [--list-services] [--verbose]
```

Options

Command Options Overview

Option	Description
`-s, --services <services>`	Specify a single GUID or a list of GUIDs of file-system service instances to back up. Run the command `xs services` with the option `--guids` to retrieve the GUID for a particular service instance.

Option	Description
`-l, --list-services`	Displays a list of the existing backups
`--verbose`	Display all available information during the selected operation

recover-fss

Restores the file-system service (FSS) instances from the database. The command recovers all FSS instances with existing backups unless the `--services` option is specified.

Usage

```
XSA recover-fss [--services <service>] [--list-services] [--verbose]
```

Command Options Overview

Option	Description
`-s, --services <services>`	Specify a single GUID or a list of GUIDs of file-system-service (FSS) instances to recover Run the command `xs services` with the option `--guids` to retrieve the GUID for a particular service instance.
`-l, --list-services`	Display a list of the existing backups
`-f,--force`	Force execution of the recover command without requiring any user confirmation
`--verbose`	Display all available information during the selected operation

delete-personal-data

Remove user data from the log files generated in an XS advanced run-time instance. The `delete-personal-data` command triggers log-rotation for all log files and removes all user-related data (for example, user names, IP addresses) from the log files up to a specified point in time. You can use the `delete-personal-data` command to clean up the following types of XS advanced log files:

- XS advanced Controller log files (`controller_#.log`)
- XS advanced Execution Agent log files (`ea_#.log`)
- HTTP application-access files (`webdispatcher`)
- Application trace files (`myApp.trace`)
- User Account and Authentication (UAA) service artifacts

Usage

```
XSA delete-personal-data [--until <DATE>] [--force] [--exclude <EXCLUDES>] [--
user <NAME>] [--verbose]
```

Options

Command Options Overview

Option	Description
`-u, --until <DATE>`	The date that specifies the last day of the period of time for which you want to delete log-file entries that contain user-related information. → Tip `<DATE>` must be specified in the format YYYY-MM-DD, for example, 2017-06-30.
`-f, --force`	Force execution of the data-deletion command without requiring any user confirmation
`-e, --exclude <EXCLUDES>`	Exclude the specified log types from the deletion operation. Multiple log types are specified in a comma-separated list consisting of the following values: `[PLATFORM,AUDIT,APP,ACCESS,USERS]`
`--user <NAME>`	The name of the XS advanced **administrator** user in whose account the data-deletion command will run
`--verbose`	Display all available information during the selected operation

collect-traces

Creates a Zip file containing the trace files written by the XS advanced services; the created zip file overwrites any existing Zip file previously created by the `collect-traces` command.

Usage

```
XSA collect-traces --output <xsa_traces.zip>
```

Options

Command Options Overview

Option	Description
`-o, --output <xsa_traces.zip>`	The path to the Zip file containing the collected trace files
`--include-audit-logs`	Include audit logs which are not collected by default.

grant-privileges-to-support-user

Grants an SAP HANA user access to XS advanced database schemas. If the target user does not already exist, a new user is created with the specified name.

> → Tip
>
> Using this command does not require the privileges of the `SYSTEM`.

A lesser-privileged user can be created and used to call the `grant-privileges-to-support-user` command; this user must have the following privileges:

- `SYSTEM` privilege `USER ADMIN`
- `EXECUTE` privileges on the following procedures:
  - `SYS_XS_SUPPORT.XS_SUPPORT_GRANT_SELECT`
  - `SYS_XS_SUPPORT.XS_SUPPORT_GRANT_ALL`
  - `SYS_XS_SUPPORT.XS_SUPPORT_REVOKE_ALL`

Usage

```
XSA grant-privileges-to-support-user --support-user <SUPPORT_USER_NAME>
[--system-user <SYSTEM_USER_NAME>] [--system-password <SYSTEM_USER_PASSWORD>]
[--support-password <SUPPORT_USER_PASSWORD>] [--grant-all]
```

Options

Command Options Overview

Option	Description
`--system-user <SYSTEM_USER_NAME>`	The name of the `SYSTEM` user of the XS advanced run-time database
`--system-password <SYSTEM_USER_PASSWORD>`	The password of the `SYSTEM` user of the XS advanced run time database. ⚠ Caution For security reasons, it is not recommended to use the --password option to specify a password for the `SYSTEM` user. If no password is specified, the system interactively requests one.
`-u, --support-user <SUPPORT_USER_NAME>`	The name of the support user to whom support privileges should be granted. ⓘ Note If the target user does not already exist, a new user is created with the specified name.

Option	Description
`-p, --support-password <SUPPORT_USER_PASSWORD>`	The password of the support user to whom access privileges are to be granted. **Note** If no password is specified, the system interactively requests one, and creates a new user with the specified name, if the target user does not already exist.
`-a, --grant-all`	Grant **all** access privileges, not only SELECT privileges

du

Provides the disk usage for application instances and file-system service.

Usage

```
XSA du [--app] [--fss] [--orderby <ORDERBY>] [--orderdir <ORDERDIR>]
```

Options

Command Options Overview

Option	Description
`-a, --app`	Display the disk usage for application instances running on target XS advanced hosts
`-f, --fss`	Display the disk usage for the file-system service **Note** Requires SAP HANA to be started.
`--orderby <ORDERBY>`	Sort by specific columns for application disk usage **Tip** The option accepts comma-separated values.
`--orderdir <ORDERDIR>`	Specify the sort order direction for application disk usage, for example, asc(ending) or desc(ending)

list-tenants

Display details of all (tenant) databases registered for use with XS advanced model. It is also possible to check the validity of the user credentials that XS advanced uses to connect to each registered database.

Usage

```
list-tenants [--user <XSA admin user>] [--password <XSA admin password>] [--
check-credentials]
```

Options

Command Options Overview

Option	Description
`-u, --user <XSA admin user>`	The name of an XS advanced administration user (for example, `XSA_ADMIN`)
`-p, --password <XSA admin password>`	The password for the XS advanced administration user specified with the --user option
`--check-credentials`	Verify the credentials of the XS advanced technical users that XS advanced uses to connect to each registered database

select-xsa-runtime-db

If a database already contains an XS advanced installation, this command activates XS advanced for the specified tenant database. As of XS advanced 1.0.116 this command makes a backup of the file-system service instances, stores it in the currently active XS advanced database and restores the file-system services stored in the target database. It establishes database trust between UAA and the tenant databases, that are registered in the target XS advanced system.

Usage

```
select-xsa-runtime-db --tenant-db-name <tenant_db_name>
[--tenant-db-system-user <tenant_db_system_user>] [--tenant-db-system-user-
password <tenant_db_system_user_password>]
[--system-db-user <system_db_user>] [--system-db-password <system_db_password>]
[--user <XSA admin user>] [--password <XSA admin password>] [--skip-fss]
```

Options

Command Options Overview

Option	Description
`-n, --tenant-db-name <tenant_db_name>`	The name of the tenant database
`-t, --tenant-db-system-user <tenant_db_system_user>`	The name of the tenant database's `SYSTEM` user

Option	Description
`-p, --tenant-db-system-user-password <tenant_db_system_user_password>`	The password of the tenant database's `SYSTEM` user **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `SYSTEM` user. If no password is specified, the system interactively requests one.
`-c, --system-db-password <system_db_password>`	The password of the SYSTEM database user **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `SYSTEM` user. If no password is specified, the system interactively requests one.
`-s, --system-db-user <system_db_user>`	The name of the `SYSTEM` database user
`-u, --user <XSA admin user>`	The name of the XS advanced administrator (`XSA_ADMIN`) user
`-x, --password <XSA admin password>`	The password of the XS advanced administrator (`XSA_ADMIN`) user **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `XSA_ADMIN` user. If no password is specified, the system interactively requests one.
`-s , skip-fss`	Skip backup and restore of the file system service instances. (Available with XSA > 1.0.115)

unlock-technical-users

Ensures the connectivity of all technical XSA database users.

Usage

```
unlock-technical-users [--user <XSA admin user>] [--password <XSA admin password>]
[--system-db-system-user <system_db_user>] [--system-db-system-user-password <param>]
[--xsa-db-system-user <system_db_user>] [--xsa-db-system-user-password <param>]
```

Options

Command Options Overview

Option	Description
`-u, --user <XSA admin user>`	The name of the XS advanced administrator (`XSA_ADMIN`) user
`-p, --password <XSA admin password>`	The password of the XS advanced administrator (`XSA_ADMIN`) user **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `XSA_ADMIN` user. If no password is specified, the system interactively requests one.
`-s, --system-db-system-user <system_db_user>`	The name of the `SYSTEM` user of the system database
`-c, --system-db-system-user-password <param>`	The password for the `SYSTEM` user of the system database **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `SYSTEM` user. If no password is specified, the system interactively requests one.
`-t, --xsa-db-system-user <xsa_db_system_user>`	The name of the `SYSTEM` user of the database where XS advanced is installed
`-v, --xsa-db-system-user-password <param>`	The password of the `SYSTEM` user of the database where XS advanced is installed **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `SYSTEM` user. If no password is specified, the system interactively requests one.

renew-passwords-of-technical-users

Sets a new password for all technical users required by XS advanced system services.

Usage

```
renew-passwords-of-technical-users [--user <XSA admin user>] [--password <XSA admin password>]
[--system-db-system-user <system_db_user>] [--system-db-system-user-password <param>]
[--xsa-db-system-user <system_db_user>] [--xsa-db-system-user-password <param>]
```

Options

Command Options Overview

Option	Description
`-u, --user <XSA admin user>`	The name of the XS advanced administrator (`XSA_ADMIN`) user
`-p, --password <XSA admin password>`	The password of the XS advanced administrator (`XSA_ADMIN`) user **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `XSA_ADMIN` user. If no password is specified, the system interactively requests one.
`-s, --system-db-system-user <system_db_user>`	The name of the `SYSTEM` user of the system database
`-c, --system-db-system-user-password <param>`	The password for the `SYSTEM` user of the system database **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `SYSTEM` user. If no password is specified, the system interactively requests one.
`-t, --xsa-db-system-user <xsa_db_system_user>`	The name of the `SYSTEM` user of the database where XS advanced is installed
`-v, --xsa-db-system-user-password <param>`	The password of the `SYSTEM` user of the database where XS advanced is installed **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the user. If no password is specified, the system interactively requests one.

update-tenants

Allows XS advanced to grant new permissions, coming with a new SAP HANA version, to database artifacts used by XS advanced applications.

Usage

```
update-tenants [--user <XSA admin user>] [--password <XSA admin password>]
[--databases <list_of_databases>] [-rebind-hana-broker-services ]
```

Options

Command Options Overview

Option	Description
`-u, --user <XSA admin user>`	The name of the XS advanced administrator (`XSA_ADMIN`) user
`-p, --password <XSA admin password>`	The password of the XS advanced administrator (`XSA_ADMIN`) user **⚠ Caution** For security reasons, it is not recommended to use the --password option to specify a password for the `XSA_ADMIN` user. If no password is specified, the system interactively requests one.
`-d, --databases <list_of databases_to_update>`	Names of databases to update. By default, all databases
`-r, --rebind-hana-broker-services`	Rebind all SAP HANA broker service bindings, mapped to processed databases, to update required HDI container permissions.

renew-uaa-signing-keys

Rotates the UAA signing keys. By default, if you don't specify additional options, it rotates all available UAA signing keys. You can use this command, e.g. after a system copy, to separate UAA signing keys from the source system. This command restarts XS Advanced.

ⓘ Note

The command is available as of XS Advanced 1.4.1

Usage

```
renew-uaa-signing-keys [--jwt] [--saml] [--force]
```

Options

Command Options Overview

Option	Description
`--jwt`	Renew the JWT signing keys
`--saml`	Renew the SAML signing keys
--force	Runs the command non-interactively

Related Information

Offline Maintenance and Diagnostics of XS Advanced with the XSA CLI [page 1369]

14.2.3 Maintaining the XS Advanced Run-time Environment with a Graphical User Interface

Use a graphical user interface to administrate and maintain XS advanced-model run-time components.

SAP HANA XS advanced model includes a Web-browser-based administration tool called XS Advanced Cockpit with a graphical user interface that enables you to maintain important elements of the XS advanced application-development environment, for example, security and authentication methods.

The XS Advanced Cockpit includes a selection of tools and features which enable you to configure and maintain the basic administration-related elements of the application-development process for the XS advanced run-time environment.

Related Information

Maintaining the XS Advanced Run-time Environment with SAP HANA XS Advanced Cockpit [page 1387]
Maintaining the XS Advanced Run-time Environment with a Command-Line Interface [page 1261]
Scheduling Jobs in XS Advanced [page 1439]

14.2.3.1 Maintaining the XS Advanced Run-time Environment with SAP HANA XS Advanced Cockpit

Configure the graphical user interface that enables you to configure and maintain the basic administration-related elements of the XS advanced run-time environment.

The SAP HANA XS advanced cockpit provides a graphical user interface that enables you to configure and maintain the basic administration-related elements of the application development process for the XS advanced run-time environment. To start the XS advanced cockpit, you can use the following URL, which depends on whether XS advanced is running in hostname- or port-based routing mode:

- Host-based routing
  `https://xsa-cockpit.<default-domain>:3<instance number>33`
- Port-based routing
  `https://xsa-cockpit.<default-domain>:<router_port>`
  Port-based routing for XS advanced is configured with the `router_port` parameter in the `xscontroller.ini` file.

→ Tip

The `xsa-cockpit` service listed in the output returned by the command `xs -version` provides a URL that you can use to start the XS advanced cockpit. This URL is also displayed when you run the `xs apps` command in the XS advanced space where the XS advanced cockpit is running.

Alternatively, type the following URL in a Web browser: `https://<xs-controller-api-url>/go/xsa-cockpit`. The URL for the XS controller API is displayed in the information returned by the `xs system-info` command. See *Related Information* below.

The XS advanced cockpit's navigation pane displays the standard run-time options. Choose *More...* in the navigation pane to display a list of additional run-time options, for example, any plug-ins that have been configured for use. Your role in an XS advanced organization determines which options are made available to help you perform the tasks specific to your role.

The details pane displays the available organizations. You can be a member of one or more organizations. To work with applications and services, you need to navigate to spaces within an organization. For more information about working with organizations and spaces, see *Maintaining Organizations and Spaces in XS Advanced* in *Related Information* below.

The following table provides a brief overview and scope of the different run-time options:

Runtime Options	Description	Scope
System Information (Home)	Display information about the various component areas of XS advanced.	XS advanced: • System summary (incl. SID and controller API end point) • XS controller details • Running execution agents • Registered service URLs • Summary of spaces
Organizations (and Spaces)	Create, list, or delete organizations and spaces in the XS advanced model run time.	• Organizations and spaces • Users in organizations and spaces • XS advanced business-user roles for organizations and spaces • Applications and services within a space
Security	Manage SAML identity providers for use by XS advanced applications that need to authenticate the XS advanced business users signing in by means of SSO. Set, manage, and maintain the certificates used to establish secure and trusted connections between SAP HANA systems and SAP HANA XS advanced applications.	• XS advanced roles • SAML identity providers (including IDP metadata, certificates, and destinations) • XS advanced trust certificates

Runtime Options	Description	Scope
Application Runtimes	Manage and maintain the application runtimes, for example, Node.js, Java (Tomcat, TomEE), SapMachine, etc.	• Version • State (Enabled/Disabled) • Maintenance (Outdated/In Use) • Number of bound apps • Description
Tenant Databases	Manage, maintain, and configure SAP HANA tenant databases for use with SAP HANA XS advanced model applications.	• Tenant databases in XS advanced • Creating tenant databases in XS advanced • Preparing tenant databases
Host Management	Manage the SAP HANA hosts that are pinned to SAP HANA XS advanced applications or spaces.	• SAP HANA hosts that are pinned to XS advanced applications or spaces
User Management	Maintain and manage database users for SAP HANA XS advanced	• XS advanced platform users
Application Monitoring	Monitor the system resources used by applications running in the XS advanced model run-time environment.	• XS advanced applications' use of system resources
Application Management	Perform various actions on an application, such as scaling an application, stopping or restarting an application, establishing secure access, verifying application logs or events for troubleshooting.	• Scaling application instances up or down • Managing application security • Troubleshooting an application

Related Information

Maintaining Organizations and Spaces in XS Advanced [page 1393]
Monitoring XS Advanced Application Runtimes [page 1400]
Managing XS Advanced Applications [page 1404]
Displaying the XS Advanced System Overview (xs system-info CLI)

14.2.3.1.1 Managing Users in XS Advanced

The *User Management* option in the XS Advanced Cockpit enables you to create a new user or add a user from SAP HANA so that users have the required authentication and authorization to work with the various runtime options included with the XS Advanced platform. You can perform the following tasks:

- Create a new user
- Modify user details
- Search for a user
- Delete a user

14.2.3.1.1.1 Create Users in XS Advanced

Context

This option enables you to create users or to promote existing SAP HANA users to access organizations and spaces in the SAP HANA XS Advanced runtime (including XS Advanced Cockpit).

Procedure

1. In the home navigation pane, choose *User Management*.
2. In the *User Management* screen, choose one of the following options:

Action	Description
New User	Create a new user
Migrate SAP HANA User:	Promote an existing SAP HANA user to an XS advanced runtime user. You can either provide the SAP HANA user ID or select the user from the user list.

Related Information

Manage Users [page 1391]
User Details [page 1392]

14.2.3.1.1.2 Manage Users

Context

You can perform multiple actions to maintain user details, such as update user details, reset password, assign role collection, search, or delete users from the XS Advanced Cockpit.

Procedure

1. In the home navigation pane, choose *User Management*.
2. In the *User Management* screen, perform any of the following tasks:

Task	Option	Description
Update user details	*(Edit User)*	You can modify user details such as first name, last name, or e-mail address for the selected user. This option is available underneath the *Actions* column.
Reset password of a user	*(Change Password)*	You can change the password required to access the XS Advanced runtime including the XS Advanced Cockpit. This option is available underneath the *Actions* column.
Assign role collections to a user	*(Assign Role Collection)*	As well as the default role collection that is available for all users, you can add additional role collections and assign them to specific users. This option is available underneath the *Actions* column.
Delete a user	*(Delete User)*	This option deletes a user. This option is available underneath the *Actions* column.

Task	Option	Description
Search for a specific user	Search	If your member management screen contains a long list of users, you can use the search option to list a specific user. You can search for users using any of the following user details: member ID, user name, or role collection.

Related Information

Assign Roles to Role Collection [page 1423]
Control Access to Applications [page 1409]
User Details [page 1392]
Create Users in XS Advanced [page 1390]

14.2.3.1.1.2.1 User Details

You can view the details of an XS Advanced platform user with the XS Advanced Cockpit. These are the details that are provided in the *New User* dialog.

User Interface Element	Description
User ID	Unique ID of the XS Advanced user. This is a mandatory field.
First Name	First name of the user.
Last Name	Last name of the user.
E-Mail	E-mail address of the user to whom the account belongs. The e-mail address must be unique to the user. This is a mandatory field.
Password	Password for the XS Advanced user. This is a mandatory field.

14.2.3.1.2 Maintaining Organizations and Spaces in XS Advanced

Organization is a logical entity to group spaces. By default, an organization does contain a space. You can create one or more spaces to enable developers to collaborate by sharing resource, services, and applications. Access to the shared resources, services, and applications is controlled by roles, for example, Organization Manager or Organization Auditor. The role defines the scope of permissions available for a user in an organization. For example, an Organization Manager can add new users to organizations, create, modify, or delete organizational spaces, and add domains to the organization.

A space enables users to develop and maintain applications. Each space provides shared access to users of the space for application development, deployment, and maintenance. Access to the resources is controlled by roles, for example, Space Manager, Space Developer, or Space Auditor. The role defines the scope of permissions available for a user in a space. For example, a Space Developer can deploy and start an application.

You can perform the following tasks within an organization and space:

- Create an organization
- Manage an organization
- Maintain users in an organization
- Create a space
- Manage a space
- Maintain users in a space

Related Information

Maintaining Organizations and Spaces in XS Advanced [page 1273]

14.2.3.1.2.1 Create an Organization

Context

You create an organization to establish a collaborative environment for sharing resources, services, and applications.

Procedure

1. In the *Home* navigation pane, choose *Organizations*.
2. On the *Organizations* screen, choose *New Organization*.
3. Provide the following details:

User Interface Element	Description
Name	Provide a name for the organization. This field is mandatory.
Assign Roles to User	This option provides the self-service option of assigning one or both of the roles (Organization Manager or Organization Auditor). You need to have these roles to perform organizational tasks.

14.2.3.1.2.2 Manage an Organization

Context

You can update the details or delete organizations.

Procedure

1. In the *Home* navigation pane, choose *Organizations*.
2. On the *Organizations* screen, perform any of the following tasks:

Tasks	Choose	Description
Update organizational details	*(Edit)*	You can update details such as the name of an organization.
Delete an organization	*(Delete)*	You can remove an organization.

Tasks	Choose	Description
Search for an organization	Search	If you want to view only a specific organization from the list of organizations, enter the organization name in the search field.

Related Information

Create an Organization [page 1393]
Maintain Users in an Organization [page 1395]
Maintain Users in a Space [page 1399]

14.2.3.1.2.3 Maintain Users in an Organization

Add and configure users in an organization.

Context

To perform tasks within an organization, you need to add users to specific roles within the organization. Roles provide controlled access to users within an organization.

Procedure

1. In the *Home* navigation pane, choose *Organizations*.
2. In the right pane, choose an organization tile.
3. In the members navigation pane, choose *Members*.
4. On the *Members* page, choose *Add Members*.
5. Enter one or more user IDs.
6. Assign one or both of the roles (*Organization Manager* or *Organization Auditor*)

Related Information

Maintain Users in a Space [page 1399]
Maintaining Platform Users in XS Advanced [page 1281]

14.2.3.1.2.3.1 Organizational Roles

You can grant or restrict access to organizations by assigning roles. The following table lists the roles that you can assign to XS Advanced users in an organization:

Role	Description
Organization Manager	An Organization Manager can perform the following tasks: • Create and manage organization users • Create, modify, or delete organizational spaces • Add domains to the organization
Organization Auditor	An Organization Auditor can perform the following tasks: • View all users in the organization • View the roles assigned to a user (or users) in the organization • View spaces within an organization

14.2.3.1.2.4 Create a Space

Add a space to an organization.

Prerequisites

You are already a member of an organization and have the Organization Manager role.

Context

You create a space to enable users to collaborate when developing and maintaining applications.

Procedure

1. In the *Home* navigation pane, choose *Organizations*.
2. On the *Organizations* screen, select an organization.
3. On the *Spaces* screen, choose *New Space*.

4. Provide the following details:

User Interface Element	Description
Space Name	Provide a name for the space. This field is mandatory.
Assign space roles	This is a self-service option for assigning one or all of the roles (Manager, Developer, or Auditor). You need to have these roles to perform space-related tasks.

14.2.3.1.2.5 Manage a Space

Maintain the spaces in an organization.

Prerequisites

You have already selected an organization.

Context

A space contains various deployed applications and resources such as services and members. You can perform various tasks in a space, such as stop, start, delete, or search for an application. You can also do the following:

- Bind a service to an application deployed to the space
- Assign (pin) one or more hosts (SAP HANA systems) to a space

Procedure

1. In the *Home* navigation pane, choose *Organizations*.
2. On the *Organizations* screen, choose *Spaces*.
3. On the *Spaces* screen, choose a space (for example, *SAP*), where you can perform any of the following tasks:

Task	Choose	Description
Start a stopped application	▷ *(Start)*	You can use this option to start an application from its stopped state.

Task	Choose	Description
Stop a running application	*(Stop)*	You can use this option to stop an application that is currently running.
Delete an application	*(Delete)*	You can use this option to remove an application from a space.
Search for an application	Search	If a space contains a long list of deployed applications, you can use the search option to list a specific application. Search for an application using the application name.
Manage an application	Application name	You can use this option to perform multiple tasks within an application, such as define security, check application logs or events, and so on.
Assign one or more hosts to a space	Pinned Hosts	You can use this option to select one or more hosts from the displayed list. The applications in the space will be deployed to the selected hosts. You can define one of the following modes: • Strict: In this mode, an application cannot start if the pinned host is not available. • Relaxed: In this mode, an application can start on another host if the pinned host is not available. **Note** You can switch from one mode to the other.
Add users to a space	Members	You can use this option to grant permissions to users to perform specific tasks within the space. For more information, see Maintain Users in a Space [page 1399].
Use a service within the space	Service Marketplace / Service Instances / User-Provided Services	You can use this option to use the services available within the space with any applications deployed to the space.

Related Information

Managing XS Advanced Applications [page 1404]
Managing Services in XS Advanced [page 1434]

14.2.3.1.2.6 Maintain Users in a Space

Add and configure users in a space.

Prerequisites

- You are already a member of the organization that contains the space.
- You have the `Space Manager` role in the space.

Context

To enable users to perform tasks within a space, you need to add the users to the space within the organization. You can use roles to provide and control access for users, for example, a `SpaceDeveloper`, within a space.

Procedure

1. In the *Home* navigation pane, choose *Organizations*.
2. On the *Organizations* screen, choose *Spaces*.
3. On the *Spaces* screen, select a space.
4. In the navigation pane, choose *Members*.
5. On the members page, choose *Add Members*.
6. Enter one or more user IDs.
7. Assign one or all of the roles (for example, `Space Manager`, `Space Developer`, or `Space Auditor`).

Related Information

Maintain Users in an Organization [page 1395]
Maintaining Platform Users in XS Advanced [page 1281]

14.2.3.1.2.6.1 Space Roles

You can grant or restrict access to spaces by assigning roles. The following table lists the roles that you can assign to XS Advanced users in a space:

Role	Description
Space Manager	A Space Manager can perform the following tasks: • Manage users in the selected space • View details of applications running in the space (for example: status, instances, service bindings, and resource usage)
Space Developer	A Space Developer can perform the following tasks: • Deploy, start, stop an application • Bind an application to (or unbind an application from) a service • View details of applications running in the space (for example: status, instances, service bindings, and resource usage)
Space Auditor	A Space Auditor can perform the following tasks: • View details of applications running in the space (for example: status, instances, service bindings, and resource usage)

14.2.3.1.3 Monitoring XS Advanced Application Runtimes

View, manage, and maintain XS advanced application runtime instances.

You can use the *Application Runtimes* tool included in the XS Advanced Cockpit to provide an overview of the currently enabled application runtimes, for example, `hanajdbc2`, `node18`, `sapjvm8` or `sapmachine11`. On the *Application Runtimes* screen, you can perform perform the following tasks:

- Enable or disable a runtime instance
- Delete one or more outdated versions of an application runtime
- Display details of an individual application runtime, including the version and a short description as well as its current status (enabled/disabled)
- See how many applications are bound to an application runtime version.

> → Tip
>
> To start the *Application Runtimes* tool, choose *Home* > *Application Runtimes* .

Related Information

Maintain Application Runtimes in XS Advanced [page 1401]

14.2.3.1.3.1 Maintain Application Runtimes in XS Advanced

View and manage the application runtimes installed in your XS advanced instance.

Context

The XS advanced cockpit includes a tool named *Application Runtimes* that enables you to display a list of the application runtimes currenty installed in your XS advanced instance, for example, Node.js, SAP JVM, Tomcat, TomEE, SapMachine, etc. The information provided includes the runtime version, the current state (enabled or disabled), whether or not the runtime is outdated (for example, superceded by another, newer version), and the number of applications that are currently using (bound to) the listed runtime.

> → Tip
>
> Use this tool to enable or disable a runtime instance and clean up any (or all) outdated runtimes.

Procedure

1. Display a list of all the application runtimes in your XS advanced instance.

   Choose *Application Runtimes* in the home screen of the XS Advanced Cockpit.
2. Display details of an individual application runtime.
   a. Select a runtime in the list of application runtimes.
   b. Choose > to display details of the selected application runtime's properties.
3. Enable or disable an application runtime.
   a. Select an application runtime in the list of runtimes.
   b. Choose *On* (or *Off)* in the list of actions.This includes information about the runtime type, version, and ID as well as whether the selected runtime is currently enabled, out of maintenance, or resolved (correctly detected by the XS advanced runtime environment).

      The information in the *State* column reflects the result of the chosen action, for example, *Enabled* (On).
4. Remove outdated application runtimes.

   To delete an individual outdated runtime, perform the following steps:

   > ⚠ Caution
   >
   > The *Delete Outdated Runtimes* tool enables you to delete **all** application runtimes that XS advanced has flagged as "outdated", which means out of maintenance or replaced by a newer runtime version. Before

deleting an application runtime, however, make sure you unbind any applications that are currently bound to it, as described below.

a. Display a list of all available application runtimes.

   In the *Home* page of the XS Advanced Cockpit, choose *Application Runtimes*.

b. Locate the runtime you want to delete.

c. Check if any applications are bound to the runtime that you want to delete.

   > ⚠ Caution
   >
   > The value displayed in the *Bound Apps* column indicates how many applications are bound to an application. XS Advanced cockpit displays a warning if you attempt to delete a runtime that is bound to an application. Deleting the runtime anyway could lead to inconsistencies and further problems.

   If no applications are bound to the runtime you want to remove, choose 🗑 *(Delete runtime)* in the *Actions* column.

   If any applications are bound to the runtime you want to delete, you need to restage and start each bound application and then clear all application instances, as described in the following step.

d. Unbind any applications that are bound to the runtime you want to delete.

   In the list of application runtimes, select the runtime you want to delete and perform the following steps

   1. Display details of the runtime.
      Either select a runtime in the list or locate the runtime and choose › .
      The *Bound Applications* pane in the *Runtime Details* screen shows a list of all the applications that are currently bound to the selected runtime. The information displayed includes details of the runtime type, version, and ID.

      > → Tip
      >
      > If no applications are bound to the selected runtime, the *Bound Applications* pane is not displayed.

   2. Restage and start each bound application.
      In the *Bound Applications* pane, click each bound application link, and in the *Application Overview* screen, use the tools available in the toolbar to *Restage* and *Start* each of the applications that are still bound to the runtime you want to delete.
   3. Remove all instances of the restaged application.
      In the toolbar of the *Application Overview* screen, choose *Clear App Instances*.
   4. Check that no applications are bound to the runtime selected for deletion.
      Return to the *Runtime Details* screen to confirm that the list of bound applications is empty.

      > → Tip
      >
      > If any applications are still bound to the runtime, repeat the steps above. If no applications are bound to the runtime, you can delete the runtime, as described in the next step.

e. Delete the runtime.

   Choose 🗑 *(Delete runtime)*.

Related Information

Monitoring XS Advanced Application Runtimes [page 1400]
View the System Resources Utilized by Applications [page 1411]

14.2.3.1.4 Managing Domains & Certificates in XS Advanced

View and manage the domains and domain certificates in your XS Advanced runtime platform.

Context

The SAP HANA XS Advanced Cockpit includes a tool named **Domains & Certificates** that enables you to display a list of domains currently setup in your XS advanced instance. It shows the type of the domain, e.g. shared, external or private. If it's a shared domain, it can be the default domain or an external domain. The external domain may include a port which is also listed in the table.

Procedure

1. Display a list of all domains in your XS advanced instance
   Chose **Domains & Certificates** in the home screen of the XS Advanced Cockpit.
2. Create a domain
   1. Click on **Create Domain**.
   2. In the Create Domain dialog, provide the domain name and select one option from the following options: **Create Shared Domain**, **Create External Domain** and **Create Domain in Organization**.
3. Delete a domain
   1. Make sure there are no routes associated with the domain, otherwise deleting the domain is not possible.
   2. Click on the **trash-can symbol** to delete the selected domain.

   > → Tip
   >
   > The default domain can not be deleted.
4. Display the domain certificates of an individual domain
   1. Select a domain in the list of domains.
   2. Choose **>** to go the domain certificate details page of the selected domain with information about the certificate, e.g. **Updated At**, **Created By** and the **Certificate Chain**.
5. Set a custom certificate in the certificate details page
   1. Click on **Set Domain Certificate**.
   2. In the **Set New Domain Certificate** dialog, click on **Upload** and navigate to the certificate on your local file system. If the certificate is provided as multiple files, select them all.

3. The certificate can be in **PKCS8/PEM**, **PKCS12** or **PSE** format. If the certificate file has in some common file ending, it will auto select the correct certificate format radio button. Otherwise the user can manually select the correct certificate format.

6. Reset from expired or corrupted Certificates in the certificate details page
    1. Click on **Reset Domain Certificate**.
    2. This will generate a new, self-signed, default certificate for XS Advanced. Restaging the applications using the domain might be required.

For more information, see Maintaining Domains and Certificates in XS Advanced (CLI).

14.2.3.1.5 Managing XS Advanced Applications

Display a list of options available for managing XS advanced applications.

You can start, stop an application or restart an application that is currently running. For example, if you need to make changes to the active application. You can also create a new application instance. For example, if the performance of your application is impaired because of excessive load on the application, you can manually start a new instance of the application to take up any new load.

The XS advanced cockpit provides the following tool panes to help you manage XS advanced applications:

> → Tip
>
> To display a list of all the XS advanced applications, for example, those running in the *SAP* space, choose *Home* > *Org* > *SAP*.

Application-Management Tools in XS Advanced Cockpit

Views	Description
Overview	Display the summary of an XS advanced application's configuration and perform actions on the application instances, for example; stop, start, restart, restage, etc.
Service Bindings	Manage the bindings between applications and the services that are available in the SAP service marketplace.
Security	Manage the roles, role templates (and any optional attributes), and authorization scopes required to provide access to an XS advanced application.
User-provided Variables	Manage user-defined information for an XS advanced application instance, which can include deployment variables required for an application, MTA services, the target runtime, etc.

Views	Description
Environment Variables	Manage system-provided information for an XS advanced application instance, which can include details of the `VCAP_APPLICATION` (application, instance ID, etc.) and `VCAP_SERVICES` (database name, service plan, etc.).
Instance Monitoring	Display an overview of the system resources currently being consumed by the various instances of your XS advanced applications
Pinned Hosts	View a list of the hosts to which an application is pinned.
Logs	Display the application-specific logs.
Events	View a list of the actions performed on an application instance.

Related Information

Manage an Application [page 1405]
Control Access to Applications [page 1409]
View the System Resources Utilized by Applications [page 1411]

14.2.3.1.5.1 Manage an Application

Context

The application-overview page shows information about the following areas:

- *Application Routes*
  This includes the route's URL.
- *Application Information*
  This includes the number of instances (and their current state, for example, "running") and the buildpack used for deployment.
- *Instances*
  This includes the instance ID, the current state (Stopped, Running, Crashed), the host where the instance is running, the internal port and user account where the application is running, as well as the date and time when the application instance was created and stopped.
- *Most Recent Application Events*
  This includes the five most recent actions performed on the application instance with a time stamp and description as well as the user involved and an indication of the incident's severity.

You can use the application overview page to perform a number of important tasks.

Procedure

1. Navigate to the space where the XS advanced applications that you want to maintain are running.

   For example, to view the application installed in the space *SAP*, choose *Home* > *org* > *SAP*

   > → Tip
   >
   > To switch to another space in the same organization, choose *(Switch to another space)*.

2. Select the application you want to maintain and choose a task.

   In the *Applications* pane, you can perform the following tasks:

Task	Choose
Start an application that is currently stopped Start the selected application. → Tip The information in the *Requested State* column indicates the current state of the appliction, for example, *Started*/*Stopped*.	▷ *(Start)*
Stop an active application	⦿ *(Stop)*
Delete an application	🗑 *(Delete)*

3. Perform additional tasks on a selected XS advanced application.

   Choose the application link in the *Name* column to display an overview of the application's configuration including routes, available application instances, and a summary of the most recent events. In the application *Overview* screen, you can perform the following tasks:

Task	Choose
Stop and restart the currently selected application instance.	*Restart* → Tip If the application instance is not running, this action is not possible. Use the *Start* option instead.
Restage the currently selected application instance.	*Restage*

Task	Choose
Start the currently selected application instance	*Start* → Tip If the application instance is already running, this action is not possible. Use the *Restart* option instead.
Stop the currently selected application instance	*Stop* → Tip If the application instance is already stopped, this action is not possible.
Create a new instance of an application	⊕ *Instance*
Stop an **instance** of an application	⊖ *Instance* → Tip If there is only one application instance currently running, this action is not possible.
Delete all instance of an application	*Delete*
Clear all stopped or crashed instances of an application.	*Clear App Instances*

4. Bind an application to a service available in the service marketplace.

   In the navigation pane, choose *Service Bindings*
5. View and manage the security-related components of an XS advanced application instance.

   Choose between the following areas:

   - *Roles*
     A role instance is based on a role template which can (cut does not have to) have attributes.
   - *Scopes*
     Users maintained in an SAML identity provider need authorization scopes to access XS advanced applications. These scopes are defined in permissions available for a role,which are grouped in role collections.
   - *Attributes*
     Optional attributes are used to define role templates. Values are provided by the XS administrator when creating instances of a role based on a role template.
   - *Role Templates*
     The XS adminstrator uses roles templates to create role instances by providing concrete values for the attributes defined in the role templates.
6. Manage an XS advanced application's environment.

Display an overview of the variables used to configure an XS advanced application's environment; the information displayed is split into the following sections:

- *System-Provided Variables*
  The information includes details of the `VCAP_APPLICATION` (application, instance ID, etc.) and `VCAP_SERVICES` (database name, service plan, etc.).
- *User-Provided Variables*
  This information can include deployment variables required for an application, information about the Java Machine version (if applicable), MTA services, target runtime, etc.

> → Tip
>
> To add a new user-provided variable, choose *Add Variable*; to maintan an existing variable (for example, edit or remove), choose either *(Modify)* or *(Delete)*, respectively

7. Display an overview of the system resources currently being consumed by application instance.

   In the navigation pane, choose *Instance Monitoring*
8. View a list of the actions performed on an application instance.

   In the navigation pane, choose *Events*
9. Display the application-specific logs.

   In the navigation pane, choose *Logs*. The following options are available:

   - *Change Log Level*
     Use this option to display log details of a specific type, such as fatal, error, warning, and so on.
   - *Source*
     Use the drop-down menu to view only those logs generated by the specified source, for example:
     - *API*
       A log entry created by the XS advanced platform regarding this application
     - *APP*
       `[APP/<Droplet index>-<Instance index>]`
       The output of an application instance, either on `stdout` (prefixed with `'OUT'`) or on `stderr` (prefixed with `'ERR'`). In the log type tag, the droplet and application instance is encoded in this case. For example, "`APP/1-0`" means that the application instance with index 0 of droplet 1 dumped a log line to `stdout`.
     - *RTR*
       The access log for this application created by the platform router (RTR).
     - *STG*
       `[STG/<Droplet index>]`
       A log entry created by the staging process regarding this application. The entry displays the droplet index as a suffix. For example, "`STG/1`" means that this staging process produced the droplet with index 1.
   - *Type*
     View logs of specified types.
     - *OUT*
       A log entry was written to Standard Out (`stdout`).
     - *ERR*
       A log entry was written to Standard Error (`stderr`).
     - *ACC*

The log entry was written to the application instance's access log file.

- *LOG*
  The log entry was written in list log format to `stdout` and the logger starts with "`/Application`".
- *SYS*
  The log line was written in list-log format to Standard Out (`stdout`)
- *TRC*
  A log entry was written to trace

- *Recent | All | More*
  View all logs, only recent logs, more logs from the same log file..
- *Download*
  Download recent or all logs.

10. View a list of the hosts to which an application is pinned.

    In the navigation pane, choose *Pinned Hosts*. The following options are available:

    - *Pin to Host*
      Choose this option to pin the application to the listed hosts.
    - *Change Pin Mode*:
      Strict is the default pin mode. In this mode, the application cannot start if the pinned host is not available.
      You can also choose the **Relaxed** pin mode. In this mode, the application can start on any other host if the pinned host is not available.

Related Information

Managing Services in XS Advanced [page 1434]
Displaying Application Information in XS Advanced [page 1288]

14.2.3.1.5.2 Control Access to Applications

Use roles, role templates, and authorization scopes to control access to XS advanced applications.

Context

You can create user roles in the SAP HANA XS Advanced Cockpit to control access to XS Advanced applications. These roles are derived from role templates defined in the security description (xs-security.json) of applications that have been registered as OAuth 2.0 clients in the User Account and Authentication (UAA) service during application deployment. The application security description file also contains details of the authorization scopes that are used for application access, and defines any attributes that need to be applied. You can then add roles to role collections before the roles are assigned to SAP HANA database users or users logging on with SAML 2.0 assertions.

Procedure

1. On the organization *Overview* screen, choose a space.

   The applications deployed in the space are displayed as a list.
2. In the application pane, select an application from the *Name* column.

   The application overview page appears.
3. In the navigation pane, choose *Security*.
4. Perform the following actions to maintain secure access to the application:

Task	Choose	Details
Create a new role	*Roles* in the navigation pane	A role is an instance of a role template. You can create a role based on a role template and assign the role to a role collection. You need to provide the following details to create a role: • *Name*: Name of the role • *Description*: Additional details about the role • *Template*: Templates are predefined for an application. Choosing a template identifies the scope and attribute applicable for the role. • *Attribute*: Provides attribute details applied to the role. Depending on the value of the attributes defined, access to resources is either granted or restricted. For example, in a sales scenario, the attribute region emea could be used to restrict access to the sales orders for the geographical region EMEA.
Update a role description	*Roles* in the navigation pane. In the *Roles* pane, choose *(Edit)* in the *Actions* column.	Maintain role descriptions.
Delete a role	*Roles* in the navigation pane. In the *Roles* pane, choose *(Delete)* in the *Actions* column.	Maintain one or more roles.

Task	Choose	Details
Assign roles to role collections	Roles from the navigation pane. In the *Roles* pane, choose ⊕ *(Add to role collection)* underneath the *Actions* column.	Role collections contain a list of assigned roles, which you can maintain with this option. This option enables you to assign roles to role collections, which are then assigned in turn to SAP HANA users or SAML 2.0 groups.
Check scopes applicable for the role	Scopes from the navigation pane	View the permissions available for a role.
Check attributes	Attributes from the navigation pane	Attributes define information that comes with the respective user, for example, 'cost center' or 'country'. This information can only be resolved at runtime.
Check role templates available for the application	Role templates from the navigation pane	The role template defines the type of access permitted for an application, for example, the authorization scope and any attributes that need to be applied.

14.2.3.1.5.3 View the System Resources Utilized by Applications

Display an overview of the resources used by an XS advanced application.

Context

You can view the system resources utilized by an XS advanced application.

Procedure

1. Open the *Organizations* pane.
2. Navigate to the space whose resources you want to view.

   For example, choose *org* > *SAP*

   A list of the applications running in the selected space is displayed along with a summary of the instances available along with a selection of actions that you can perform on an individual application instance, for example: ▷ *(Start)*, ⊙ *(Stop)*, 🗑 *(Delete)*.

3. Open the *Monitoring* pane.

   This displays a list of the applications running in the selected space along with details of the resources currently being consumed by each application, for example: *Memory*, *CPU Time*, *User Mode* or *Kernel Mode*, etc.

4. To sort or group applications based on resource consumption details, choose (View Settings).
5. In the *View* dialog, choose *Sort By* or *Group By* to arrange applications.

Related Information

Resource Details [page 1412]
Maintaining Organizations and Spaces in XS Advanced [page 1393]

14.2.3.1.5.3.1 Resource Details

An overview of the system-utilization components that are displayed.

The *Monitoring* screen provides a summary of the resource details of the XS applications running in the XS Advanced runtime. The following table contains descriptions of the resource details:

User Interface Element	Description
Name	The name of the application
Memory (KB)	The amount of memory used by an application
CPU (ms)	The amount of processing time (in milliseconds) used by the application
User Mode (ms)	The amount of user-mode time (in milliseconds) consumed by an application. In user mode, an application cannot directly access hardware or modify memory; it can only do so by means of a proxy such as an API.
Kernel Mode Time (ms)	The amount of kernel-mode time (in milliseconds) consumed by an application. In kernel mode, an application has unrestricted access to CPU instructions and memory addresses.
Access Count	The number of times an application has been accessed.
MTA File	The name of the multi-target application (MTA) to which the listed application belongs.

User Interface Element	Description
Host	The name of the host where the listed instance of an XS Advanced application is running.

Related Information

View the System Resources Utilized by Applications [page 1411]

14.2.3.1.6 Managing Hosts in XS Advanced

This page displays the host systems on which XS Advanced applications are running. It provides specific details about the number of applications running on a host system and the spaces pinned to it. It also displays whether an application can only run on a specific host.

Related Information

Maintaining Host Pinning [page 1338]

14.2.3.1.6.1 View Host Systems

Display a list of the hosts running XS advanced applications.

Context

You can display an overview of the systems hosting XS advanced applications. The information displayed shows which applications and spaces are currently pinned to the selected host.

Procedure

1. In the *Home* navigation pane, choose *Host Management*.
2. Select a host from the list to which applications can only run on the selected host and which spaces are pinned to it.

> → Tip
>
> For more details about the information displayed in the *Host Management* pane, for example, *Exclusive Pin (false)* see *Related Information* below.

Related Information

Host System Details [page 1414]
Managing Hosts in XS Advanced [page 1413]
Maintain Users in a Space [page 1399]

14.2.3.1.6.1.1 Host System Details

An overview of the configuration of a system hosting XS advanced applications.

The *Host Management* page displays the following information:

User Interface Element	Description
Host ID	The name of the system hosting the XS advanced applications
Exclusive Pin	• *true* Indicates that the system is available exclusively for specific applications and spaces. • *false* Indicates that the system is available as a shared entity for all applications and all spaces.
Pinned Applications	The number of applications that are set to run on the host system → Tip The *Pinned Applications* section displays a list of XS advanced applications pinned to the current XS advanced host system and the corresponding space where each application is deployed.

User Interface Element	Description
Pinned Spaces	The number of spaces mapped to the host system → Tip The *Pinned Spaces* section displays a list of the spaces pinned to the currently selected XS advanced host system and the corresponding organization to which each pinned space belongs.

Related Information

Managing Hosts in XS Advanced [page 1413]
View Host Systems [page 1413]

14.2.3.1.7 Maintaining Database Instances in XS Advanced

Applications deployed within a space in XS Advanced must be able to persist data. The XS Advanced Cockpit includes an option to maintain a database to persist application data. You can perform the following tasks to persist data:

- Create a database in XS Advanced
- Search for a database by name (or part of a name) in XS Advanced
- Display the status of all databases currently available in XS Advanced
- Enable a database for use with XS Advanced
- Disable a database in XS Advanced
- Delete a database in XS Advanced
- Map a database to an organization or space

Related Information

Maintaining Tenant Databases in XS Advanced [page 1331]

14.2.3.1.7.1 Create a Database

Maintain new database instances.

Context

Create a database to store data from applications running in XS Advanced, as follows:

Procedure

1. In the *Home* navigation pane, choose *Tenant Databases*.
2. On the *Tenant Databases* screen, choose *New Tenant Database*.

   While the new database is being created, the system shows the status *Creating*. When the new database is ready for use, the status automatically changes to *Running*.

Related Information

Database Details [page 1416]

14.2.3.1.7.1.1 Database Details

A brief summary of the XS advanced database configuration.

You provide the following details while creating a new logical database:

User Interface Element	Description
Name	Name of the database

User Interface Element	Description
Internal Port	This is an optional field. You can provide details of a specific port on the system to run the tenant database. The port must adhere to one of the following rules: • Accepts a value range and meets the condition (`<port number>` - (30000+40)) % 3 == 0. For example, the range can vary from 30040 to 30100. **Note** The port value (3`<instance number>`00) varies with the instance number. If the instance number is 08, the port value will be 30800. Confirm the instance number on your system and provide the respective port details. • Accepts the port number of the default index server.
Host	This is an optional field. If you want to host the tenant database on a specific system, provide the host details.
Tenant Database Password	Provide the password of the `SYSTEM` user to access the tenant database.
System Database User	Provide the `SYSTEM` user name as available on the SAP HANA system database.
System Database Password	Provide the password of the `SYSTEM` user of the SAP HANA system database.

Related Information

Create a Database [page 1416]
Manage a Database [page 1418]

14.2.3.1.7.2 Manage a Database

View and maintain the configuration of an XS advanced database.

Context

To use a database once you have created it, you need to enable and map the database to an organization or space.

Procedure

1. In the *Home* navigation pane, choose *Tenant Databases*.
2. On the *Tenant Databases* screen, you can perform any of the following tasks:

Task	Choose	Description
Make a logical database available for use	Enable	To use the database once you have created it, you need to enable it. The enabled database is available for use in XS Advanced. This is a toggle option. After enabling, the toggle option appears as *Disable*.
Use an organization or space with a database	*(Map)*	To persist application data, you must map an organization or a space to the database. You can map an organization to a single database whereas a space can be mapped to multiple databases or vice versa.
Prevent a database from being used	*(Disable)*	To ensure that a database is no longer available for use with XS Advanced, use the disable option.
Delete a database	*(Delete)*	To remove a database, use the delete option.

Task	Choose	Description
Search for a specific database from a list of available databases	Search	Use the *Search* option to find an existing database by name (or any part of the name). If you enter only part of a name, the search option filters the list of logical databases and displays only those whose names include the string you typed. If you type **DB** in the search field, the list of databases displayed is restricted to any names that contain the string "DB", for example, "`MyTenantDB`", "`DB1`", or "`MyLogicalDB5`".

Related Information

Create a Database [page 1416]
Database Details [page 1416]

14.2.3.1.8 Maintaining Security in XS Advanced

During application development, developers create authorization information for business users. This information is made available to the administrators who complete the authorization setup. They are responsible for assigning authorizations to business users.

Developers store authorization information as design-time role templates in the security descriptor file *xs-security.json*. Using the xsuaa service broker, they deploy the security information to a dedicated XS Advanced application. The XS Advanced administrators view the authorization information in role templates, which they use as part of the runtime configuration. The administrators use the role templates to build roles, which are aggregated in role collections. The role collections are then assigned to business users.

The tasks required to set up authorization artifacts in SAP HANA XS Advanced are performed using two distinct user roles: the application developer and the SAP HANA XS Advanced administrator. After the authorization artifacts have been deployed as role templates, the administrator of the SAP HANA XS Advanced application uses the role templates provided by the developers to build role collections and assign them to business users using the SAP HANA XS Advanced Cockpit.

> ⓘ Note
>
> To test authorization artifacts after deployment, developers can use the role templates to build role collections and assign authorizations to business users in the SAP HANA XS Advanced Cockpit.

The table below lists the sequence of tasks along with the user roles needed to set up authorization artifacts:

> **Note**
>
> Application developers must have the *Space Developer* role in the required space. XS Advanced administrators must have the *XS_AUTHORIZATION_ADMIN* role.

Step	Task	User Role	Option to be Used
1	Specify the security descriptor file containing the functional authorization scopes for your application.	Application Developer	Text Editor
2	Create role templates for the XS Advanced application using the security descriptor file.	Application Developer	Text Editor
3	Create a service instance from the xsuaa service in XS Advanced using the service broker.	Application Developer	XS Advanced CLI tool / XS Advanced Cockpit
4	Bind the service instance to the XS Advanced application by including it in the manifest file.	Application Developer	Text Editor
5	Deploy the XS Advanced application.	Application Developer	XS Advanced CLI tool
6	If required, create a new role using role templates available within the application in the XS Advanced Cockpit.	XS Advanced administrator	Access applications within the space.
7	Create a role collection and assign roles to it.	XS Advanced administrator	Access applications within the space.
8	Assign the role collection to an SAML 2.0 identity provider or to SAP HANA database users.	XS Advanced administrator	Access applications within the space. Access SAML Identity Provider.
9	Assign the users to roles using the role collections	XS Advanced administrator	User interface of SAP HANA XS Advanced Cockpit

Related Information

Control Access to Applications [page 1409]
Managing SAML Identity Providers in XS Advanced [page 1423]
Manage Users [page 1391]
Map Role Collections to an SAML Identity Provider [page 1428]

14.2.3.1.8.1 Enable Access to XS Advanced Cockpit

Assign the roles and role collections required to enable access to XS Advanced Cockpit.

Context

The XS Advanced Cockpit displays all the options required to maintain the XS Advanced model runtime configurations. The options listed in the following table are available only to users who have been assigned the suitable role collection. The table shows the role collection required to use an option.

Options	Role Collection	Comments
Monitor	XS_CONTROLLER_ADMIN	Based on the role collection assigned, the user is permitted to perform some, most, or all operations in the application. The different role collections are: • Admin : No access restrictions • User: Modify access within the assigned organization or space • Auditor: Read-only access within the assigned organization or space
Organization and Space Management	XS_CONTROLLER_USER or	
SAP HANA Service Broker Configuration	XS_CONTROLLER_AUDITOR	
Application Roles	XS_AUTHORIZATION_ADMIN or	Based on the role collection assigned, the user is permitted to perform some (or all) operations in the XS Advanced Cockpit. The different role collections are: • Admin: Full admin edit access to the tool • Display: Read-only access to the tool
Trust Configuration	XS_AUTHORIZATION_DISPLAY	

Options	Role Collection	Comments
Member Management	XS_USER_ADMIN	Based on the role collection assigned, the user is permitted to perform all operations in the application.
SAP HANA Database Setup	XS_CONTROLLER_ADMIN	The user is permitted to perform all operations in the application.
Trusted Certificates		

As an administrator, you can see the default role collections available for your current role.

Procedure

1. In the *Home* navigation pane, choose *Security* > *Role Collections*.
2. On the *Role Collections* screen, perform any of the following tasks:

Tasks	Choose	Description
Create a new role collection	*New Role Collection*	Use this option to create a role collection.
Edit a role collection	*(Edit)*	Use this option to update descriptions of an existing role collections. → Tip You cannot edit a default role collection.
Delete a role collection	*(Delete)*	Use this option to delete an existing role collection → Tip You cannot delete the default role collection.

Related Information

Maintaining Security in XS Advanced [page 1419]
Assign Roles to Role Collection [page 1423]

14.2.3.1.8.2 Assign Roles to Role Collection

Use roles and role collections to control access to XS advanced applications.

Context

As an administrator, you can control access to applications by providing role-based access to business users. To provide role-based access, you select the role templates and roles available for an XS Advanced application. You then assign the roles to role collections. The role collections are then assigned to business users to authorize them to access the application.

Procedure

1. In the *Home* navigation pane, choose *Security* > *Role Collections*.
2. On the *Role Collections* screen, choose the name of a role collection.

   The roles already assigned to the selected role collection are displayed.
3. On the *Roles* screen, choose *Add Role*.
4. In the *Add Role* dialog, select an application from the *Application Identifier*.
5. Choose the respective *Role Template* and *Role*.

Related Information

Manage Users [page 1391]
Map Role Collections to an SAML Identity Provider [page 1428]

14.2.3.1.9 Managing SAML Identity Providers in XS Advanced

Enable applications to use SSO authentication based on Security Assertion Markup Language (SAML) certificates.

You can configure an SAP HANA system to act as a service provider for XS Advanced applications that use single sign-on (SSO) authentication based on Security Assertion Markup Language (SAML) certificates.

The XS Advanced Cockpit includes a *IdP Trust Configuration* tool, which you can use to configure SAML identity providers at run time. You must perform this step if you want your SAP HANA XS advanced applications to use SAML assertions as the logon authentication method.

To configure an SAML identity provider (IdP) in XS advanced, you need to perform the following tasks:

1. Generate the user-account-and-authentication (UAA) trust metadata and configure it at the SAML IdP.
2. Create an IdP trust configuration for the SAML identity provider.
3. At the SAML identify provider, define SAML groups that you want to map to role collections, and assign users to these groups.
4. Map SAML groups to role collections defined by applications or by the platform.

Related Information

Generate the Metadata Required to Configure an Identity Provider [page 1424]
Create an SAML Identity Provider [page 1425]
Map Role Collections to an SAML Identity Provider [page 1428]

14.2.3.1.9.1 Generate the Metadata Required to Configure an Identity Provider

Configure an SAML Identity Provider (IdP) using metadata that you generate and download.

Context

You need to generate a metadata (XML) file that contains information about the service provider (SAP HANA system) to configure the SAML identity provider. You do not have to be logged on to the SAP HANA XS Advanced Cockpit to generate the metadata.

Procedure

1. Log into the XS Advanced Cockpit as `XSA_ADMIN`.
2. In the *Home* navigation pane, choose *Security* > *IdP Trust Configuration*.
3. Display the trust metadata for user account and authentication.

   Choose *Show UAA Trust Metadata*.

   Alternatively, you can display the UAA trust metadata in a Web browser with the following URL without being logged into XS Advanced Cockpit:

   ```
   https://<authorizationEndpoint>/saml/metadata
   ```

   > → Tip
   >
   > The authorization end-point for XS advanced is included in the output displayed by the command `xs -v[ersion]`, for example, under the key `authorizationEndpoint`.

4. Copy the metadata to the clipboard.

   Choose *Copy to Clipboard*.
5. Configure the SAML Identity Provider.

   Use the copied metadata to configure the new SAML IdP.

Related Information

Map Role Collections to an SAML Identity Provider [page 1428]
The XS Command-Line Interface [page 1262]

14.2.3.1.9.2 Create an SAML Identity Provider

Configure SSO authentication based on Security Assertion Markup Language (SAML) certificates

Prerequisites

You have configured the SAML identity provider with the assertion attribute **Groups** to be sent at runtime. Ensure that you have entered **G** in uppercase.

Context

You need to create an SAML Identity Provider (IdP) for an SAML service provider to authenticate users signing in to the application by means of single sign-on (SSO). SAP HANA supports the use of SSO authentication based on Security Assertion Markup Language (SAML) certificates.

Procedure

1. In the *Home* navigation pane, choose *Security* > *IdP Trust Configuration*.
2. On the *IdP Trust Configuration* screen, choose *New Trust Configuration*.
3. In the *New Trust Configuration* dialog. provide the required values.

Mandatory IdP Trust Configuration Values

Option	Details
Metadata *	The identity provider's SAML metadata which you must fetch from the IdP. This step is specific to the particular IdP.
Name *	The human-readable name of the new trust configuration
*Origin Key**	The unique technical name that identifies this IdP. You need this key to identify an IdP later on, for example, with the xs command-line client. You can choose this key freely, for example SAML.

4. Check the validity of the new trust configuration.

   Chose *Parse* to verify that the details provided in the new trust configuration are valid.
5. Save the configuration.

   Choose *Save* to close the *New Trust Configuration* dialog and add the new trust configuration to the list.

> → Tip
>
> You can use the *(Edit)* and *(Delete)* tools provided in the *Actions* column to maintain details of the trust configuration.

Related Information

Trust Configuration Details [page 1426]
Managing SAML Identity Providers in XS Advanced [page 1423]
Maintaining Security in XS Advanced [page 1419]
Generate the Metadata Required to Configure an Identity Provider [page 1424]

14.2.3.1.9.2.1 Trust Configuration Details

You can choose the *Edit* option to view details of an SAML IDP. To provide the required details for trust configuration, choose *New Trust Configuration*.

User Interface Element	Description
Upload	Choose this button to upload the XML format of the SAML certificate.
Origin Key	Unique name for the SAML identity provider. This value is mandatory.

User Interface Element	Description
Name	Human-readable name of the remote SAML identity provider. This value is mandatory.
Description	More details describing the SAML identity provider.
Status	State of the SAML identity provider.
Show SAML login link on login page	If the value is set as *Yes*, the SAML link appears on the logon page.
Link Text	The text representing the SAML logon URL on the logon page.
Metadata	The text containing the SAML certificate.
Show Details	If the contents of the XML document are valid, the parsing process extracts the information that needs to be inserted into the Origin Key, Subject, Entity ID, and Issuer fields, and the URL fields such as Single Sign-On URLs and Single Log-out URLs.
Parse	Choose this option to validate the SAML certificate if it is copied in XML format.
SingleSignOn URL (RedirectBinding)	URL of the IDP endpoint for SSO requests using SAML redirect binding.
SingleSignOn URL (PostBinding)	URL of the IDP endpoint for SSO requests using SAML post binding.
SingleLogout URL (RedirectBinding)	URL of the IDP endpoint for single logout (SLO) requests using SAML redirect binding.
SingleLogout URL (PostBinding)	URL of the IDP endpoint for single logout (SLO) requests using SAML post binding.

14.2.3.1.9.3 Map Role Collections to an SAML Identity Provider

Map SAML assertion attributes to XS advanced attributes to enable SAML-based single sign-on (SSO).

Prerequisites

- You must already have created a trust configuration for the SAML identity provider.
- The SAML identity provider must be configured to send a user's SAML groups membership in a single SAML assertion attribute named `Groups`, with an attribute value per group.
- The `Groups` attribute name must start with an uppercase *G*.

Context

Users maintained in an SAML identity provider need authorization scopes to access XS Advanced applications. These scopes are contained in roles grouped in role collections. To provide authorization scope to SAML users logging on to an application using single sign-on (SSO), or using SAML-based SSO, you need to map SAML groups maintained in an SAML identity provider to XS Advanced role collections.

Procedure

1. In the *Home* navigation pane, choose *Security* > *IdP Trust Configuration*.
2. On the *IdP Trust Configuration* page, choose the name of the SAML IdP.
   The *Overview* page of the SAML IdP appears.
3. In the trust configuration navigation pane, choose *Role Collection Mappings*.
4. Choose *New Role Collection Mapping*.
5. In the *Create Role Collection Mapping* dialog, provide the following details:

User Interface Element	Description
Role Collection	The dropdown list contains names of assertion-based role collections associated with the selected application.

User Interface Element	Description
Attribute	Displays the attribute defined in the selected application's security configuration (`xs-security.json`) file. Currently, the only attribute allowed is *Groups*. **⚠ Restriction** The initial letter **G** must be typed in uppercase.
Operator	The operator to be used along with the attribute in the specified rule. **⚠ Restriction** Currently, the only operator allowed is *equals*.
Value	The value of the attribute (the name of the SAML group) to be used for the rule that triggers the assignment of the selected role collection.

Related Information

Managing SAML Identity Providers in XS Advanced [page 1423]
Create an SAML Identity Provider [page 1425]
Trust Configuration Details [page 1426]
Manage IdP Trust Configuration [page 1429]
Generate the Metadata Required to Configure an Identity Provider [page 1424]

14.2.3.1.9.4 Manage IdP Trust Configuration

Maintain details of the trust configuration for an Identity Provider (IdP).

Prerequisites

Make sure that you have configured the SAML identity provider with the assertion attribute **Groups** to be sent at runtime.

> **ⓘ Note**
>
> The name of the authorization-scope attribute `Groups` must start with the letter **G** (in uppercase).

For information about how to configure the identity provider of your choice, see *Federation Attribute Settings of Any Identity Provider* in *Related Information* below.

Context

You can modify the status or metadata details of SAML certificates. You can also delete unwanted certificates.

Procedure

1. In the *Home* navigation pane, choose *Security* > *IdP Trust Configuration*.
2. On the *IdP Trust Configuration* screen, perform any of the following tasks:

Task	Action	Description
Update the status of an SAML certificate	Choose *(Edit)*	Choose this option in the *Actions* column to change the status of a certificate.
Update the SAML metadata	Select the name of the *SAML IdP* and choose *Edit*.	Choose this option in the IdP details (*Overview*) screen to modify the SAML metadata.
Delete an unwanted SAML certificate	Choose *(Delete)*	Choose this option in the *Actions* column to remove an SAML certificate.

Related Information

Create an SAML Identity Provider [page 1425]
Generate the Metadata Required to Configure an Identity Provider [page 1424]
Federation Attribute Settings of Any Identity Provider (SAP BTP)

14.2.3.1.10 Managing Trust Certificates in XS Advanced

For secure connections between SAP HANA systems and XS Advanced applications, each system needs to maintain a trusted certificate. This page displays a list of certificates currently used to establish trusted connections. It also displays whether the current certificate being used is still active, expired, or is due to expire. You can perform the following tasks:

- Create a certificate
- Set a certificate for the default domain
- Manage certificates

Related Information

Upload a Trust Certificate [page 1431]
Set the Default Certificate for XS Advanced [page 1432]
Manage Certificates [page 1433]
Maintaining Trust Certificates in XS Advanced [page 1321]

14.2.3.1.10.1 Upload a Trust Certificate

Add a certificate to the XS advanced platform's IdP trust store.

Context

The XS advanced run-time platform includes tools that enable administrators to maintain and manage the certificates used to set up trusted relationships. You can use these tools to upload custom trust certificates to the XS advanced platform. The trust certificates are propagated to the applications by the build packs. For example, the propagation of the certificates is performed by default when using either the SAP Node.js or the SAP Java build pack. It is also possible, however, to use custom trust certificates, for example, if you have configured a custom “certificate authority” within your corporation.

This option allows you to upload an existing trust certificate to the XS advanced platform's trust store, for example, to enable any application to establish fully trusted SSL sessions with corporate servers, even those configured outside the SAP HANA system..

Procedure

1. In the *Home* navigation pane, choose *Security* > *Trust Certificates*.
2. Choose *New Trust Certificate*.
3. In the *New Trust Certificate* dialog, provide the following details:

User Interface Element	Description
Alias	Name for the new trusted certificate. This name appears in the *Alias* column of the *Trusted Certificates* list.
Choose Certificate	Browse to the file containing the certificate.
Use with SAP HANA Service Broker	This option means that the certificate can be used when establishing the database connection.

Related Information

Managing Trust Certificates in XS Advanced [page 1430]
Set the Default Certificate for XS Advanced [page 1432]
Manage Certificates [page 1433]

14.2.3.1.10.2 Set the Default Certificate for XS Advanced

Set the default trust certificate for the XS advanced platform domain.

Context

You can set a certificate for the default domain in XS advanced, as follows:

Procedure

1. In the *Home* navigation pane, choose *Security* > *Trust Certificates*.
2. Choose *Set Platform Default Certificate*.
3. In the *Set Platform Default Certificate* dialog, provide the following details:

User Interface Element	Description
Choose Certificate	Browse to the file containing the certificate.
Choose Key	Private key in PKCS8 PEM format.

Related Information

Managing Trust Certificates in XS Advanced [page 1430]
Upload a Trust Certificate [page 1431]
Manage Certificates [page 1433]

14.2.3.1.10.3 Manage Certificates

Maintain trust certificates for XS advanced.

Context

The *Trust Certificates* screen enables you to view existing XS advanced trust certificates or delete any unneeded certificates.

Procedure

1. In the *Home* navigation pane, choose *Security* > *Trust Certificates*.
2. On the *Trust Certificates* screen, perform any of the following tasks:

Task	Choose	Description
View a certificate	*(View)*	You can use this option to view the details of a certificate. This option is available underneath the *Actions* column.
Delete a certificate	*(Delete)*	You can use this option to delete any unwanted certificates. This option is available under the *Actions* column.
Search for a specific certificate	Search	If the *Trust Certificates* screen lists a large number of certificates, you can use the search option to list a specific certificate. You can search for a certificate using any of the following certificate details: alias, usage, issuer, or valid details.

Related Information

Managing Trust Certificates in XS Advanced [page 1430]
Upload a Trust Certificate [page 1431]
Set the Default Certificate for XS Advanced [page 1432]

14.2.3.1.11 Managing Services in XS Advanced

Every space provides access to a marketplace for SAP services which can be bound to XS advanced applications.

The service marketplace displays SAP-approved services available for a selected space. Based on your role, you can access and use the services with your application. To use a service with your application, you need to create an instance of the service. After the instance is created, you can bind the new instance to your application. You can also perform the bind operation later using the *Service Instances* page, which displays a list of all currently available instances of a selected service, for example, *hana*.

> → Tip
>
> User-provided services enable you to create services for your application, which are not listed in the service marketplace, or custom services.

Related Information

Maintain Service Instances [page 1434]
Bind a Service Instance to an XS Advanced Application [page 1436]
Manage User-Provided Services [page 1438]
Maintaining Services in XS Advanced (CLI) [page 1305]

14.2.3.1.11.1 Maintain Service Instances

Create and manage instances of SAP HANA services.

Context

To use a service from a service marketplace with an XS advanced application, you first need to create an instance of the service. Once the instance is created, you bind it to the application.

→ Tip

For instructions describing how to bind a service instance to an application, see *Related Information* below.

To create a new instance of an existing SAP HANA service, perform the following steps

Procedure

1. Navigate to the Organization > Space in which the application is deployed.
2. Open the Service Marketplace to display a list of all available services.

   In the navigation pane, choose *Services* > *Service Marketplace*.
3. Select the service for which you want to create a new instance.

   For example, choose *hana (SAP HANA database)* .

   The *Service: hana - Overview* page displays a summary of the selected service, which includes a list of the corresponding service plans.
4. Display a list of available instances of the *hana* service.

   In the navigation pane, choose *Instances*.
5. Create a new service instance.

   Choose *New Instance* to start the instance-creation Wizard.
6. Choose a service plan.

   Select a service plan to assign to the new instance of the *hana* service and choose *Next*.

   "hana" Service Plans

Plan Option	Description
HDI Shared	The application receives the credentials required for access to an HDI container, which is basically a database schema that is equipped with additional metadata.
Schema	Creates a plain schema, which you need to manage manually or with application code. No automated deployment or schema-management services are provided.
Securestore	Enables a bound application to make use of stored procedures that provide access to the SAP HANA Secure Store.

7. (Optional) Provide parameters that need to be passed to the application during binding.

   The parameters must be provided in JSON format, either from a file that you can browse for in the local file system or directly in the text-box provided. Choose *Next* to move to the next step.
8. Check the details that you have provided for the service-binding configuration.

   Confirm that the service name and plan are correct, and choose if you want to check the parameters again.

> → Tip
>
> If you have not provided any parameters, the button is inactive and displays the message *No parameters*.

Enter a name for the service instance and choose *Finish* to create the new service instance and bind it to the selected application.

Related Information

Managing Services in XS Advanced [page 1434]
Bind a Service Instance to an XS Advanced Application [page 1436]
Manage User-Provided Services [page 1438]

14.2.3.1.11.2 Bind a Service Instance to an XS Advanced Application

Create an instance of a service which you can bind to an application.

Context

To use a service from a service marketplace with an XS Advanced application, you need to create an instance of the service. Once the instance is created, you bind it to the application.

Procedure

1. Navigate to the Organization > Space in which the application is deployed.
2. Choose the application.

   The application overview page appears.
3. In the navigation pane, choose *Service Bindings*.
4. In the service bindings pane, choose *Bind Service*.
5. Choose the service type.

In the *Bind Service* Wizard, select one of the following service-provider options:

Option	Description
Service from the Catalog	Use this option to bind an application to a instance of a service from the service marketplace.
User-Provided Services	Use this option to bind an application to an instance of a user-provided service.

Choose *Next*.

6. Select a service.

   In the list of services displayed, choose the service you want to bind your XS advanced application to and choose *Next*.

7. Choose a service plan.

   In this step, you choose the type of service instance to create and the service plan to assign to it, as follows:

   a. First, select one of the following service-instance options:

Option	Description
Create new instance:	This option creates a new instance of the service, for which you will need to choose a service plan.
Re-use existing instance:	This option enables you to bind the application to an existing instance of the service.

   b. Select the service plan to use with the new service instance.

Plan Option	Description
HDI Shared	The application receives the credentials required for access to an HDI container, which is basically a database schema that is equipped with additional metadata.
Schema	Creates a plain schema, which you need to manage manually or with application code. No automated deployment or schema-management services are provided.
Securestore	Enables a bound application to make use of stored procedures that provide access to the SAP HANA Secure Store.

8. (Optional) Provide parameters that need to be passed to the application during binding.

   The parameters must be provided in JSON format, either from a file that you can browse for in the local file system or directly in the text-box provided. Choose *Next* to move to the next step.

9. Check the details that you have provided for the service-binding configuration.

   Confirm that the service name and plan are correct, and choose if you want to check the parameters again.

> → Tip
>
> If you have not provided any parameters, the button is inactive and displays the message *No parameters*.

Enter a name for the service instance and choose *Finish* to create the new service instance and bind it to the selected application.

Related Information

Managing Services in XS Advanced [page 1434]
Manage User-Provided Services [page 1438]

14.2.3.1.11.3 Manage User-Provided Services

Maintain services that are not available in the service marketplace.

Context

To use services that are not listed in the service marketplace with your XS advanced application, you need to create user-provided services. User-provided service instances deliver service credentials to an application. A service instance for user-provided services behaves like the service instance created from the service marketplace.

Procedure

1. In the *Space* navigation pane, choose *Services* > *User-Provided Services*.
2. In the *User-Provided Services* pane, choose *New Instance*.
3. Provide an instance name.
4. Enter the service credentials required to deliver to the application.
5. Choose *Save*.

   The service instance is created. Similarly to services from the service marketplace, you need to bind the user-provided service instance to an application.

Related Information

Maintain Service Instances [page 1434]
Bind a Service Instance to an XS Advanced Application [page 1436]

14.2.3.2 Scheduling Jobs in XS Advanced

The Job Scheduler service enables you to create and schedule long-running operations or jobs.

In the SAP HANA XS advanced model, Job Scheduler is an application service. The Job Scheduler service enables you to create and schedule long-running operations or jobs. This service is deployed during the installation of the SAP HANA XS advanced model.

For high availability of Job Scheduler service, you can scale the service. All components of the Job Scheduler application such as Job Scheduler service, Job Scheduler REST, and Job Scheduler Broker support scaling. Scaling is required in scenarios where you have too many schedules, resulting in job execution delays. You need to scale only the job scheduler service component for the delay scenario. Scaling of the job scheduler service involves increasing the number of instances of the service, which needs to be done one by one. After successfully increasing with one instance, continue with creating the next one. For more information about scaling applications, see Scaling Applications in XS Advanced [page 1304].

The following table lists the sequence of tasks required to use an instance of the Job Scheduler service:

> ⓘ Note
>
> To configure and setup Job Scheduler, you require specific roles and permissions.

Step	Task	Role
1	Configure the Service Broker for Job Scheduler	Space Developer
2	Create a Job Scheduler Service Instance	Space Developer
3	Bind an Application to the Job Scheduler Service	Space Developer
4	Maintain jobs and job schedules	Administrator. To access and use the Job Scheduler Dashboard without having an administrator role, refer *The Job Scheduler Dashboard* topic.

Job Scheduler Execution Mode

Job Scheduler supports the following **modes** for applications to execute a job:

- Synchronous Mode
  Suitable for jobs that run for a short span of time, for example, an OData service endpoint. If an application uses this mode, it must adhere to the following guidelines:

- When the scheduler invokes the endpoint, the application must return the response with appropriate HTTP status codes, indicating success or failure.
- To indicate success, the application must use a suitable standard status code between 200 and 399, except 202-ACCEPTED.
- To indicate an execution failure, the application must use one of the server error codes as outlined in the HTTP protocol specification.

- Asynchronous Mode
  Suitable for jobs that run for a long span of time, for example, endpoints which trigger batch processing. If an application uses this mode, it must comply with the following guidelines and workflow:
  - When the scheduler invokes the endpoint, it passes the request headers **x-sap-job-id**, **x-sap-job-schedule-id**, **x-sap-job-run-id**, and **x-sap-scheduler-host** values for the Job ID, Job Schedule ID, Job Run ID, and the Job Scheduler Host URI, respectively. The application must extract the header values and store them using a suitable mechanism, such as in-memory storage that uses caches or libraries, or persistent storage that uses a database.
  - The application must return an acknowledgement response with the HTTP status code **202-ACCEPTED**. This response is an indication to the scheduler that the application has accepted and is processing the request. If the application returns a server error code, the scheduler interprets it as a failure of the job run.
  - After the application completes the job processing, it must invoke the **Job Run Log Update** API to indicate success or failure and (optionally) create log text for the job run. For information about this API, see *Job Scheduler REST API for XS Advanced* topic . The path parameters for the **Job Run Log Update** API call must be the values of the headers as described above.
  - If the application doesn't invoke the **Job Run Log Update** API, the scheduler isn't notified of the status of the job run and, after a configurable time interval, reverts the job to the status UNKNOWN.

Behavior of Job Scheduler

Job Scheduler contains certain default behaviors that you can override. To override the default behavior, define parameters with specific values in the mtaext file during deployment. For more information about the mtaext file, refer to the topic *The MTA Deployment Extension Descriptor*, in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*. The default behaviors are as follows:

- Triggering missed executions (**ENABLE_MISFIRE**): Job Scheduler uses this parameter to trigger all missed schedules as soon as it recovers from downtime.
- Checking new schedules (**SCHEDULER_CHECKIN_INTERVAL**): This parameter defines the duration after which Job Scheduler checks for new schedules. The value for this parameter is set in seconds. The default value being 120 seconds (2 mins). You can configure this duration based on system/application requirements. It is recommended not to set the polling duration for less than 60 seconds (1 minute).

  > ### Note
  > The minimum value of the parameter is 5 seconds, the maximum is 214748.

- Make Job Scheduler configurable (**ENABLE_JOBSCHEDULER**): This parameter enables Job Scheduler to poll for new schedules or run any queries. Disabling this option will prevent Job Scheduler from creating or editing any new jobs or schedules.

With the release of **XS Advanced Job Scheduler version SP06 Patch 14 for XS SERVICES 1** or **1.6.15**, deployment of XS Advanced Job Scheduler can be configured to override the default behaviors.

For overriding the default behavior in case of previous deployments, you can upgrade the Job Scheduler component to **SP06 Patch 14 for XS SERVICES 1** or version **1.6.15** or higher.

> ⓘ Note
>
> Use the mtaext file for any Job Scheduler deployment, install, or update scenarios. In the absence of this file, existing values will be overwritten by default values.

Default values are overridden by changing specific properties in the mtaext file. A sample mtaext file with default behavior appears as below:

> {·} Sample Code

```
_schema-version: "2.0.0"
ID: com.sap.xs.jobscheduler
extends: com.sap.xs.jobscheduler
modules:
  - name: jobscheduler-service
    properties:
      ENABLE_MISFIRE: "true"
      ENABLE_JOBSCHEDULER: "true"
      SCHEDULER_CHECKIN_INTERVAL: 120
  - name: jobscheduler-broker
    properties:
      ENABLE_JOBSCHEDULER: "true"
```

Deploy jobscheduler with the mtaext file using the command

```
xs deploy  <Job_scheduler.mtar> -e <absolute-path-of-mtaext-file>
```

> ✤ Example

```
xs deploy MTA_archive.mtar -e myMTADeployExtension.mtaext
```

Job Scheduler Execution Type

Job Scheduler provides the following **types** of schedules for a job:

- Recurring Schedule
  Runs periodically at a specified time, dates, or interval. Recurring schedules can be created in the following ways:
  - The `repeatInterval` parameter:
    Defines the interval in human-readable text (for example, “2 minutes”), which can be used to set up a recurring schedule. The repeat interval defines the gap between each run of the schedule.
  - The `cron` parameter:
    Defines a `cron` expression (for example, `"cron": "* * * * * * *"`) used to represent a set of times, when the job is executed. The cron format referred here is the `xscron` format and not the publicly used Linux `cron`. For more details about the `xscron` format, see *The XS Job File* in *Related Information* below.
  - The `repeatAt` parameter:
    Defines the exact time, every day, when the job is executed.

- One-Time Schedule
  Runs only once at the specified time. One-time schedules can be created in the following ways:
  - Human-readable text string:
    A human-readable text string that defines the specific time for schedule execution (for example: “10 hours from now”, “3.30pm”, or “Friday at 2am”)
  - Using a `Date` object, with a pre-defined format, for example,
    `"startTime": {"date": "2015-10-20 4:30 +0000", "format": "YYYY-MM-DD HH:mm Z"}`
    The string is checked against both IETF-compliant RFC 2822 timestamps and ISO-8601

Job Scheduler Access

Job Scheduler can be accessed and used in the following ways during application development:

- APIs:
  The Job Scheduler service offers RESTfull and client specific APIs for Java and Node.js. The administrator **scope** is required to use the Job Scheduler API to maintain run time configurations for jobs and job schedules.
- User Interface:
  The *Job Scheduler Dashboard* is the tool used to manage the jobs and job schedules. Administrator authorization is required to maintain jobs and job schedules in the *Job Scheduler Dashboard*. For more information about permissions required to access the dashboard, see *The Job Scheduler Dashboard* topic.

> ### Note
>
> You can program actions in any programming language or platform. The runtime also supports jobs created in the SAP HANA XS classic version.

Related Information

Maintain Jobs and Job Schedules in XS Advanced [page 1443]
Job Scheduler REST API for XS Advanced [page 1448]
The Job Scheduler Dashboard [page 1467]
Scaling Applications in XS Advanced [page 1304]
The XS Job File [page 1240]

14.2.3.2.1 Maintain Jobs and Job Schedules in XS Advanced

Maintain run time configurations for jobs and job schedules in SAP HANA XS advanced.

Prerequisites

- The service broker and the service instance for the Job Scheduler service are available.
- The application using the Job Schedule is deployed in the space and bound to the Job Scheduler service instance.
- You have the authorization scope for `POST`, `PUT`, and `DELETE` requests (for example, *jobscheduler.Admin*).
- To access the *Job Scheduler Dashboard*, you must have the authorization scopes defined in the roles grouped together in one of the following role collections:
  - XS_CONTROLLER_ADMIN
    Full access: no access restrictions
  - XS_CONTROLLER_USER
    Modify and read-only access
  - XS_CONTROLLER_AUDITOR
    Read-only access

> → Tip
>
> Role collections can be assigned to an SAP HANA user in SAP HANA studio by means of user parameters, for example, `XS_RC_XS_CONTROLLER_ADMIN` or `XS_RC_XS_CONTROLLER_USER`, or `XS_RC_XS_CONTROLLER_AUDITOR`.

Context

To maintain jobs and job schedules, you use the Job Scheduler REST APIs (for example, *Job Creation*, *Job Configuration*, or *Job Deletion*) as illustrated in the following examples.

> ⓘ Note
>
> The code examples are not always complete; they are intended for illustration purposes only.

Procedure

1. Create a new job.
   Use the *Job Creation* API (`POST /scheduler/Jobs`), as illustrated in the example request:

```
POST /scheduler/jobs HTTP/1.1
Host: localhost:4242
Authorization: Basic YWJjOmRlZg==
```

```
Content-Type: application/json
Cache-Control: no-cache
{"name":"validateSalesOrder", "description": "cron job that validates
sales order requests", "action":"http://salesOrderApp.hana.acme.com:40023/
salesOrders/validate","active": true, "httpMethod":"PUT", "schedules":
[{"cron":"* * * * * * */10", "description": "this schedule runs every
10 seconds", "data":{"salesOrderId":"1234"}, "active": true, "startTime":
{"date": "2015-10-20 04:30 +0000", "format": "YYYY-MM-DD HH:mm Z"}}]}
```

The response to the job-creation request should look like the following example:

```
{"name":
"validateSalesOrder", "action":"http://salesOrderApp.hana.acme.com:40023/
salesOrders/validate","active":true,"httpMethod":"PUT","description":"cron
job that validates sales order
requests","startTime":null,"endTime":null,"signatureVersion":0,"schedules":
[{"active":true,"startTime":"2015-10-20
04:30:00","endTime":null,"description":"every 10 seconds, every
2 minutes","data":"{\"salesOrderId\":\"1234\"}","cron":"* *
* * * * */10","type":"recurring","scheduleId":"cb5c9def-
e2a0-4294-8a51-61e4db373f99"}],"_id":3}
Headers:
Connection → keep-alive
Content-Length → 468
Content-Type → application/json; charset=utf-8
Date → Mon, 09 Nov 2016 09:08:53 GMT
ETag → W/"1d4-P7BnAm3yordzbrYyJtpalg"
Location → /scheduler/jobs/3
X-Powered-By → Express
```

2. Modify (configure) a new job.
   Use the *Job Configuration* API (`PUT /scheduler/Jobs`), as illustrated in the example request:

```
PUT /scheduler/jobs/3 HTTP/1.1
Host: localhost:4242
Authorization: Basic YWJjOmRlZg==
Content-Type: application/json
Cache-Control: no-cache
{"active": true, "user":"abc", "password":"def", "httpMethod": "GET"}
```

   The response to the job-configuration request should look like the following example:

```
{"success":true}
Headers:
Connection → keep-alive
Content-Length → 16
Content-Type → application/json; charset=utf-8
Date → Mon, 09 Nov 2016 09:30:36 GMT
ETag → W/"10-c2PoX+nt7m8FOksxlYjAhg"
X-Powered-By → Express
```

3. Delete an existing job.
   Use the *Job Deletion* API (`DELETE /scheduler/Jobs`), as illustrated in the example request:

```
DELETE /scheduler/jobs/4 HTTP/1.1
Host: localhost:4242
Authorization: Basic YWJjOmRlZg==
Content-Type: application/json
Cache-Control: no-cache
```

   The response to the job-deletion request should look like the following example:

```
{"success":true}
Headers:
```

```
Connection → keep-alive
Content-Length → 16
Content-Type → application/json; charset=utf-8
Date → Mon, 09 Nov 2016 09:30:36 GMT
ETag → W/"10-c2PoX+nt7m8FOksxlYjAhg"
X-Powered-By → Express
```

4. Create a new job schedule.
   Use the *Job Schedule Creation* API (`POST /scheduler/jobs/3/schedules`), as illustrated in the example request:

   ```
   POST /scheduler/jobs/3/schedules HTTP/1.1
   Host: localhost:4242
   Authorization: Basic YWJjOmRlZg==
   Content-Type: application/json
   Cache-Control: no-cache
   {"repeatEvery":"2 hours","data":{"order_id":"abcd"}, "active":true,
   "description":"New Schedule", "startTime":{"date": "2016-04-21", "format":
   "YYYY-MM-DD"}}
   ```

   The response to the job-schedule creation request should look like the following example:

   ```
   "repeatInterval":"2
   hours","repeatAt":null,"time":null,"cron":null,"data":"{\"order_id\":\"abcd\"}
   ","description":"New
   Schedule","type":"recurring","active":true,"startTime":"2016-04-21
   18:30:00","endTime":null,"jobId":3,"scheduleId":"0e29c67c-563e-4931-
   af08-43acb10813e8"}
   Headers:
   Connection → keep-alive
   Content-Length → 274
   Content-Type → application/json; charset=utf-8
   Date → Mon, 09 Nov 2016 09:42:13 GMT
   ETag → W/"112-rdQSXHBVY0u6JNI/Wf0I7w"
   Location → /scheduler/jobs/3/schedules/0e29c67c-563e-4931-af08-43acb10813e8
   X-Powered-By → Express
   ```

5. Delete an existing job schedule.
   Use the *Job Schedule Deletion* API (`DELETE /scheduler/jobs/3/schedules`), as illustrated in the example request:

   ```
   DELETE /scheduler/jobs/4 HTTP/1.1
   Host: localhost:4242
   Authorization: Basic YWJjOmRlZg==
   Content-Type: application/json
   Cache-Control: no-cache
   ```

   The response to the job-schedule deletion request should look like the following example:

   ```
   {"success":true}
   Headers:
   Connection → keep-alive
   Content-Length → 16
   Content-Type → application/json; charset=utf-8
   Date → Mon, 09 Nov 2016 09:51:39 GMT
   ETag → W/"10-c2PoX+nt7m8FOksxlYjAhg"
   X-Powered-By → Express
   ```

Audit Logging in Job Scheduler

> **Note**
>
> As of version 1.7.32 of the Job Scheduler, the Job Scheduler REST interface has an automatically enabled audit logging that captures the below table-listed events.

For more information about the prerequisites and Audit Log service, see: Audit Log Storage Location.

Event	Event action	Key	Prev value	Current value	XSA Object
Update job configuration	CONFIGURATION CHANGE	JOBSCHEDULER-REST	{"httpMethod":"POST","id":"2"}	{"httpMethod":"GET","id":"2"}	{"type":"JOB CONFIGURATION","id":{"name":"JOBSCHEDULER-REST","configuration":"UPDATE_JOB_CONFIG"}}
Update global configuration	CONFIGURATION CHANGE	JOBSCHEDULER-REST	{"ASYNC_EXEC_TIME-OUT":"1801","MAX_JOB_FAILURE_COUNT":"2"}	{"ASYNC_EXEC_TIME-OUT":"1800","MAX_JOB_FAILURE_COUNT":"3"}	{"type":"GLOBAL CONFIGURATION","id":{"name":"JOBSCHEDULER-REST","configuration":"UPDATE_GLOBAL_CONFIG"}}
Update schedule configuration	CONFIGURATION CHANGE	JOBSCHEDULER-REST	{"description":"New Schedule","id":"eb3801d6-1809-4119-a51b-84c97cd42e30"}	{"description":"Edited Schedule","id":"eb3801d6-1809-4119-a51b-84c97cd42e30"}	{"type":"SCHEDULE CONFIGURATION","id":{"name":"JOBSCHEDULER-REST","configuration":"UPDATE_SCHEDULE_CONFIG"}}

Event	Event action	Key	Statement String
Create job	SECURITY EVENT	JOBSCHEDULER-REST	jobscheduler-auditlog - Successful: JobScheduleCreation - Job: {"jobId":59,"scheduleId":"3477568e-2561-4b0c-99b6-48e88228559c","scheduleDescription":"New Schedule"} - Info: {"correlationId":"7514dd5f-81c1-4bfb-bdf1-2b96ac36efdc","userId":"SBSS_4695289062304037692191520245614357878114736070417864190544415880 6","serviceInstance":"d46da331-c4a8-47d7-95f4-ff4fe315b3ef"}
Delete job	SECURITY EVENT	JOBSCHEDULER-REST	jobscheduler-auditlog - Successful: JobDeletion - Job: {"jobId":"59"} - Info: {"correlationId":"3aa7bfe4-d3fa-4803-a920-da0d0bf0778d","userId":"XSA_ADMIN","serviceInstance":"d46da331-c4a8-47d7-95f4-ff4fe315b3ef"}
Delete schedule	SECURITY EVENT	JOBSCHEDULER-REST	jobscheduler-auditlog - Successful: JobScheduleDeletion - Job: {"jobId":"4","scheduleId":"155c7ea7-cca2-4a98-8818-f13b14d99e04"} - Info: {"correlationId":"7fa0c9f4-f3aa-4d9c-86f5-2f138a19a807","userId":"XSA_ADMIN","serviceInstance":"d46da331-c4a8-47d7-95f4-ff4fe315b3ef"}
Activate schedules	SECURITY EVENT	JOBSCHEDULER-REST	jobscheduler-auditlog - Successful: JobScheduleActivation - Job: {"jobId":"59","activationStatus":true} - Info: {"correlationId":"3a71cddf-ed71-4dfe-913d-aec64db0b5dd","userId":"SBSS_4695289062304037692191520245614357878114736070417864190544415880 6","serviceInstance":"d46da331-c4a8-47d7-95f4-ff4fe315b3ef"}

Event	Event action	Key	Statement String
Create schedule	SECURITY EVENT	JOBSCHEDULER-REST	jobscheduler-auditlog - Successful: JobScheduleCreation - Job: {"jobId":59,"scheduleId":"3477568e-2561-4b0c-99b6-48e88228559c","scheduleDescription":"New Schedule"} - Info: {"correlationId":"7514dd5f-81c1-4bfb-bdf1-2b96ac36efdc","userId":"SBSS_4695289062304037692191520245614357878114736070417864190544415880 6","serviceInstance":"d46da331-c4a8-47d7-95f4-ff4fe315b3ef"}
Create schedule - fail	SECURITY EVENT	JOBSCHDULER-REST	jobscheduler-auditlog - Failed: JobScheduleDeletion - Job: {"jobId":"59","scheduleId":"3477568e-2561-4b0c-99b6-48e88228559"} - Info: {"correlationId":"6f8ed7b5-ee4f-42a2-ab6c-d41f09dfd74a","userId":"SBSS_46952890623040376921915202456143578781147360704178641905444158806","serviceInstance":"d46da331-c4a8-47d7-95f4-ff4fe315b3ef"}

Related Information

Job Scheduler REST API for XS Advanced [page 1448]
The Job Scheduler Dashboard [page 1467]
Scheduling Jobs in XS Advanced [page 1439]

14.2.3.2.1.1 Job Scheduler REST API for XS Advanced

The Job Scheduler APIs enable applications to use the functionality provided in Job Scheduler.

The Job Scheduler-as-a-Service is a microservice component, which enables you to create, schedule, and run application tasks. The component exposes REST endpoints for interaction, with JSON as the format for data communication. The Job Scheduler API for SAP HANA XS advanced includes the commands listed in the following table. For more information about the configuration parameters required for the request, see the API documentation provided with the *Job Scheduler Dashboard* tool.

> **Note**
>
> Access to the APIs is controlled by authorization scopes, for example, `admin` for `POST` and `PUT` requests, or `view` for `GET` requests. Scopes are built into roles, which can be assigned to users in role collections. The Job Scheduler REST APIs are protected with basic authentication.

An application, which has been bound to the Job Scheduler service and wants to interact with the Job Scheduler service, must extract the authentication credentials from the `<VCAP_SERVICES>` environment variable and use these credentials to call the REST APIs. To invoke the API, the user-authentication credentials must be encoded and passed in the "Authorization" header. If the credentials are not passed or they are passed wrongly, the APIs return a response with the status code "`401- Unauthorized`".

In this section, you can find information about the following topics:

- Command Overview
- Human-Readable Dates
- Time Formats

Command Overview

XS Advanced Job Scheduler REST API

API	Description	Required Scope
Job Creation	Used to create a job. Job creation can accept a collection of job schedules to be created.	`admin`
Job Configuration	Configure a job with updated run time information. The API can also be used to create a job if a Job with the Job Name in the URI segment, is not found.	`admin`
Job Deletion	Delete a job and purge all its run time information such as job schedules and logs.	`admin`
Job Schedule Creation	Create a job schedule for a specified job. All job configuration values (Action URL, HTTP Method, User, Password & Job Activation Status) are valid for the newly created schedule. A job schedule will only run if both the job and the schedule are active.	`admin`
Job Schedule Modification	Configure the run time information of a job schedule for a specified job. All job configuration values (for example: `Action URL`, `HTTP Method`, `User`, `Password`, and `Job Activation Status`) remain valid for the modified schedule.	`admin`
Job Schedule Deletion	Delete and purge run time information of the job schedule of the specified job. All related information like job schedule configurations and logs are purged. The processing of the schedule is also immediately stopped.	`admin`

API	Description	Required Scope
Bulk Job Schedule Activation Bulk Job Schedule Deactivation	This is a utility API used to activate or deactivate all existing schedules of a job. This API triggers the immediate processing (or a halt in processing) of all job schedules for the specified job.	`admin`
Job Details	Retrieve the saved details and configurations of a specified job. If the `displaySchedules` parameter is not provided, the schedules for the job are not returned and only the job details are returned.	`view`
Job Schedule Details	Retrieve the saved details and configurations of a specified job schedule & optionally the generated logs for the schedule.	`view`
Bulk Job Schedule Deletion	Delete and purge run time information of all the currently configured job schedules of the specified job. All related information like job schedule configurations and logs are purged. The processing of the schedules is also immediately stopped.	`admin`
Job Run Log Update	Used by applications, to inform the Job Scheduler about the status of an asynchronous, long-running job run.	`admin`

Job Creation

To create a job schedule, at least one of the fields `repeatAt`, `repeatEvery`, `cron` and `time` must be used. The response from the job creation API is a JSON body with the job details, including the ID of the job.

- **Route**
  `POST /scheduler/jobs`
- **Response**
  A JSON body containing the job details, including the ID of the job with status code "`201-CREATED`" , if the call was successful. A location header with the relative path to the job-details is included in the response.

Sample Code

```
{
  "name": "validateSalesOrder",
  "description": "cron job that validates sales order requests",
  "action": "http://salesOrderApp.hana.ondemand.com:40023/salesOrders/
validate",
  "active": true,
  "httpMethod": "PUT",
  "schedules": [
    {
      "cron": "* * * * * */10 0",
      "description": "this schedule runs every 10 minutes",
      "data": {
        "salesOrderId": "1234"
      },
      "active": true,
      "startTime": {
        "date": "2015-10-20 04:30 +0000",
        "format": "YYYY-MM-DD HH:mm Z"
```

```
      }
    }
  ]
}
```

```
Response:
{
  "name": "validateSalesOrder",
  "action": "http://<application-url>/action",
  "active": true,
  "httpMethod": "PUT",
  "description": "cron job that validates sales order requests",
  "startTime": null,
  "endTime": null,
  "signatureVersion": 0,
  "schedules": [
    {
      "active": true,
      "startTime": "2015-10-20 04:30:00",
      "endTime": null,
      "description": "every 10 seconds, every 2 minutes",
      "data": "{\"salesOrderId\":\"1234\"}",
      "cron": "* * * * * * */10",
      "type": "recurring",
      "scheduleId": "schedule ID details"
    }
  ],
  "_id": 3
}
```

The job schedule creation request is defined with the parameters listed in the following table:

> **Note**
>
> Parameters marked with an asterisk (*) are mandatory.

Job Creation: Request Body Fields

Request Field	Type	Description
`name` *	String	The unique name of the job to be created **Note** If a job with the same name for the technical user credentials already exists, the job creation request fails.
`description`	String	Describes the user-defined job
`action` *	String	The fully qualified URL endpoint to be called when the job runs, for example: `http://host.acme.com/app/call`

Request Field	Type	Description
`active`	Boolean	Defines if the job should be activated on creation. Allowed values are: • `false` (default) The job is in inactive mode on creation • `true` The job is activated on creation
`httpMethod`	String	The HTTP method to be used to call the end-point URL for the job action . Allowed values are: GET, POST (default), PUT, and DELETE
`startTime`	Object	The start time for the job. If the start time is specified for the job, the scheduler checks if a start time is provided for the schedule as well. If a start time is provided for the schedule, it is used for determining the start of the schedule run. If no job-schedule start time is defined, the start time for the job is used. The date and time-formats must be specified as strings.
`endTime`	Object	The end time for the job. If the end time is specified for a job, the scheduler checks if an end time is provided for the schedule as well. If an end time is provided for the schedule, it is used for determining the end of the schedule run If not, the end time for the job is used. The date and time-formats must be specified as strings.
`schedules` *	Array	The array of job schedule objects, to be created on job creation.

The `schedules` parameter can be used to provide details of the job schedule (as properties of each job schedule object); the following table lists the permitted properties:

Schedule Parameter Fields

Schedule Field	Type	Description
`data`	object	Optional data to be passed to the job action endpoint when invoked. Typically, the custom data is sent based on the HTTP method configured for invoking the end point URL,for example: `{"dataParam": "somevalue"}`
`time`	string or object	For one-time schedules, the parameter denoting the time at which the task executes. A human-readable text can be used to specify the time, for example, "`3.30pm`" or "`tomorrow at 2am`". If an object is used, the date and time-formats must be specified as strings.
`repeatInterval`	string	For recurring schedules, the parameter denoting the intervals when the schedule should run. The parameter supports the use of human readable formats.
`repeatAt`	string	For recurring schedules, the parameter denoting the exact time when the job schedule must run. A human-readable text can be used to denote a specified time, for example, "`3.30pm`" or "`tomorrow at 2am`", if the schedule runs repeatedly.

Schedule Field	Type	Description
`cron`	string	For recurring schedules, the parameter denoting the `xscron` pattern. For more information, see *Cron Parameter* section
`startTime`	object	The time when the job scheduling should start. The date and time-formats must be specified as strings.
`endTime`	object	The time when the job scheduling should end. The date and time-formats must be specified as strings
`description`	string	The user-provided description of the job schedule

Job Configuration

Configure a job with updated run time information. The API can also be used to create a job if a Job with the Job Name in the URI segment, is not found. If the API is being used to create a job, the parameters must conform to the same constraints as provided in the Job Creation API

- **Route**
  `PUT /scheduler/jobs/:jobId`
  `PUT /scheduler/jobs/:jobName`
  "`:jobId`" is the ID of the job previously created using the Job Creation API. If the job name is used in the URI, it is first checked if the job with the name, exists. If no such named job exists, the API tries to create the job. If it does exist, the API configures the job with the details provided in the request body.

  > ⓘ Note
  >
  > If the API is used to create a job, care must be taken to ensure that the job name in the request URI matches the name of the job in the request body. If the names do not match, an error is returned.

- **Response**
  If the API finds an existing job, the response has a status code of "`200-OK`", if the call was successful. The response has a status code of "`201-CREATED`", if the API is used to create a new job; for new jobs, a location header containing the relative path to the job-details is returned in the response.

> {·} Sample Code

```
PUT /schedule/jobs/5 HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
content-length: 500
{"active": true, "user":"abc", "password":"def", "httpMethod": "GET"}
```

```
Response:
status: 200 OK
content-type: application/json; charset=utf-8
{"success": true}
```

Sample Code

```
PUT /schedule/jobs/jobwhichdoesnotexist HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
content-length: 500
{"name":"jobwhichdoesnotexist", "jobDescription": "greets
the world periodically", "action":"http://httpbin.org/basic-auth/abc/
def","active":true, "httpMethod":"GET", "schedules": [{"repeatEvery":"2
minutes", "scheduleDescription": "every 2 minutes, run this schedule", "data":
{"time":"abc"}, "active": true}, {"cron":"* * * * *", "scheduleDescription":
"every 4 minutes, run this schedule", "data":{"time":"abc"}, "active":
false}]}
```

```
Response:
status: 201 CREATED
content-type: application/json; charset=utf-8
{"_id":120,"name":"jobwhichdoesnotexist","description":"","action":"http://
httpbin.org/basic-auth/abc/
def","active":true,"user":null,"httpMethod":"GET","schedules":
[{"scheduleId":"b373469c-c6d4-4d5f-a002-c56f18455dc5","description":"Default
Schedule","data":
{"time":"abc"},"type":"recurring","active":true,"startTime":null,"endTime":nul
l,"repeatInterval":"2 minutes"},{"scheduleId":"2f98471c-26de-4293-ae53-
e4a16e1513f5","description":"Default Schedule","data":
{"time":"abc"},"type":"recurring","active":false,"startTime":null,"endTime":nu
ll,"cron":"* * * * *"}]}
```

The job schedule configuration request is defined with the parameters listed in the following table:

Job Creation: Request Body Fields

Request Field	Type	Description
`active`	Boolean	Defines if the job should be activated on configuration. Allowed values are: • `false` (default) The job is in inactive mode when configured • `true` The job is active when configured
`user`	String	The name of the user account to run the configured job
`password`	String	The password for the user account to run the configured job
`httpMethod`	String	The HTTP method to be used to call the end-point URL for the job action . Allowed values are: `GET`, `POST` (default), `PUT`, and `DELETE`
`startTime`	Object	The start time for the job. If the start time is specified for the job, the scheduler checks if a start time is provided for the schedule as well. If a start time is provided for the schedule, it is used for determining the start of the schedule run. If no job-schedule start time is defined, the start time for the job is used. The date and time-formats must be specified as strings.

Request Field	Type	Description
`endTime`	Object	The end time for the job. If the end time is specified for a job, the scheduler checks if an end time is provided for the schedule as well. If an end time is provided for the schedule, it is used for determining the end of the schedule run If not, the end time for the job is used. The date and time-formats must be specified as strings.

Job Deletion

Delete a job and purge all its run time information such as job schedules and logs.

- **Route**
  `DELETE /scheduler/jobs/:jobId`
- **Response**
  If the call is successful, the response has a status code "`200-OK`" and includes a JSON response `{"success": true}`.

Sample Code

```
DELETE /schedule/jobs/:jobId HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
```

```
Response: Status: 200 OK
Content-Type: application/json;charset=utf-8
{"success":true}
```

Job Schedule Creation

Create a job schedule for a specified job. All job configuration values (Action URL, HTTP Method, User, Password & Job Activation Status) are valid for the newly created schedule. A job schedule will only run if both the job and the schedule are active.

- **Route**
  `POST /scheduler/jobs/:jobId/schedules`
- **Response**
  If the call is successful, the response has a status code of "`201-CREATED`". A location header with the relative path to the schedule-details, is returned in the response.

Sample Code

```
POST /scheduler/jobs/3/schedules
{
    "repeatInterval":"2 hours",
    "active":true,
    "description":"New Schedule",
```

```
    "startTime": {
        "date": "2017-08-21",
        "format": "YYYY-MM-DD"
    }
}
```

```
Response:
{
  "repeatInterval": "2 hours",
  "repeatAt": null,
  "time": null,
  "cron": null,
  "data": "{\"order_id\":\"abcd\"}",
  "description": "New Schedule",
  "type": "recurring",
  "active": true,
  "startTime": "2015-04-20 18:30:00",
  "endTime": null,
  "jobId": 3,
  "scheduleId": "<schedule ID details>"
}
```

Job Schedule Creation Parameters

Request Field	Type	Description
`time`	string or object	For one-time schedules, the parameter denoting the time at which the task executes. A human-readable text can be used to specify the time, for example, "`3.30pm`" or "`tomorrow at 2am`". If an object is used, the date and time-formats must be specified as strings.
`repeatInterval`	string	For recurring schedules, the parameter denoting the interval when the schedule should run. The parameter supports the use of human readable formats.
`repeatAt`	string	For recurring schedules, the parameter denoting the exact time when the job schedule must run. A human-readable text can be used to denote a specified time, for example, "`3.30pm`" or "`tomorrow at 2am`", if the schedule runs repeatedly.
`cron`	string	For recurring schedules, the parameter denoting the `xscron` pattern. For more information, see *Cron Parameter* section.
`data`	object	The parameter denoting optional data to be passed to the job action endpoint when invoked. Typically, the custom data is sent based on the HTTP method configured for invoking the end point URL,for example: `{"dataParam": "somevalue"}`
`startTime`	object	The time when the job scheduling should start. The date and time-formats must be specified as strings.
`endTime`	object	The time when the job scheduling should end. The date and time-formats must be specified as strings

Request Field	Type	Description
`active`	Boolean	Defines if the job should be activated on configuration. Allowed values are: • `false` (default) The job is in inactive mode when configured • `true` The job is active when configured
`description`	string	The user-provided description of the job schedule

Job Schedule Modification

Configure the run time information of a job schedule for a specified job. All job configuration values (for example: `Action URL`, `HTTP Method`, `User`, `Password`, and `Job Activation Status`) remain valid for the modified schedule.

- **Route**
  `PUT /scheduler/jobs/:jobId/schedules/:scheduleId`
- **Response**
  If the call is successful, the response has a status code of `200- OK`.

Calling this API stops further scheduling of the previously configured job schedule and, if activated, the processing for the newly configured schedule is started. This API cannot be used to change the scheduling mode for the job schedule. For example, if the schedule was created as a recurring “cron”-type schedule, it cannot be changed to a “repeatEvery”-type schedule. However, existing schedule values can be changed.

Sample Code

```
PUT /schedule/jobs/:jobId/schedules/:scheduleId HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
content-length: 500
{"description": "Edited Schedule", "startTime": {"date": "2013-02-08
09:30:26.123"}, "endTime": {"date": "2015-06-08 09:30:26.123"}, "active":
true, "cron": "* * * * *" }
```

```
Response:
Status: 200 OK
Content-Type: application/json; charset=utf-8
{"scheduleId":"80e23846-734e-4b4b-a130-159a492ec482","name":"greet the
world3","data":
{"time":"abc"},"type":"recurring","priority":0,"action":"http://httpbin.org/
basic-auth/abc/
def","nextRunAt":"2015-04-23T03:58:21.358Z","startTime":"2013-02-08T04:00:26.1
23Z","endTime":"2015-06-08T04:00:26.123Z","active":true,"description":"Edited
Schedule","jobId":"136","cron":"* * * * *"}
```

Job Schedule Modification Parameters

Request Field	Type	Description
`time`	string or object	For one-time schedules, the parameter denoting the time at which the task executes. A human-readable text can be used to specify the time, for example, "`3.30pm`" or "`tomorrow at 2am`". If an object is used, the date and time-formats must be specified as strings.
`repeatInterval`	string	For recurring schedules, the parameter denoting the interval when the schedule should run. The parameter supports the use of human readable formats.
`repeatAt`	string	For recurring schedules, the parameter denoting the exact time when the job schedule must run. A human-readable text can be used to denote a specified time, for example, "`3.30pm`" or "`tomorrow at 2am`", if the schedule runs repeatedly.
`cron`	string	For recurring schedules, the parameter denoting the `xscron` pattern. For more information, see *Cron Parameter* section.
`data`	object	The parameter denoting optional data to be passed to the job action endpoint when invoked. Typically, the custom data is sent based on the HTTP method configured for invoking the end point URL,for example: `{"dataParam": "somevalue"}`
`startTime`	object	The time when the job scheduling should start. The date and time-formats must be specified as strings.
`endTime`	object	The time when the job scheduling should end. The date and time-formats must be specified as strings
`active`	Boolean	Defines if the job should be activated on configuration. Allowed values are: • `false` (default) The job is in inactive mode when configured • `true` The job is active when configured
`description`	string	The user-provided description of the job schedule

Job Schedule Deletion

Delete and purge run time information of the job schedule of the specified job. All related information like job schedule configurations and logs are purged. The processing of the schedule is also immediately stopped.

> ⚠ Caution
>
> This API removes all the run time configuration information of the job schedule, irrespective of whether the schedule is active or not.

- **Route**
  `DELETE /scheduler/jobs/:jobId/schedules/:scheduleId`
- **Response**
  If the call is successful, the response has a status code, "`200-OK`" and includes a JSON response `{"success": true}`.

Sample Code

```
DELETE /scheduler/jobs/:jobId/schedules/:scheduleId HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
```

```
Response: {"success": true}
Status Code: 200 OK
```

Bulk Job Schedule Activation/Deactivation

This is a utility API used to activate or deactivate all existing schedules of a job.

- **Route**
  `POST /scheduler/jobs/:jobId/schedules/activationStatus`
- **Response**
  If the call is successful, the response has a status code, "`200-OK`" and includes a JSON response `{"success": true}`.

Sample Code

```
POST /scheduler/jobs/:jobId/schedules/activationStatus HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
{"activationStatus": true}
```

```
Response: {"success": true}
Status Code: 200 OK
```

Bulk Job Schedule Activation Parameters

Request Field	Type	Description
`activationStatus`	Boolean	The desired activation status of the job schedules for the job. Allowed values for the activation status are: • `false` (default) All job schedules for the specified job should be **de**activated • `true` All job schedules for the specified job should be activated

Job Details

Retrieve the saved details and configurations of a specified job.

- **Route**
  `GET /scheduler/jobs/:jobId?displaySchedules=true`
  `GET /scheduler/jobs?jobId=:jobId&displaySchedules=true Route`
  `GET /scheduler/jobs?name=:jobName&displaySchedules=true`
- **Response**
  If the call is successful, the response has a status code, "`200-OK`"
  and includes a JSON response with the schedule details, for example:
  `{"schedules":[{"data":{"time":"abc"},"type":"recurring","repeatInterval":"2 minutes","active":false,"startTime":null,"endTime":null,"repeatAt":null, [...]}`.

Sample Code

```
GET /scheduler/jobs/:jobId?displaySchedules=true HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
```

```
Response:
Status: 200 OK
Content-type: application/json;charset=utf-8
{"schedules":[{"data":{"time":"abc"},"type":"recurring","repeatInterval":"2
minutes","active":false,"startTime":null,"endTime":null,"repeatAt":null,"sched
uleId":"0d3b4cc1-0f7b-4ee6-ab12-63d474b900f2","description":"Default
Schedule"},{"data":{"time":"abc"},"type":"recurring","cron":"* * * *
*","active":false,"startTime":null,"endTime":null,"repeatAt":null,"scheduleId"
:"1b1bb70f-cada-46c9-9974-a7a1b87ba24f","description":"Default
Schedule"}],"name":"greet the world2","description":"","action":"http://
httpbin.org/basic-auth/abc/
def","user":null,"httpMethod":"GET","active":false,"_id":111}
```

Job Details Parameters

Request Field	Type	Description
`displaySchedules`	Boolean	Display details of the job schedules for the job. Allowed values for the job details are: • `false` Do **not** display details of job schedules for the specified job • `true` Display details of job schedules for the specified job
`jobId`	String	The job ID needed to query for the job details. This can be passed as a URI segment parameter or as a query parameter.
`name`	String	The job name needed to query the job details. This can be passed as a query parameter

Job Schedule Details

Retrieve the saved details and configurations of a specified job schedule & optionally the generated logs for the schedule. Either `:jobId` or `:name` is required to invoke this API. If `displayLogs` is not provided, the logs for the schedule are not returned and only the schedule details are returned.

- **Route**
  `GET /scheduler/jobs/:jobId/schedules/:scheduleId?displayLogs=true`
- **Response**
  If the call is successful, the response has a status code, "`200-OK`" and includes a JSON response with the schedule details, for example: `{"data":{"time":"abc"},"type":"recurring","repeatInterval":"2 minutes","plannedTime":"2015-04-19T15:12:44.000Z","active":true,"startTime":null,"endTime":null,"nextRunAt": "2017-08-11 10:00:00","repeatAt":null, [...]}`.

Sample Code

```
GET /scheduler/jobs/112/schedules/550d1b96-8002-4d0d-850e-368aaa591671?
displayLogs=true
HTTP/1.1 content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
```

```
Response:
Status: 200 OK
Content-Type: application/json; charset=utf-8
{"data":{"time":"abc"},"type":"recurring","repeatInterval":"2
minutes","plannedTime":"2015-04-19T15:12:44.000Z","active":true,"startTime":nu
ll,"endTime":null,"nextRunAt": "2017-08-11 10:00:00","repeatAt":null,"logs":
[{"text":null,"httpStatus":null,"executionTime":null,"status":"SCHEDULED","sch
eduleTime":"2015-04-19T15:10:53.000Z","completionTime":null}],"scheduleId":"55
0d1b96-8002-4d0d-850e-368aaa591671","description":"Default Schedule"}
```

Job Schedule Details Parameters

Request Parameter	Type	Description
`displayLogs`	Boolean	Controls whether the API should return (`true`) all the generated logs for the job schedule or not (`false`)

Bulk Job Schedule Deletion

Delete and purge run time information of all the currently configured job schedules of the specified job. All related information like job schedule configurations and logs are purged. The processing of the schedules is also immediately stopped.

⚠ Caution

This API removes all the run time configuration information of the job schedule, irrespective of whether the schedule is active or not.

- **Route**
  `DELETE /scheduler/jobs/:jobId/schedules`
- **Response**
  If the call is successful, the response has a status code, "`200-OK`" and includes a JSON response `{"success": true}`.

Sample Code

```
DELETE /scheduler/jobs/:jobId/schedules HTTP/1.1
content-type:application/json;charset=utf-8
host:https://scheduler.service.acme.com
```

```
Response: {"success": true}
Status Code: 200 OK
```

Job Run Log Update

Inform the Job Scheduler about the status of an asynchronous, long-running job run. This API must be invoked by the application after the asynchronous execution of the job has completed, with the status of the job run and optionally some text about the job execution.

Caution

This API must be invoked by the application after the **asynchronous** execution of the job has completed, with the status of the job run and optionally some text about the job execution.

- **Route**
  `PUT /scheduler/jobs/:jobId/schedules/:scheduleId/runs/:runId`

Note

Parameters marked with an asterisk (*) are mandatory.

Job Run Log Update Parameters

Request Parameter	Type	Description
`success` *	Boolean	Indicates that the job run was successful (`true`) or failed (`false`)
`message`	String	Additional log/text about the job run

Human Readable Dates

The job scheduler for XS advanced supports human readable dates and ranges for the parameters `time`, `repeatAt` and `repeatEvery`, which are used for configuring job schedules. The job scheduler uses an

embedded English language date parser for this facility. Valid human readable strings for the parameters are shown below:

> **Note**
>
> The date parser expects a valid readable string; invalid strings will either throw parser errors or cause the job scheduling to happen inconsistently.

Date and Time Parameters

Parameter	Comments	Examples
`time`	Designates a particular timestamp for running a job schedule. If an invalid string is provided, the scheduler falls back to the current timestamp and runs the schedule immediately. The following example strings are valid for the `time` parameter:	“10 hours from now” “20 minutes from now” “in 2 hours” “tomorrow at 4pm” “next week monday at 5am” “9pm tonight” “3.30pm”
`repeatAt`	Represents a convenient way to create daily timestamp-based schedules. The string should designate a particular timestamp for repeatedly running a job schedule. This follows the same pattern as the recommendations for the "time" parameter, barring a few discrepancies. While the text for the "time" parameter must denote something concrete and in the future, the 'repeatAt' must designate a timestamp, which is valid and constant daily. If an invalid string is used, the scheduler falls back to the current timestamp and runs the schedule immediately. **Note** Second-based precision can sometimes be inaccurately timed; timezones must be specified using the offset (in hours), for example, “+07:00”	“4.40pm” “18.40” “6.20am” “17.20:30 ”
`repeatInterval`	The string should designate a interval to repeat the job execution. Word strings for denoting the numeric value are not supported yet. For example, for “twenty minutes”, use “20 minutes” to denote the interval. Supported time-units for this parameter are "years", "months", "weeks", "days", "hours", "minutes", "seconds". **Note** The minimum value for the `repeatInterval` option is 5 seconds.	“10 seconds” “5 minutes” “2 hours” “9 days” “7 weeks” “1 month” (assumes 30 days) “3 years” (assumes 365 days)

Cron Parameter

The `cron` parameter enables defining the schedule at which a job runs. It uses xscron syntax, which is different from the standard Linux cron but provides more functional flexibility for scheduling application-based jobs.

The following table explains the order of items used in the xscron expression and lists the permitted values for each item:

Cron Item (from left to right)	Meaning and Permitted Value
`Year`	4-digit representation for year (for example, 2025).
`Month`	Numeric representation for a month. Permitted values are 1 to 12.
`Day`	Numeric representation for the day of the month. Permitted values are from -31 to 31. (Negative values are calculated from the end of the month).
`Day of Week`	3-letter representation for day of week. Permitted values are `mon, tue, wed, thu, fri, sat` and `sun`.
`Hour`	Numeric representation for an hour. Permitted values are 0 to 23.
`Minute`	Numeric representation for a minute. Permitted values are 0 to 59.
`Second`	Numeric representation for a second. Permitted values are 0 to 59.

XSCron Syntax

The following table illustrates the item syntax allowed to define xscron expressions:

Expression	Where-used	Value
`*`	Anywhere	Any value
`*/a`	Anywhere	Any a-th value
`a:b`	Anywhere	Values in range a to b
`a:b/c`	Anywhere	Every c-th value between a and b
`a.y`	DayOfWeek	On the a-th occurrence of the weekday y (a = -5 to 5)
`a,b,c`	Anywhere	a or b or c

Examples

The table below displays the possible xscron expression with their descriptions:

Expression	Description
`* * * * 10:12 */30 0`	Run the schedule every day between 10:00 and 12:00, every 30 minutes. The schedule is executed at 10:00, 10:30, 11:00, 11:30, 12:00, 12:30, and the next day it starts again at 10:00.

Expression	Description
`2025 * * mon 12 0 0`	Run the schedule every Monday in 2025 at 12:00.
`* * 3:-2 * 12:14 0 0`	Run every hour between 12:00 and 14:00 every day between the third and second-to-last day of the month.
`* * * -1.sun 9 0 0`	Run the schedule on the last Sunday of every month at 09:00.
`* */3 10 * 9 0 0`	Run the schedule every 3 months on the 10th day at 9:00.
`* 3, 6, 9, 12 10 * 12 0 0`	Run the schedule on March, June, September and December on the 10th day at 12:00.

Date and Time Formats in Job Schedule Parameters

The date-time parameters for job schedules (for example, `startTime`, `endTime`, and `time`) can be passed as objects , with the mandatory `date` field denoting the date as a string and an optional `format` field denoting a date-time format for correctly parsing the user-provided date value. If the parameters are passed as strings, they must be valid date representations, in either the ISO-8601 or IETF-compliant RFC 2822 formats. For object representations, the following rules apply:

- **Date field as input**
  If only the date field is provided as input, the string is checked against both IETF-compliant RFC 2822 time stamps and ISO-8601. If the date string is of an unknown format, the parser displays an error. For ISO-8601 compliant dates, calendar dates (for example, "2013-02-08"), week dates ("2013-W06-5"), ordinal dates ("2013-039") and time-based dates ("2013-02-08 09+07:00") are all supported.
- **Date string format**
  If the format of the date string is customized, an optional format string can be passed. The allowed parsing tokens are as described in the following table:

Date and Time Parameters

Input Token	Example	Description
YYYY	2014	4 digit year
YY	14	2 digit year
Q	1-4	Quarter of year. Sets month to first month in quarter
M MM	1-12	Month number
MMM MMMM	January- Dec	Month name in locale
D DD	1- 31	Day of month
Do	1st- 31st	Day of month with ordinal
DDD DDDD	1-365	Day of year

Input Token	Example	Description
X	1410715640.579	Unix Timestamp
x	1410715640579	Unix Timestamp (ms)
gggg	2015	Locale 4 digit week year
gg	15	Locale 2 digit week year
w ww	1- 53	Locale week of year
e	1-7	Locale day of week
GGGG	2015	ISO 4-digit week year
GG	15	ISO 2-digit week year
W WW	1- 53	ISO week of year
E	1-7	ISO day of week
H HH	0 -23	24 Hour Time
h hh	1 -12	12 hour time used with 'a A'
a A	am pm	Post or ante meridiem
m mm	0 -59	Minutes
s ss	0 -59	Seconds
S	0 -9	Tenths of a second
SS	0 -99	Hundredths of a second
SSS	0 -999	Thousandths of a second
Z ZZ	+12:00	Offset from UTC as +-HH:mm, +-HHmm, or Z

Date-Time Format Examples

- `startTime`
  `"startTime": {"date": "2015-10-20 4:30 +0000", "format": "YYYY-MM-DD HH:mm Z"}`
  4.30 UTC on 20th Oct 2015
- `endTime`
  `"endTime": {"date": "2015-W06-5"}`
  Friday, February 06, 2015
- `time`
  `"time": {"date": "2010-10-20 4:30", "format": "YYYY-MM-DD HH:mm"}`
  4.30 Local Time (the timezone for the scheduler service is considered here)

Related Information

Scheduling Jobs in XS Advanced

14.2.3.2.1.2 The Job Scheduler Dashboard

The Job Scheduler dashboard enables you to manage job schedules for a service instance.

The dashboard lists the available jobs. Select a job to create a schedule or to view existing schedules.

How to access the Job Scheduler dashboard

1. **Get the Dashboard URL**
   To access the Job Scheduler dashboard, you can connect remotely using secure shell (SSH) to the SAP XS Advanced server and perform the following steps:
   1. List the applications running on the server using the command `xs apps`.
   2. In the *URL* column, identify the relevant URL for the application (“jobscheduler-dashboard” for Job Scheduler).
   3. Copy the URL to any Web browser to launch the application.

      > **Note**
      >
      > If a valid certificate is not available, the Web browser indicates an issue with the certificate. To resolve the issue, add the required certificate.

   4. Enter the logon credentials to access the application.
2. **Permission to Access the Dashboard**
   The administrator role with the **XS_CONTROLLER_ADMIN** role template contains the **controller.admin** permission. This role template gives you full access to perform all the administration-related tasks. If you do not have the administrator role, you need the roles and role collection templates listed in the table below to view or modify settings.

Task	Roles Required	Role Collection Template Required
Open the dashboard and only view the job listing page **Note** You cannot create jobs.	Org Manager Space Developer	jobscheduler_viewer_template
Configure settings	Org Manager Space Developer	jobscheduler_config_template

Task	Roles Required	Role Collection Template Required
Create or edit jobs or schedules	Org Manager Space Developer	jobscheduler_admin_template

> **Note**
>
> If you already have the permission to create or edit jobs or schedules, you still need **jobscheduler_config_template** to configure settings.

The SAP HANA administrator creates a role collection and adds the Job Scheduler roles to the role collection. For more information, see *Maintaining the SAP HANA XS Advanced Model Run Time* in the *SAP HANA Administration Guide*.

Dashboard Screens

The following table contains the various screens available on the dashboard and their descriptions:

Screen	Description
Configuration	Enables you to maintain the global configuration required for a specific Job Scheduler service instance. • *Max. Invocation Attempts:* The number of attempts made by the Job Scheduler to reach the job action endpoint before it deactivates the job. If the Job Scheduler fails repeatedly to reach the endpoint, it sets the job to inactive. The default value of this parameter is 3. • *Asynchronous Execution timeout (ms):* The duration (in milliseconds) that the Job Scheduler waits for a response for the asynchronous job from the application endpoint. If the application does not provide a response in the specified duration, the run status is set to COMPLETED/UNKNOWN.
Jobs	Lists all the jobs created for a specific service instance. You can delete a job or navigate to the job details by choosing the job name.
Overview	Displays the details of the selected job. You can edit a job.
Schedules	Enables you to create and configure schedules for a job. To access schedules, select a job listed on the dashboard. The *Schedules* screen is displayed. Select a schedule to see the history and logs corresponding to the schedule. To display the run logs of a schedule, choose *Logs*.

Screen	Description
Action History	Maintains the history of a job or schedule for a specific job.

Related Information

Roles for Running the Job Scheduler
Scheduling Jobs in XS Advanced [page 1439]

14.2.4 XS Advanced User Management

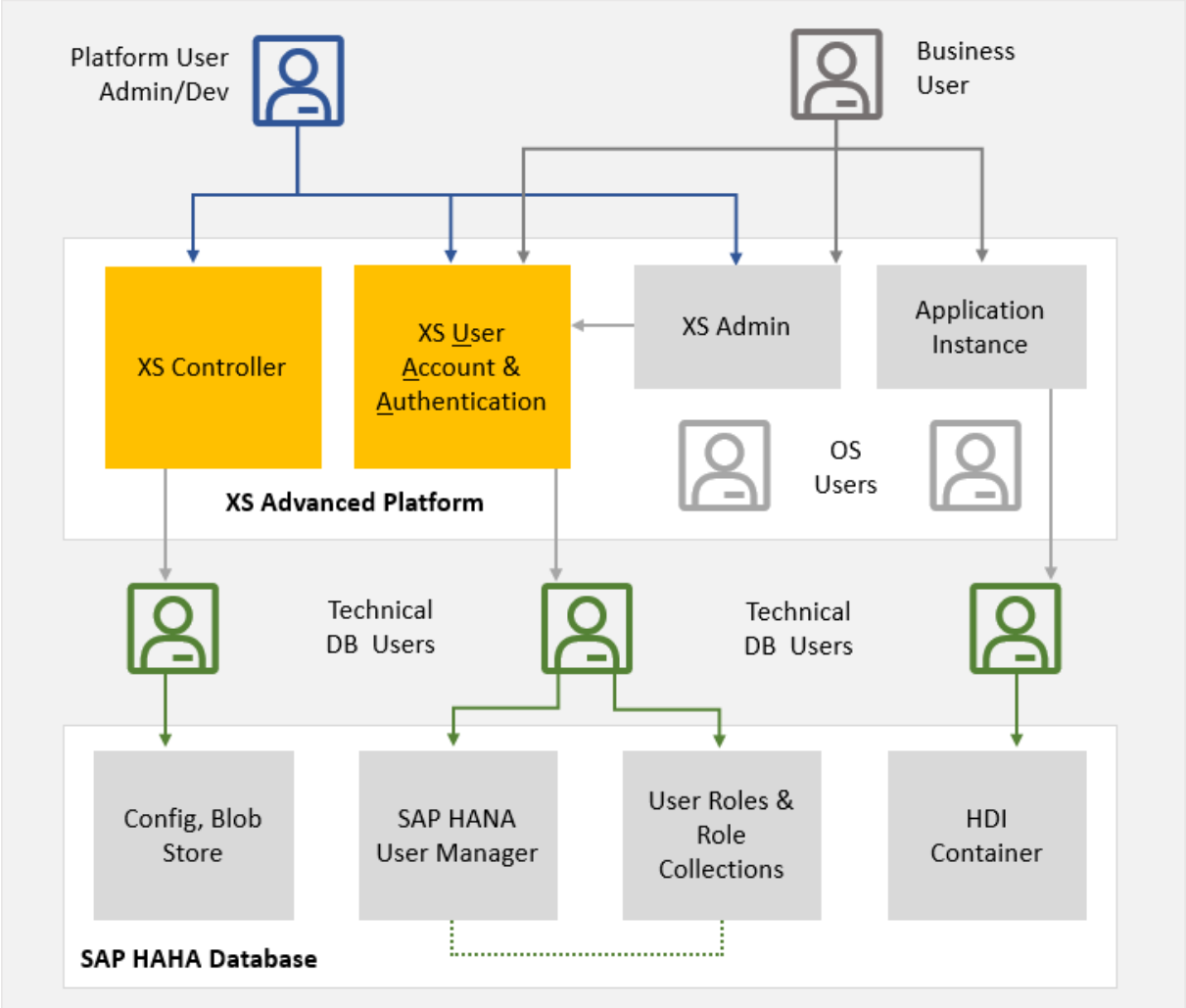
Manage XS advanced users with the tools integrated into the XS advanced platform.

SAP HANA XS, advanced model, requires the configuration and availability of a number of different users, for example: application developers, application users, and XS advanced administrators, as well as technical users, which includes technical database users: database users with restricted permissions. You can manage the users and their roles and scopes with a selection of tools, for example, the `xs` command-line interface, the *XS Admin Tools*, or an Identity Provider (IdP).

XS Advanced User Overview

In traditional application servers, user information is kept in a local user store. However, although the SAP HANA XS advanced platform uses the underlying SAP HANA user store as an Identity Provider (IdP) by default, it is possible to integrate an external IdP such as SAP ID Service or SAP Cloud Identity. It is also possible to configure a custom IdP provided the IdP implements the SAML 2.0 standard.

The **User Account and Authentication** (UAA) service represents the central platform service for the management and authentication of users, as illustrated in the following diagram:



User Account and Authentication Service (UAA)

User information, such as first name, last name, user ID and user privileges, is provided in the form of signed OAuth2 access tokens the central UAA issues when a client logs in successfully. For more information about the authentication procedure, see the section on XS advanced user authentication.

XS Advanced User Categories

XS advanced users access the back-end instances typically through end-user interfaces such as a Web browser or command-line tools. Unlike technical users, application users and some types of system user can be also identified by personal data such as name, e-mail address, and so on. As the same identity provider is the basis for all of XS advanced users, an application user may also be granted developer privileges and the other way around.

XS advanced users who have their source in the SAP HANA user store (default) are typically restricted users with no access to SAP HANA database schemas. In contrast, applications and server components

use **technical SAP HANA users** with certain access privileges. The platform passes these credentials to applications, enabling them to execute SQL statements, if the XS advanced user has sufficient privileges. Decoupling XS advanced users from technical users is the precondition for leveraging external IdPs, even though XS advanced users are also SAP HANA users by default. As technical SAP HANA users are generated by the platform in the background, you typically won't use them to interact with the system.

The XS Advanced User Account and Authentication (UAA) service provides the authentication end point for individual users who need to interact either with SAP HANA XS, advanced model, or with applications deployed and running on the XS advanced model platform. Although such users are often referred to simply as **XS advanced users**, they can have the following roles and areas of responsibility:

- Platform users [page 1471]
- Application or business users [page 1471]
- Operating-system users [page 1471]
- Technical database users [page 1472]

XS Advanced Platform User

Platform users are administrators or developers who are assigned to one or more specific organizations or spaces in the XS advanced platform. An XS advanced administrator user (for example, `XSA_ADMIN`) is allowed to perform any platform operation in any organization or space. However, it is also possible to maintain additional platform user **roles** and use them to restrict the type of access granted to certain users for particular organizations or spaces. XS advanced “administration users” are system users who manage the configuration of the XS advanced application server components, and in particular the XS Controller.

XS advanced platform users are SAP HANA users who have been assigned to a specific XS advanced role collection. Non-administrator platform users can also be managed by means of an external Identity Provider (IdP).

> → Tip
>
> You can use the `xs` command-line interface to maintain XS advanced platform users. For more information, see *Maintaining Platform Users in XS Advanced with the XS CLI* in *Related Information* below.

XS Advanced Application (Business) User

Often referred to as “business users”, application users interact with application instances deployed to and running on the XS advanced run-time platform. Application users are also referred to as business users, for example, employees, customers, and so on.

Application users can be identified by personal data such as name or e-mail address, and this data along with other credentials are stored in a user store, for example, an Identity Provider (IdP); any request to log on to an XS advanced application is managed by the XS advanced User Account and Authentication service (UAA). Authorization scopes (defined in user roles) are granted to application users to restrict or enable access to particular data.

XS Advanced Operating-System Users

In the context of XS advanced, the following predefined operating system users are available by default:

- `<sid>adm`
  Operating-system and administrative SAP HANA system user who owns all platform services as well as the system’s file storage.
- `sap<sid>xsa`

Operating-system user required for staging and running applications in the pre-configured SAP space.

- <sid>xsa
  Operating-system user required for staging and running applications in the pre-configured PROD space.

The <SID>adm operating system user exists to provide an operating system context. From the operating system perspective, the operating system administrator is the user that owns all SAP HANA files and all related operating system processes. Certain administration operations require the operating system user's credentials, for example, starting or stopping the system.

> **Note**
>
> XS advanced application files are also owned by the *xsa operating-system users sap<sid>xsa and <sid>xsa.

XS Advanced Technical Database User

A technical database user does not correspond to a real person and should be used for administrative tasks such as creating objects and granting privileges for a particular application. For example, an application server may log on to the SAP HANA database using a dedicated technical database user. In the context of XS advanced, technical SAP HANA users are generated by the platform in the background.

For XS advanced, the technical user SYS_XS_RUNTIME owns the XS Advanced Controller's SAP HANA schema, which contains the Blob Store, Config Store, and Secure Store. Similarly, the technical user SYS_XS_UAA owns the SAP HANA schema provided for the User Account and Authentication (UAA) for user management

Additional technical database users are created on demand and as required for application-specific purposes. For example, the SBSS_* users are created as a result of an application-service binding. XS advanced also makes use of a number of USR_* users, too; USR_* users are created by the SAP HANA Service Broker for the service plans schema, securestore, and sbss. Similar to the predefined users created when binding an application to an HDI container, USR_* users are used by applications to access their schema. For more information, see *Predefined Users* in *Related Information* below.

> **Note**
>
> With HANA 2.0 SPS 03, the SAP HANA Service Broker no longer uses the SBSS_ prefix for HDI container users. Instead, these HDI container users have the name of the corresponding HDI container as the prefix. For example, for users created during service binding, the following format is used: <HDI_Container_Name>_<GUID>_DT (design-time access) or <HDI_Container_Name>_<GUID>_RT (run-time access). Binding users are assigned the role PUBLIC by default.

Related Information

Predefined Users in XS Advanced [page 1473]
Predefined XS Advanced Database Roles [page 1479]
Configure SAP HDI Parameters
Maintaining Platform Users in XS Advanced [page 1281]
Maintaining Organizations and Spaces in XS Advanced [page 1273]
Maintaining the XS Advanced Run-time Environment with a Graphical User Interface [page 1387]

14.2.4.1 Predefined Users in XS Advanced

The installation of the XS advanced application server creates a small set of predefined users that enable the operation of the underlying system.

The system's super user (<sid>adm) needs to be available in order to manage the life cycle of the system. Similarly, an administrative XS advanced system user (XSA_ADMIN by default) is necessary to perform the initial setup of the application server, for example, granting other users the privilege to create spaces in a dedicated organization and so on. Technical database users are created during installation for all server components that need to persist data in SAP HANA schemas.

This topic contains information about the following types of predefined users in XS advanced:

- Predefined XS Advanced System Users
- Predefined SAP HANA Technical Users
- Predefined XS Advanced Operating-System Users

Predefined XS Advanced System Users

The table below lists the predefined XS advanced system users that are necessary for operating the XS advanced application server. First, an administrative user named XSA_ADMIN is required for the XS advanced Controller; this administrative user configures the application server at a global level. Non-administrative users of the XS advanced Controller are not allowed to perform administration tasks, for example, uploading custom certificates, adding custom buildpacks, or registering platform service URLs. Bear in mind that, although the credentials for the technical users for the SAP HANA Service Broker and UAA Broker are generated automatically during installation, the XSA_ADMIN user is created interactively with a user-defined password. As a first-level administrator user with irrevocable privileges, the XSA_ADMIN has unlimited access to the XS advanced Controller and therefore needs to be handled carefully.

→ Recommendation

- Keep the number of people with XSA_ADMIN credentials as small as possible. Where possible, delegate specific tasks such as space management to users with less privileges instead.
- Avoid creating other powerful users with privileges similar to XSA_ADMIN.
- Change the XSA_ADMIN password at regular intervals and avoid sharing the same password.

User ID	User Type	Description
XSA_ADMIN	XS advanced user	Administrative user for the XS advanced application server with unlimited access to XS advanced Controller API

User ID	User Type	Description
SYS_XS_SUPPORT	Database user	Owns the stored procedures used to grant access privileges on schemas of XS-advanced-related technical database users. Permission to call the stored procedures is also granted to the SYSTEM user. See also SAP note 2656132 in *Related Information* below. **Note** The SYS_XS_SUPPORT user is deactivated by default.

Although the technical users in this table are created in the SAP HANA database, and database authentication checks are used to confirm the technical users' credentials, the technical users are not used to connect to the SAP HANA database.

Predefined Technical SAP HANA Users

Most of the server agents require a data store in the SAP HANA database and therefore need secure access to schemas. For this reason, a dedicated technical SAP HANA user is generated for each such schema, and the credentials of the technical SAP HANA user are passed to the server agent. As the management of technical users is performed at the infrastructure level, end users do not interact with these users.

Caution

It is not recommended to change the properties of a predefined technical SAP HANA user, for example, by changing a user's password or deactivating a user. Modifying a technical user manually will cause XS advanced services to stop working properly. However, for security reasons, it is recommended to rotate the passwords of these technical users from time to time, for example, with the `XSA` command-line interface and the command `XSA renew-passwords-of-technical-users`, as described in the *SAP HANA Administration Guide*.

Starting with SPS04, the technical users listed in the following table are grouped into a single user group named `SYS_XS_UG_RUNTIME_<db-id>`, where `<db-id>` is the unique identifier for a database. The only exception to this user-group rule is SYS_XSA, which is created in the user group `SYS_XS_UG_RUNTIME`. These technical users are used to connect to the SAP HANA database with a specific set of conditions.

User ID	Service	Description
HDI_ADMIN_USER	SAP HANA service broker	Owns SAP HANA schema of SAP HANA Service Broker

User ID	Service	Description
`HDI_BROKER_CONTROLLER`	SAP HANA service broker	Has authorization to access the service broker API of SAP HANA service broker
`SYS_XS_HANA_BROKER`	SAP HANA service broker	Owns the SAP HANA Service Broker's SAP HANA schema
`SYS_XS_HANA_BROKER_INTERNAL`	SAP HANA service broker	Has authorization to execute stored procedures for creating users, and so on.
`SYS_XS_INSTANCE_MANAGER_ADMIN_USER`	Instance Manager	Owns SAP HANA schema of the Instance Manager
`SYS_XS_INSTANCE_MANAGER_BROKER_USER`	Instance Manager	Has authorization to access service broker API of Instance Manager
`SYS_XS_OID_USER`	OIDC	Owns the SAP HANA schema for the OpenID Connect provider
`SYS_XS_OID_USER_SEC`	OIDC	Owns the SAP HANA secure store for the OpenID Connect provider
`SYS_XS_RUNTIME`	Controller	Owns the Controller's SAP HANA schema containing `BlobStore`, `ConfigStore` and `SecureStore`
`SYS_XS_SBSS`	SAP HANA service broker	Owns SAP HANA schema containing procedures to generate user passwords in a secure manner; used by the SAP HANA service broker
`SYS_XS_SYSTEMDB_INFO`	Controller	Has authorization to access database system catalog and configuration
`SYS_XS_UAA`	UAA	Owns the UAA's SAP HANA schema for user management
`SYS_XS_UAA_SEC`	UAA	Owns the UAA's SAP HANA secure store for user credentials
`SYS_XS_UAA_USER_ADMIN`	UAA	Owns the UAA stored procedures that perform user-management operations and assign role collections
`SYS_XSA`	Installer	Owns SAP HANA schema containing a unique tenant ID

User ID	Service	Description
_SYS_DI	HDI	Owns all HDI SQL-based APIs, for example all API procedures in the _SYS_DI schema and API procedures in containers
_SYS_DI_*_CATALOG	HDI	Technical users used by the HDI to access database system catalog tables and views
_SYS_DI_SU	HDI	Technical superuser of the HDI created at installation time
_SYS_DI_TO	HDI	Owns transaction and connections of all internal HDI transactions

Technical Users for HDI Schema-Based Containers

The deployment of database objects with SAP HANA Deployment Infrastructure (HDI) is based on a container model where each container corresponds roughly to a database schema. Each schema, and the database objects deployed into the schema, are owned by a dedicated technical database user.

For every container deployed, a new technical database user and schema with the same name as the container are created. Additional schemas and technical users required for metadata and deployment APIs are also created.

For example, for a container named S, HDI creates the following users:

- S:
  The user who is the owner of the container schema S
- User S#DI:
  The user who is the owner of the schema S#DI containing metadata and deployment APIs
- User S#OO:
  The user who is the owner of database objects in schema S
- Users _DI#S#METADATA_COM_SAP_HANA_DI_<metadata>:
  The users who are the owners of schemas containing build plug-in metadata

These technical users are used internally by HDI only. They are created as restricted database users who do not have any privileges by default (not even the role PUBLIC). They cannot be used to log on to the database.

For more information, see *Maintaining HDI Containers* in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

Technical Users for Default Application Services

XS advanced applications can make use of a number of services managed by a service broker. To make use of a service, an instance of the service must be created and the application must be bound to the specified service instance. Several services are available by default; they are installed with the XS advanced run-time platform.

The installation of the following default application services results in the creation of a number of internal technical users:

- Product-Installer
  Used for the installation and installation management of applications
- Deploy-Service

Used in the technical deployment of applications packaged in multi-target application (MTA) archives

The operation of binding these services to an application generates a technical user and random password according to the following naming convention `USR_<generated_ID>`. These technical users are required to make database schemas available for applications. For every combination of application and schema, such a technical user is created.

In addition, the `Job-Scheduler` service, used to create and schedule long-running operations in the XS advanced environment, uses an HDI container with the `SBSS_` prefix and a randomly generated name. The above-mentioned HDI schemas and users will be created for this container.

For more information, see *The SAP HANA XS Advanced Services: Deployment Infrastructure* in the *SAP HANA Developer Guide (For SAP HANA XS Advanced Model)*.

User Groups for XS Advanced Technical Users

Starting with SAP HANA 2.0 SPS04, technical SAP HANA users are managed in user groups. This is also true for the technical users owned by the XS advanced run-time and the users created when the SAP HANA Service Broker creates a service instance or a service binding. The following table provides an overview of the user groups that contain users who are relevant for XS advanced:

User Group Name	Description	Example Group Members
`SYS_XS_UG_RUNTIME_<db-id>`	Includes all technical users created by the core XS advanced run-time services	`SYS_XS_RUNTIME` `SYS_XS_UAA`

`_SYS_DI #SYS_XS_HANA_BROKER`	Includes all technical users created when the SAP HANA service broker creates a service instance with the service plan "`hdi-shared`"	If "S" is the container name, then:
		`S`
		`S#DI`
		`S#OO`

`SYS_XS_UG_BROKER_HDI_SHARED #SYS_XS_HANA_BROKER`	Includes all service binding users created when the SAP HANA service broker creates a service instance with the service plan "`hdi-shared`"	`*_RT` `*_DT`
`SYS_XS_UG_BROKER_SCHEMA #SYS_XS_HANA_BROKER`	Includes all technical users created when the SAP HANA service broker creates a service instance with the service plan "`schema`"	`USR_*`
`SYS_XS_UG_BROKER_SECSTORE #SYS_XS_HANA_BROKER`	Includes all technical users created when the SAP HANA service broker creates a service instance with the service plan "`securestore`"	`USR_*`
`SYS_XS_UG_BROKER_SBSS #SYS_XS_HANA_BROKER`	Includes all technical users created when the SAP HANA service broker creates a service instance with the service plan "`sbss`"	`USR_*`

Predefined XS Advanced Operating System Users

Ultimately all platform services are made up of operating-system (OS) artifacts such as OS processes, network sockets, and file storage. Since operating systems come with their own user management features, these artifacts are out of necessity owned by OS users. Consequently, the XS advanced application server cannot be run without at least one OS user, although dedicated XS advanced users are able to perform the majority of the operational tasks.

The installation procedure creates the "super" OS user `<sid>adm` for the entire SAP HANA system. As the owner of all operating-system processes, the `<sid>adm` user is very powerful from a security perspective. For this reason, we strongly recommend that you limit the number of people with `<sid>adm` credentials as far as possible.

The following XS advanced platform services launch new processes at run time:

- Execution Agents start application instances.
- The application "Stager" spawns processes running build packs during application staging.

In both cases, custom code comes to execution. If these processes ran as the system's `<sid>adm` user, the whole system could be compromised. To prevent this, the platform generally spawns external processes with OS users that are attached to the application's space. To support this approach, the initial setup includes OS user `<sid>xsa` user for the `PROD` space and OS user `sap<sid>xsa` for the SAP space.

> **Note**
>
> For more information about user and application isolation in organizations and spaces, see *Maintaining Organizations and Spaces in XS Advanced* in *Related Information* below.

The following table summarizes the operating-system users that are available immediately after installation:

User ID	Type	Description
`<sid>adm`	OS user	Administrative SAP HANA system user who owns all platform services as well as the system's file storage
`<sid>xsa`	OS user	OS user for staging and running applications in the pre-configured `PROD` space
`sap<sid>xsa`	OS user	OS user for staging and running applications in the pre-configured `SAP` space

Related Information

Predefined XS Advanced Database Roles [page 1479]
XS Advanced User Management [page 1469]
SAP HANA Security Guide
SAP Note 2656132

The XSA Command Reference [page 1370]
Maintaining Organizations and Spaces in XS Advanced [page 1273]

14.2.4.2 Predefined XS Advanced Database Roles

Several predefined database roles are necessary for the operation of the XS advanced model application server.

> **Note**
>
> The following roles are SQL-based roles that are available in the catalog of the SAP HANA database.

Role	Description
`_SYS_DI_OO_DEFAULTS`	This role contains the set of default privileges that are granted to all HDI container object owner users (`<container>`#OO users). SAP HANA DI uses this role internally to grant default privileges instead of using the PUBLIC role. It contains only privileges to SYS views where additional security checks apply. The role contains SELECT privileges on the views: `SYS.DUMMY`, `SYS.PROCEDURES`, `SYS.PROCEDURE_PARAMETERS`, SYS.TABLES, `SYS.TABLE_COLUMNS`. This role is not intended to be granted to database users. **Note** Do **not** extend this role in a production system.

Role	Description
`SYS_XB_SBSS_VIEWER`	This role contains selected privileges for monitoring the status of the Service Broker Security Support (SBSS) component. The SBSS component provides service brokers with functions for creating, validating, and deleting the credentials they need for service bindings. Credential handling is achieved by creating restricted database users with secure random passwords. Specifically, this role contains read access to the SBSS component version table, in addition to read access to the SBSS bindings table that lists the credential names that have already been created with the SBSS API as well as some meta data for the bound credentials. This role is intended only for support users so they can query information such as SBSS version, number of credentials, names of services brokers that called the SBSS API. **⚠ Restriction** This role does not grant access to any SBSS credentials.

Related Information

Predefined Users in XS Advanced [page 1473]
XS Advanced User Management [page 1469]

14.2.5 XS Advanced System Configuration Parameters

The XS advanced platform can be configured with a selection of system parameters.

Most system parameters are set to a default value during the installation of the XS advanced platform component. If necessary, however, a small selection of XS advanced system parameters can be changed after the installation, too, for example, to meet the requirements of your system landscape and usage patterns. If you need to change the default SSL port or set up a fail-over router, you can change system properties with SAP Cockpit, SAP HANA Studio, or by editing the configuration file itself, for example, `xscontroller.ini`.

ⓘ Note

The XS advanced run-time platform reads its configuration from the SAP HANA initialization (`.ini`) files. To apply any changes made to the XS advanced system parameters, restart the XS advanced run time.

This section contains information about the following XS advanced configuration parameters:

- Global XS Advanced Configuration Parameters [page 1481]
- XS Advanced Controller Configuration [page 1481]
- XS Advanced Execution Agent Configuration [page 1486]

Global XS Advanced Configuration Parameters

The following table lists the system-wide parameters that can be changed in the `global.ini` file and suggests possible scenarios where a change might be necessary.

XS Advanced Global System Parameters global.ini

Property Name	Ini-File Section	Default Value	Description	Reason for Change
basepath_xsa_app-workspace	persistence	`hana/shared/<SID>/xs/app_working`	The path to the working directory of XS advanced. The directory is used for the execution of staging processes and applications. All files in this directory exist only temporarily. If the working directory is changed, it is not necessary to copy the directory contents to the new working directory.	If the startup times for staging processes and applications are slow, change the path to a location on the local file system with a good I/O throughput rather than the shared file system.
xsa_sizing	system_information	M	The sizing profile of the XS advanced platform. For more information, see SAP Note 2509043.	You have a development system or you expect a lot of business users on the production system.

Configuration Parameters for the XS Advanced Controller

If you need to change the behavior of the XS advanced Controller, the following table lists the parameters that you can modify in the `xscontroller.ini` file and suggests possible scenarios where a change might be necessary.

XS Controller System Parameters (xscontroller.ini)

Property Name	Ini-File Section	Default Value	Description	Reason for Change
default_domain	communication	Set during installation	The domain used for the URLs of platform components, for example, the XS Controller or XS UAA. The domain is also used by default for all application URLs (routes).	Set up a fail-over router. Change the application URLs. Set up an additional reverse proxy in front of XS advanced.
router_port	communication	3<instance nr>33	The port the XS advanced platform router is listening on if the `hostnames` routing mode is enabled. The port number is used for every URL or route. **Note** If you configure the router port to a port < 1024, you must set the appropriate file permissions for the `icmbnd` binary at `/hana/shared/<SID>/xs/router/webdispatcher` , for example, `chown root:sapsys` and `chmod 4750`.	You want to use the default SSL port 443 to reach your applications. **Note** This requires some additional configuration steps.
router_portrange_start	communication	51000	The range of ports used by the XS advanced platform to generate port-based routes if no specific port is defined.	You want to create more than 500 routes. You want to run several XS advanced systems on one host. **→ Tip** The port-range modification

Property Name	Ini-File Section	Default Value	Description	Reason for Change
router_portrange_end	communication	51500		should be completed before the installation of the second system. For more details, see SAP Notes 2507070 and 2243156. A change to the port range after installation of XS advanced is not recommended because it will not have any effect on the current routes.
router_https	communication	true	Determines whether the Platform Router provides HTTPs or HTTP endpoints for URLs and routes	You fully trust the network from which applications are accessed and want to disable HTTPs due to the additional performance overhead of SSL.
internal_https	communication	true	Determines whether the XS advanced platform components communicate using HTTPs and client certificate authentication based on the SAP HANA System PKI. If the property is turned off, no authentication between the XS Controller and the Execution Agents is performed.	You fully trust the network from which applications are accessed and want to disable HTTPs due to the additional performance overhead of SSL.

Property Name	Ini-File Section	Default Value	Description	Reason for Change
hana_ssl	communication	false	Determines whether the XS advanced platform components and applications should use SSL encrypted communication with the SAP HANA database. **Note** Additional configuration steps are required to ensure that all applications are capable of establishing an SSL connection successfully. For more information see *Maintaining Trust Certificates* in *Related Information* and SAP Note 2300943.	You want to: • Ensure connections with the SAP HANA database are encrypted • Enable SSL enforcement for connections to the SAP HANA database.

Property Name	Ini-File Section	Default Value	Description	Reason for Change
hana_ssl_validate_certificate	communication	false	Determines whether the XS advanced applications should validate the SAP HANA SSL certificate; this requires additional configuration, which is described in *Maintaining Trust Certificates*. **⚠ Restriction** This property only effects Java-based applications running in the XS advanced run-time environment. SSL validation cannot be switched off for other (non-Java) application run time environments. The XS advanced platform components always validate the SSL certificate, as they are using the System PKI to establish trusted SSL communication with the SAP HANA database.	You want to ensure that the encrypted connection of applications with the SAP HANA database are not vulnerable to man-in-the-middle attacks.
hana_ssl_certificate_hostname	communication	-	Specifies the host name to use when validating the SSL certificate during the SSL handshake. **⚠ Restriction** This property only effects Java-based applications running in the XS advanced run-time environment.	The host name in the certificate does not match the host name the XS advanced run time environment uses to connect to the SAP HANA database.

Property Name	Ini-File Section	Default Value	Description	Reason for Change
mdc_dispatcher	general	true	Determines whether MDC Dispatcher or the `sudo` command is used to start application instances as different operating-system (OS) users. → Tip For more information see SAP Note 2243156.	You want to ensure the `sudo`-based user switch is used, regardless of the SAP HANA version.
default_space_user	general	`<sid>xsa`	Determines the operating-system user defined as the default user for a newly created space. The specified default space user is used to start application instances in the target space.	You do not want to use the `<sid>xsa` user created by the command `hdblcm` as the default space user.

Configuration Parameters for the XS Advanced Execution Agent

If you need to change the behavior of the XS advanced Execution Agent, the following table lists the parameters that you can modify in the `xsexecagent.ini` file and suggests possible scenarios where a change might be necessary.

XS Execution Agent System Parameters (xsexecagent.ini)

Property Name	Ini-File Section	Default Value	Description	Reason for Change
listen_portrange_start	communication	50000	The ports from this range are used to assign the internal port to an application instance. These ports are not supposed to be reached externally. Externally the application instances are reached via the Webdispatcher.	You want to run more than 1000 apps You want to run several XS advanced systems on one host. → Tip Since ports are dynamically assigned to appli-

Property Name	Ini-File Section	Default Value	Description	Reason for Change
listen_portrange_end	communication	50999		cation instances during startup, you can change this property after installation. For more information see SAP Note 2507070.
exclusive	communication	false	If you pin applications to specific Execution Agents, this property controls whether the Execution Agent can start only its pinned applications or other applications, too.	You want to prevent applications from running on the same host. → Tip For more information, see *Maintaining Host Pinning* in *Related Information*.

Related Information

Maintaining Domains and Certificates in XS Advanced [page 1324]
Maintaining Host Pinning [page 1338]
SAP HANA Server Installation and Update Guide
Maintaining Trust Certificates in XS Advanced [page 1321]

14.2.6 Backup and Recovery in XS Advanced

Back up and restore XS advanced persistent data.

XS Advanced integrates with the standard SAP HANA database backup and restore procedures. To backup and restore XS Advanced, you need to backup and restore the tenant databases that are registered for XS Advanced. In addition, when applications make use of the XS advanced file-system service (fss), for example, the SAP Web IDE for SAP HANA, additional steps must be applied. To include the file-system service data in the backup, you have to copy the file-system data to the database, with tools provided by XS Advanced, as described below.

Performing a Backup of XS Advanced

XS Advanced stores all persistent data, such as applications, application runtimes, build packs and application data within the SAP HANA database. XS Advanced data may be stored in a single database as well as in

additional registered tenant databases. For more information, see *Maintaining Tenant Databases* in *Related Information*.

To back up XS advanced, perform the following steps:

1. Copy data stored by the file-system service into the database.
   Run the following command as `<sid>adm`:

   ```
   $ XSA backup-fss
   ```

2. Perform the database backup.
   Use SAP HANA database tools to back up the main XS advanced database and all tenant databases registered for XS advanced. If XS advanced is installed in the system database, a create a full system backup. For more information, see *Maintaining Tenant Databases* in *Related Information*.

   > → Tip
   >
   > To view details of the main database where XS advanced is installed as well as a list of all the registered tenant databases, log into XS advanced as `XSA_ADMIN` user and run the `xs tenant-databases` command, as illustrated in the following example:

   > {·} Sample Code

   ```
   $ xs tenant-databases
   Getting tenant databases as XSA_ADMIN...
   name   XSA    status   master host         sql-port   mappings
   ------------------------------------------------------------------------
   ...
   XSA    main   active   host1.example.org   30015      <default>, myorg:SAP
   APP    yes    active   host1.example.org   30043      myorg:PROD
   ...
   ```

   Make a note of the main tenant database that contains the XS advanced platform persistence. This is the database that needs to be selected after recovery.

Restoring XS Advanced from a Backup Image

When restoring a backup of XS advanced, you need to restore the backup to a target system with the same version of XS advanced as (or a higher version than) the version of XS advanced on the source system at the time of the backup.

> → Tip
>
> If you restore the backup to a system where a higher version of XS advanced is installed, you need to update XS advanced on the target system again (to the same higher version), after the backup finishes, in order to ensure that the version of the restored XS advanced content corresponds with the XS advanced software version installed on the target system.

If you want to restore an instance of XS advanced, you need to perform the following high-level steps:

1. Disable XS advanced services on the target system.

To avoid problems during the recovery operation, it is necessary to shut down XS advanced on the system where the XS advanced backup should be restored. To shut down XS advanced, run the following command as the `<sid>adm` user:

```
$ XSA disable
```

2. Restore tenant databases containing XS-advanced-related data.
   To restore the XS advanced system from a backup, you need to restore at least **all** tenant databases that contain XS advanced data. If the restored XS advanced system was installed in the system database, you have to perform a full system recovery. If several databases need to be restored, it is recommended to use SAP HANA's point-in-time recovery option to recover all databases to a consistent state. The point in time you specify must be before the first database was backed up.

   > → Tip
   >
   > For more information about how to recover a tenant database, see *SAP HANA Recovery* in *Related Information*.

3. Make sure that the `diserver` service is active on all restored tenant databases.
   To start the `diserver` service on all restored tenant databases, run the following statement as `SYSTEM` user in the system database:

   ```
   ALTER DATABASE <tenant database name> add 'diserver'
   ```

4. Select the main XS Advanced tenant database and copy file-system service data from the database back to the file-system.
   Run the following XSA command as `<sid>adm` user:

   ```
   $ XSA select-xsa-runtime-db <tenant database name>
   ```

   This command points the XS advanced services to the new main tenant database, restores the files of the file-system service along with immutable initiation (`INI`) settings such as the routing mode, and enables XS advanced.

   > ⓘ Note
   >
   > If you restore an XS advanced system that was installed in the system database, specify `SYSTEMDB` as value for `<tenant database name>`.

   For information describing how to identify the database containing the XS advanced platform data, see the command `XSA tenant-databases` in *Performing a Backup of XS Advanced* above.

Related Information

SAP Note 3445668
SAP HANA Backup [page 983]
SAP HANA Recovery [page 1029]
Maintaining Tenant Databases in XS Advanced [page 1331]
Recover a System Database to a Point in Time (recoverSys.py) [page 1053]

14.2.7 Logging and Auditing in XS Advanced

Set up and use logs and traces in XS advanced.

The XS Advanced platform writes different types of traces several different files in the following host-specific SAP HANA trace directories:

```
/usr/sap/<SID>/HDB<SN>/<host>/trace
```

The XS advanced platform keeps access logs concerning requests sent to XS advanced applications via the platform router centrally in the following directory:

```
/hana/shared/<SID>/xs/controller_data/controller/router/webdispatcher/logs
```

XS Advanced Platform Tracing

Each XS Advanced platform component, for example, the XS Controller, the XS Execution Agent, or the XS User Account and Authentication service (UAA) writes detailed tracing to log files located in the SAP HANA trace directory of the host the service is running on. The following files contain the most recent log entries:

- `xscontroller_0.log`
- `xsexeagent_0.log`
- `xsuaaserver_0.log`

> **Note**
>
> Log files are closed and renamed (rotated) if they exceed a certain size. For more information about how to set the file size and file-rotation properties, see *Configuration Properties for XS Advanced Log Files* in *Related Information* below.

Log-file rotation means that you can find older traces in files with a higher index incorporated into the file name, for example, `xscontroller_1.log` or `xscontroller_2.log`. The platform retains the five (5) most recent trace files for each XS advanced service. In addition, startup information is written to the following files in the SAP HANA trace directory:

- `xscontroller.out`
- `xsexeagent.out`
- `xsuaaserver.out`

The additional XS advanced services "HANA Broker" and "Instance Manager" write logs to the following files:

- `xshanabroker_0.log`
- `xsinstancemanager_0.log`

Audit Logs

Each of the platform components writes detailed audit logs, which contain information about login attempts and any changes made to resources managed by the platform. The audit logs are stored in the following files in the SAP HANA trace directory on the respective SAP HANA host:

- `hdbxscontroller_audit_0.log`
- `hdbxsexecutionagent_audit_0.log`
- `uaa-audit.log`
- `uaa-oidc-audit.log`

If you enable the SAP HANA audit log, XS advanced stores audit logs from the XS Controller and the Execution Agent service in the SAP HANA audit log. Audit logs created by the UAA Service are not stored in the SAP HANA audit log. For more information about enabling the SAP HANA audit log and configuring the correct audit policy, see *Application Auditing* in *Related Information*.

> **Note**
>
> Audit logs are never deleted automatically. For more information about deleting audit logs manually, see *Data Protection and Privacy* in *Related Information*.

Access Logs

For each route, the platform router maintains access log files with the naming pattern `access_log-<route guid>-<port>.log`, as illustrated in the following example

`access_log-229dba84-8ed6-45d2-91b6-7b142cc58177-51004.log`

To display details of a specific route's globally unique identifier (guid) as well as information about any bound applications, you can use the command `xs routes --guids`, as illustrated in the following example:

> **Output Code**

```
$ xs routes --guids

Getting routes in org "orgname" / space "SAP" as XSA_ADMIN...

host  domain       port  path  type  apps               guid
------------------------------------------------------------------------------
-
      example.org  51006  /    HTTP  auditlog-server
a1dac8fa-620a-4093-8...
      example.org  51007  /    HTTP  auditlog-broker
66eefa83-99fe-4d42-9...
      example.org  51008  /    HTTP  deploy-service     743db7a4-cdd6-43f0-
a...
      example.org  51009  /    HTTP  product-installer
a944c3e9-2367-4c0b-8...
[...]
```

→ Tip

Access logs to applications can also be displayed with the command `xs logs`. For more information, see *Related Information*.

To display access logs for the XS Controller and XS UAA, use the following files:

- `access_log-controller-route-30030.log`
- `access_log-external-uaa-route-30032.log`

The following example shows what an individual entry looks like in an application access log:

{·} Output Code

Access Log Entries

```
[13/Feb/2018:13:23:50 +0100] 7.7.007.007 - - to example.org:30030 "GET /v2/
info HTTP/1.1" 200 sent 471 in 182 by 000-controller-instance
```

An access log line has the following format:

- The timestamp at which the request was finished
- The remote IP address
- The client identification if available or "-" if not available
- The user name in case of basic authentication or "-" if not available
- The target host and port
- The http request
- The response code
- The number of bytes transferred
- The request processing time in milliseconds
- The target system identifier

In the example above, a request was sent from the IP address 7.7.007.007 to target "example.org", requesting the resource "GET /v2/info HTTP/1.1". The response had HTTP status code 200 and 421 bytes were sent in 182 milliseconds by the XS Controller.

ⓘ Note

It is possible to enable the XS advanced platform router to record in the access logs all query parameters used in an application request. For security reasons, the default setting is "false" (disabled). For more information, see *Configuring the XS Advanced Platform Router* in *Related Information* below.

Related Information

Application Auditing (SAP HANA Security Guide for SAP HANA Platform)
Data Protection and Privacy Tools in XS Advanced [page 1497]
Configuring the XS Advanced Platform Router [page 1506]
Configuration Properties for XS Advanced Log-file Size and Log-file Rotation [page 1493]

14.2.7.1 Configuration Properties for XS Advanced Log-file Size and Log-file Rotation

A list of the properities used to configure log-file usage in XS advanced initialization files.

XS advanced provides a number of properties that you can use in the corresponding initialisation file (for example, `xscontroller.ini`, `xsexecagent.ini`, or `xsuaaserver.ini`) to configure how log files are used by XS advanced services, applications, and the platform router.

The information in the following tables describes the properties used to configure the maximum size and number or the log files, as follows:

- *Property Max File Size*
  The maximum size of a log file, for example, 20MB for the `xscontroller_0.log`. When the maximum size for the specified log file is reached, the log file is closed, saved, and renamed, for example, as `xscontroller_1.log`, and a new file `xscontroller_0.log` is created for new log entries. The process is repeated until the maximum number of log files specified in the property "`MaxFiles`" for a given XS advanced service, application, or router is reached. Then the process starts again, rotating the number appended to the log files as needed.
- *Property Max # Files*
  The maximum number of log files allowed before the names of the log files are rotated, for example, 5. If the maximum number of allowed log files (5) is reached and the current `logfile_0.log` is full, then the oldest log file (`logfile_4.log`) is deleted, the names of all other log files are rotated upwards (`logfile_<num+1>.log`) accordingly, the current `logfile_0.log` is closed, saved, and renamed to `logfile_1.log`, and a new log file (`logfile_0.log`) is created for new log-file entries.

Usage

The following example shows how to use the log-file configuration properties to change the maximum size of the XS advanced controller's log files and the maximum number of log files to create and rotate for the `xscontroller_<num>.log` file to "100MB" and "10" respectively, by adding the corresponding properties to the XS advanced configuration file `xscontroller.ini`:

Sample Code

`xscontroller.ini` File

```
[logging]
Controller.Logging.MaxFiles=10
Controller.Logging.MaxFileSize=100MB
```

For the log files used by the XS advanced execution agent, the following example shows how to set the maximum size of each log file to 40MB and the maximum number of log files allowed to 8:

Sample Code

`xsexecagent.ini` File

```
[logging]
ExecutionAgent.Logging.MaxFiles=8
```

```
ExecutionAgent.Logging.MaxFileSize=40MB
```

XS Advanced Run-time Log Files

The following table lists the properties you can use to configure the log files written by the XS advanced `xscontroller`, `xsexecagent` and `xsuaaserver` services, for example, the maximum size of a log file and the number of log files to write before rotation (renaming and removal) starts:

> → Tip
>
> The SAP HANA trace directory is located in `/usr/sap/<SID>/HDB<SN>/<host>/trace/`.

XS Advanced Run-time Log Files

Log-File Name	Location	Default Max # Files	Default Max File Size	Property Max # Size	Property Max File Size	Configurattion File
`xscontroller_<num>.log`	SAP HANA trace directory	5	20MB	Controller.Logging.MaxFiles	Controller.Logging.MaxFileSize	`xscontroller.ini`
`xscontroller.out`	SAP HANA trace directory	1	-	-	-	-
`xsexeagent_<num>.log`	SAP HANA trace directory	5	20MB	ExecutionAgent.Logging.MaxFiles	ExecutionAgent.Logging.MaxFileSize	`xsexecagent.ini`
`xsexeagent.out`	SAP HANA trace directory	1	-	-	-	-
`xsuaaserver_<num>.log`	SAP HANA trace directory	5	20MB	UAA.Logging.MaxFiles	UAA.Logging.MaxFileSize	`xsuaaserver.ini`
`xsuaaserver.out`	SAP HANA trace directory	1	-	-	-	-
`uaa(-<num>).log`	SAP HANA trace directory	5	20MB	UAA.Logging.MaxFiles	UAA.Logging.MaxFileSize	`xsuaaserver.ini`
`uaa-audit(-<num>).log`	SAP HANA trace directory	200	1038090240	AuditLog.File.MaxFiles	AuditLog.File.MaxSizeBytes	`xsuaaserver.ini`
`xshanabroker_<num>.log`	SAP HANA trace directory	5	20MB	UAA.Logging.MaxFiles	UAA.Logging.MaxFileSize	`xsuaaserver.ini`
`xsinstancemanager_<num>.log`	SAP HANA trace directory	5	20MB	UAA.Logging.MaxFiles	UAA.Logging.MaxFileSize	`xsuaaserver.ini`

Log-File Name	Location	Default Max # Files	Default Max File Size	Property Max # Size	Property Max File Size	Configurattion File
`xsoidc_log4j(-<num>).log`	SAP HANA trace directory	5	20MB	UAA.Logging.MaxFiles	UAA.Logging.MaxFileSize	`xsuaaserver.ini`
`xsa_command_<num>.log`	SAP HANA trace directory	5	20MB	-	-	-
`hdbxscontroller_audit_<num>.log`	SAP HANA trace directory	200	1038090240	AuditLog.File.MaxFiles	AuditLog.File.MaxSizeBytes	`xscontroller.ini`
`hdbxsexecutionagent_audit_<num>.log`	SAP HANA trace directory	200	1038090240	AuditLog.File.MaxFiles	AuditLog.File.MaxSizeBytes	`xsexecagent.ini`
`xs_<num>.log`	`~/.xs_logs`	2	20MB	-	-	-

XS Advanced Application Log Files

The following table lists the properties you can use to configure the log files written by each instance of an XS advanced application.

> → Tip
>
> The base path to the location of XS advanced application logs is `<basepath_xsa_appworkspace>/<host>/executionroot/<instance-guid>/`.

XS Advanced Application Log Files

Log-File Name	Location	Default Max # Files	Default Max File Size	Property Max # Size	Property Max File Size	Configurattion File
`stderr.log_<num>`	`<basepath>/logs`	3	8MB	ExecutionAgent.Task.Logging.MaxFiles	ExecutionAgent.Task.Logging.MaxFileSize-	`xsexecagent.ini`
`stdout.log_<num>`	`<basepath>/logs`					
`stdout.sys_<num>`	`<basepath>/logs`					
`stdout.trc_<num>`	`<basepath>/logs`					

Log-File Name	Location	Default Max # Files	Default Max File Size	Property Max # Size	Property Max File Size	Configurattion File
`task-execution.log_<num>`	`<basepath>/logs`					
`staging-err.txt`	`<basepath>/logs`	1	-	-	-	-
`staging-out.txt`	`<basepath>/logs`	1	-	-	-	-
`access.log`	`<basepath>/app-logs`	4	8MB	-	-	-

XS Advanced Platform-Router Log Files

You can use the following properties to configure the log files written by the XS advanced platform router:

> → Tip
>
> The base path to the location of the log files for the XS advanced platform router is `$XSPATH/controller_data/controller/router/webdispatcher/`.

XS Advanced Platform-router Log Files

Log-File Name	Location	Default Max # Files	Default Max File Size	Property Max # Size	Property Max File Size	Configurattion File
`dev_trace`	`<basepath>/logs`	2	50MB	-	-	-
`dev_icm_sec`	`<basepath>/logs`	2	50MB	-	-	-
`stderr.txt`	`<basepath>/logs`	1	-	-	-	-
`stdout.txt`	`<basepath>/logs`	1	-	-	-	-
`access_log-controller-route-api.log`	`<basepath>/logs`	3	8MB	Router.Logs.MaxFiles	Router.Logs.MaxSize	xscontroller.ini
`access_log-external-uaa-route-uaa-server.log`	`<basepath>/logs`					

Log-File Name	Location	Default Max # Files	Default Max File Size	Property Max # Size	Property Max File Size	Configurattion File
`access_log-<route-guid>-<hostname/port>.log` *	`<basepath>/logs`					

> **Note**
>
> * An instance of the `access_log-<route-guid>-<hostname/port>.log` is written for each available application route in the system and can be identified by the route's global unique identifier (guid), which is added to the log-file name.

Related Information

Logging and Auditing in XS Advanced [page 1490]

14.2.7.2 Data Protection and Privacy Tools in XS Advanced

Ensure the protection of private user data in XS advanced.

To comply with data privacy regulations, it must be possible to delete personal and private data stored in an XS advanced system. The XS advanced run-time environment stores the following types of user-related data:

- IP addresses and user names in the audit log and trace files
- IP addresses in the router access logs
- User names in the application log files
- Shadow users in the UAA

XS advanced administrators can use the command `XSA delete-personal-data` to remove personal and private data. For example, the following command ensures the deletion of all personal data until the second of January 2018:

```
XSA delete-personal-data -u 2018-01-02
```

The `--exclude` enables you to prevent certain types of user information from being included in the delete operation, for example, if is necessary to keep the information in order to meet other regulations, such as maintaining the integrity of the audit log, where the information is required. The `delete-personal-data` command enables you to delete sensitive information from the following types of sources:

- PLATFORM
  The platform logs including the event log in the database and the trace files
- AUDIT
  The audit log files of XS advanced

- APP
  The application log files
- ACCESS
  The Webdispatcher access logs
- USERS
  The shadow users in the User Account and Authentication (UAA) server

The following command deletes personal data from all sources in XS advanced **except** the XS advanced audit logs up until February 2, 2019:

```
XSA delete-personal-data -u 2019-01-02 --exclude AUDIT
```

> ⚠ Caution
>
> It is not possible to use this command to delete entries stored in the SAP HANA audit log. For more information about deleting entries stored in the SAP HANA audit log, see *Delete Audit Entries* in *Related Information*.

Data cannot be recovered after deletion. The only exception to this rule is the shadow users in the User Account and Authentication server (UAA), which are created again the next time the users log in. Log files are not included in the SAP HANA backup.

Related Information

Delete Audit Entries
Offline Maintenance and Diagnostics of XS Advanced with the XSA CLI [page 1369]
The XSA Command Reference [page 1370]
Data Protection in XS Advanced (Security Guide)

14.2.8 Platform Sizing in XS Advanced

Set up usage profiles and resource consumption at the application and platform levels.

According to the business scenario and load profile, XS advanced platform services have a resource consumption that needs accurate alignment with the available hardware resources. For example, operation of the XS advanced platform requires certain amount of disk space, main memory and number of network connections. In addition, the resource consumption it highly depends on the applications deployed on the platform.

> ⓘ Note
>
> A tradeoff is required between the platform's resource consumption and the maximum load the server can handle.

By default, XS advanced is optimized for low-memory consumption, which initially limits the maximum number of concurrent requests sent by platform and application users. If you want to configure XS advanced to handle

higher loads, you may need to resize the platform services accordingly without hitting resource limits. Usage profiles help administrators to cope with this task as described in the following sections.

Usage Types and Profiles

To simplify sizing configuration, XS advanced comes with a predefined set of usage and sizing profiles enabling administrators to configure the server with a best-fit usage type and a pre-configured size profile, for example, "large" (L) or "extra large" (XL). Starting with version SPS03, XS advanced server provides the following usage types:

- `PlatformUsage` (Platform Usage)
- `AppUsage` (Application Usage)

Platform Usage

XS advanced systems configured with the usage type `PlatformUsage` are primarily intended to serve users that interact with the platform interface (`xscontroller`), for example, when pushing applications, viewing the application status, or administrating global platform settings. For this reason, the chosen size profile scales the platform services that are responsible for processing such requests. However, the platform router has a default (medium) layout that is independent of the size profile and which assumes that only a few requests to application end points are expected. The typical use case for this usage setting is a development system, and the impact of the profile size on the platform service capabilities is illustrated in following table.

Platform Usage Profile Settings

Profile Size (Short)	Profile Size (Long)	Max. Concurrent Platform Requests
S	Small	4
M	Medium	16
L	Large	64
XL	Extra Large	256

Application Usage (AppUsage)

Systems that are mainly intended to process application requests from business users should be configured with usage type `AppUsage`. In this scenario, the server infrastructure for handling platform user requests has a default layout while the platform router is scaled according to the chosen size profile. A production system would typically use this setting. The impact of the profile size is illustrated in following table.

Application Usage Profile Settings

Profile Size (Short)	Profile Size (Long)	Max. Concurrent Application Requests
S	Small	100
M	Medium	2,000

Profile Size (Short)	Profile Size (Long)	Max. Concurrent Application Requests
L	Large	8,000
XL	Extra Large	32,000

Server Configuration

By default, all usage types have profile size M (medium). You may adapt the profile size of a dedicated usage type in global.ini by means of the property `xsa_sizing` in section `system_information`. Using the format `<usage>:<size>` you can set the size of a specific usage type to one of the pre-defined sizes (L, XL, etc.). The following example shows a server configuration optimized to serve a maximum of 64 platform requests in parallel:

```
[system_information]
xsa_sizing = PlatformUsage:L
```

The following example shows a server configuration optimized to handle a maximum of 8,000 concurrent application requests:

```
[system_information]
xsa_sizing = AppUsage:L
```

It is possible to specify size profiles for all different usage types individually, which could be helpful in mixed scenarios. In the example below the server is configured to handle a very large number of platform users and a large number of business requests at the same time, which is the kind of scenario you would expect in a development and test system running extensive application tests:

```
[system_information]
xsa_sizing = PlatformUsage:XL, AppUsage:L
```

Providing the usage type is optional, and if it is not specified, the profile size applies to all usage types by default. So `'xsa_sizing = L'` induces the same server configuration as `'xsa_sizing = PlatformUsage:L, AppUsage:L'`. This ensures the compatibility of sizing configurations for pre-SPS03 released versions of SAP HANA.

> **Note**
>
> Changes to the `xsa_sizing` property require a restart of the XS advanced server (for example, with `XSA restart`) in order to make SAP HANA aware of the modified configuration.

Resource Consumption

Depending on the chosen platform sizing configuration, the platform services have a certain resource consumption that needs to be taken into consideration when XS advanced is activated within an SAP HANA system. For example, the XS advanced platform has a significant impact on the following resources:

- Disk space and file IO
- Main memory
- Network connections and network IO

The profile size mainly influences the amount of occupied main memory, the maximum number of network connections, and the maximum file and network IO rates. Disk space, however, is not correlated to the profile size; it is associated with the number and size of the deployed applications.

> → Tip
>
> For more information about recommendations for resource-consumption settings, see *Related Information*.

Application Sizing

The usage types and profile sizes specified in the corresponding section of the configuration-parameters (`.ini`) file enable only the scaling of the core platform services, which do not affect applications that are deployed to the XS advanced platform (including system applications such as the deploy service or audit log service). To meet load requirements, applications generally need to be scaled horizontally or vertically. For more information, see *Related Information*.

Related Information

SAP Note 2618752
Scaling Applications in XS Advanced [page 1304]
XS Advanced System Configuration Parameters [page 1480]

14.2.9 Configuring the XS Advanced File-System Service

Configure the service that XS advanced provides for storing file-based data.

The information in this section describes how to configure the XS advanced file-system service (FSS), which enables applications to store file-based data. For more information about using the file-system service for XS advanced, see *Related Information* below.

File-System Service Backup

When creating and maintaining instances of the file-system service, it is important to bear in mind that the instances of the file-system service that you create are not included in any automatic, platform-wide, backup operation. If you want to ensure that instances of the file-system service are saved to a backup copy that can

be restored at some point if necessary, you must perform a manual backup operation, for example, by using the XSA command-line interface, which includes the following administrator tools:

> **Note**
>
> To use the XSA command, you must log on to the XS advanced platform in a command shell as the operating-system user `<SID>adm`.

- `XSA backup-fss`
  Create a backup copy of the file-system service instances in the database
- `XSA restore-fss`
  Restore a backup copy of the file-system service instance from the database

The standard platform-wide operation to replicate the system does not automatically include the data maintained in an instance of the file-system service; you must perform a manual replication of the file-system data. For more information, see the section describing *Table Replication* in the *System Administration* section of the *SAP HANA Administration Guide*.

> **Note**
>
> If an application uses several different types of backing-services at the same time (for example, a file-system service instance and an SAP HANA service instance), the system cannot ensure a consistent backup of the respective service instances. For this reason, the applications themselves are responsible for dealing with service instances that are recovered from backups taken at different points in time.

Configuring the File-System Service Storage Path

By default, the XS advanced file-system service stores files in the following directory:

```
<$XSPATH>/controller_data/fss
```

However, you can change the default setting and configure a custom storage directory with the property `fss_storage_path` in the `fss` section of the `xscontroller.ini` file, as described below.

> **Restriction**
>
> Configuration of the storage path for the XS advanced file-system service is only possible from XS advanced version 1.2.0. In addition, to use the XSA command, you must log on as the operating-system user `<SID>adm`.

The following steps show how to change the default location for data stored by the XS advanced file-system service **and** move existing data to the new location:

1. Shut down XS advanced.

   ```
   XSA disable
   ```

2. Create a backup of the file-system-service data in the current location.

   ```
   XSA backup-fss
   ```

3. Specify the absolute path of the new storage location.
   Set the `fss_storage_path` in the `fss` section of the `xscontroller.ini` file as follows:

   Sample Code

   `xscontroller.ini`

   ```
   [fss]
   fss_storage_path=</absolute/path/to/custom/storage-directory>
   ```

   Note

   The new storage directory must be owned by the `<SID>adm` user. In addition, the permissions for all parent directories of the storage directory must be set to "executable by `others`".

4. Restore the file-system service-data in the new location.

   ```
   XSA restore-fss
   ```

   Note

   If XS advanced is running in a multi-host system, make sure that the new location for the file-system-service storage is accessible on all SAP HANA hosts.

5. Restart XS advanced.

   ```
   XSA enable
   ```

   → Tip

   Use the `XSA diagnose` command to measure the performance of the new storage location that you have configured for the XS advanced file-system service.

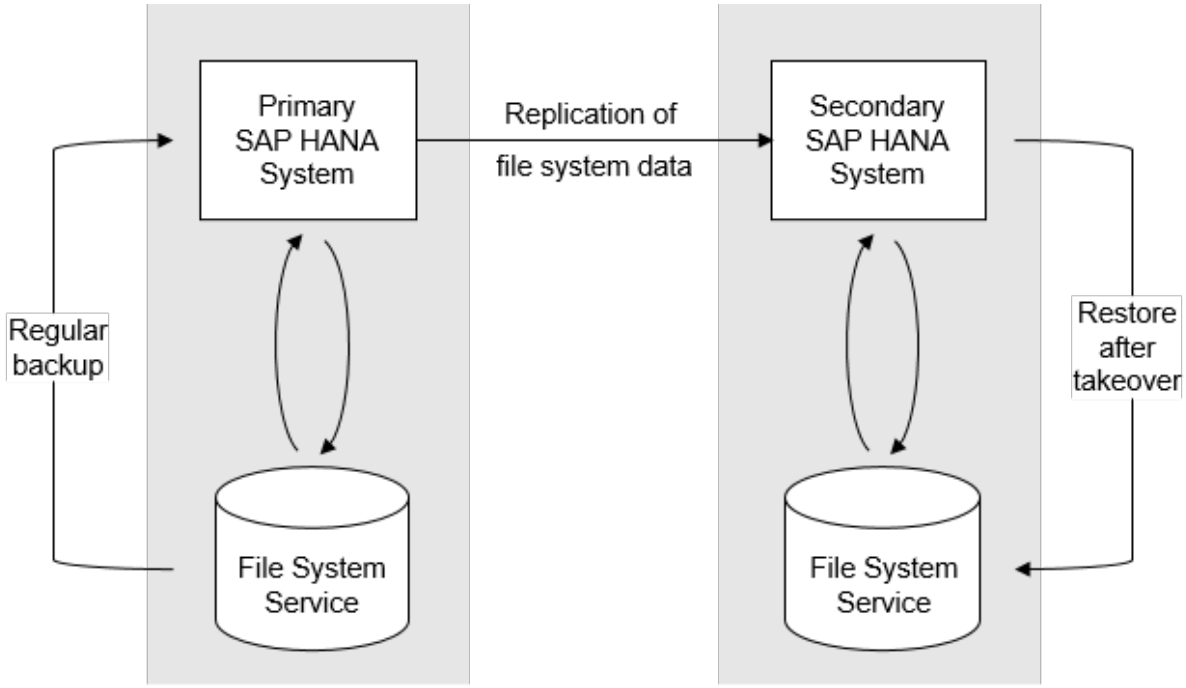
Using the File-System Service in a SAP HANA System Replication Setup

The data stored by the XS advanced file-system service is not automatically replicated to secondary systems by SAP HANA system replication (HSR). To ensure that the file-system-service data is available on secondary systems after a failover, you can choose one of the following options:

- Back-up the XS advanced file-system-service data (for example, with a `cron` job)
- Set up a shared network storage facility

Back up the XS Advanced File System Data

If you back up the data stored by the XS advanced file-system service to the database at regular intervals, SAP HANA system replication will replicate the backed-up data to any secondary systems automatically.



Separate XS Advanced File-System-Service Storage Location

For example, to automate the file-system service in Linux-based systems, the `<SID>adm` user can run the following command as a `cron` job:

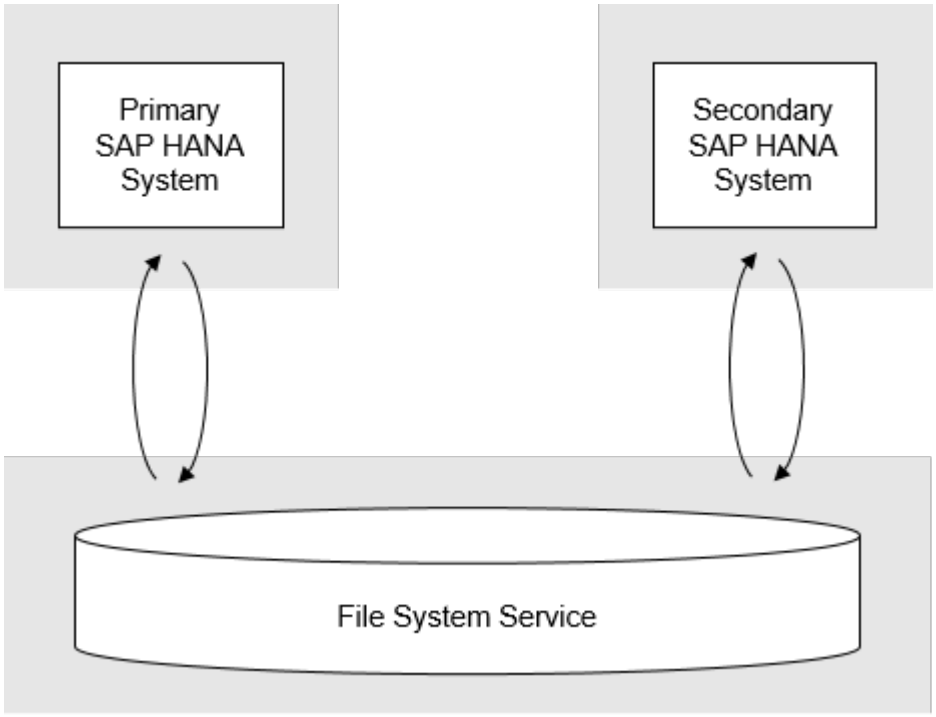
```
XSA backup-fss
```

After a failover, the `<SID>adm` user can use the following command on the newly activated primary system to restore the replicated file-system data on disk:

```
XSA restore-fss
```

Use Shared Network Storage

It is also possible to set up a shared network storage that is available on the primary and all secondary systems. With this option, no manual actions are necessary during a takeover, as the secondary system has access to the latest file-system-service data immediately.



Shared XS Advanced File-System-Service Storage Location

However, to enable the XS advanced file-system service to make use of the shared-storage option, you need to perform the following high-level steps:

1. On the primary system:
   Perform the steps defined in the section *Configuring the File-System Service Storage Path* above and point the storage path of the file-system service to the shared location.
2. On the secondary system:
   You only need to configure the property `fss_storage_path` in the `fss` section of the `xscontroller.ini` file to the same value that you configured in the primary system, namely; the location of the shared network storage.

> ### Note
>
> On the secondary system, you do not need to perform any of the other steps described in the section *Configuring the File-System Service Storage Path*, for example, disabling and reenabling XS advanced or backing up and restoring the XS advanced file system.

To revert the change to a shared-storage location, for example, to return to a configuration with separate storage locations for primary and secondary systems, use the instructions in *Configuring the File-System Service Storage Path* above to move the file-system-service data to the new (separate) locations.

> ### Note
>
> If you are setting up a shared network storage for the XS advanced file-system service, there is no need to stop SAP HANA System Replication (HSR) during the set up operation.
>
> If the XS advanced file-system service uses a shared network storage, the data of the file-system service is not backed up to the database. If required, a backup of the file-system-service data must be performed manually.

In addition, the shared-storage configuration requires that all XS advanced space users and the `<SID>adm` user on both the primary and secondary systems have the same user ID.

In the shared-storage configuration, make sure that only one of the systems is active as the primary system.

Related Information

File-System Storage Services in XS Advanced (SAP HANA Developer Guide for SAP HANA XS Advanced Model)

14.2.10 Configuring the XS Advanced Platform Router

Use a template or INI parameters to configure the XS advanced platform router.

XS advanced model uses a dedicated instance of the SAP Web Dispatcher as the central platform router. The configuration of this Web Dispatcher instance is managed by the XS Controller and is highly dynamic; the configuration regularly changes, for example, when applications are started, stopped, or scaled, or when routes are mapped to (and unmapped from) applications. Due to this dynamic behavior, it is not possible for the administrator to change the configuration of the Web Dispatcher directly. Instead, changes to the configuration are defined in a configuration template which is used by the XS advanced platform to build the final Web Dispatcher configuration that is used by the XS advanced platform router.

The information in this section describes some examples that demonstrate the process including details of the following steps:

- Configuring the Platform Router with the Configuration Template [page 1506]
- Configuring the Platform Router with INI Parameters [page 1508]

Configuring the Platform Router with the Configuration Template

XS advanced provides a template file `sapwebdisp.template` which you can use to override or add SAP Web Dispatcher configuration properties. For a complete reference of all configuration parameters, see the SAP Web Dispatcher documentation in *Related Information* below.

> **Note**
>
> Properties defined in the template file take precedence over properties determined by the XS advanced platform.

Saving any changes made to the configuration template file prompts XS advanced to automatically update the Web Dispatcher, which means that it is not necessary to restart the Web Dispatcher manually. The template

file can be edited by the <sid>adm user and is located in the "xs/" directory of the SAP HANA installation, as illustrated in the following path:

```
/hana/shared/<SID>/xs/controller_data/controller/router/webdispatcher/conf/
sapwebdisp.template
```

Any changes made to the XS advanced router-configuration template `sapwebdisp.template` will be reflected in the actual Web Dispatcher configuration file named `sapwebdisp.pfl`

> **Note**
>
> Some configuration properties of the SAP Web Dispatcher cannot be changed dynamically.

For changes to configuration properties of the SAP Web Dispatcher that cannot be updated dynamically, refer to the SAP Web Dispatcher documentation, for example, in *Related Information* below. In these cases, a restart of the SAP Web Dispatcher is required, for example, by sending signal 3 (SIGQUIT) to "kill" the Web Dispatcher process directly. Alternatively, you can restart the XS Controller service or the entire XS advanced platform, for example, using the `XSA restart` command. Bear in mind that restarting the XS Controller or the XS advanced platform will cause a longer downtime than restarting only the Web Dispatcher.

> **Caution**
>
> It is not recommended to use the template file to override any parameters that include an index in their key.

Although it is possible to override parameters that include an index in their key (for example, `wdisp/system_xx`), it is not recommended. This is because the XS advanced run time cannot guarantee that these indexes remain stable and always reference the same logical system. XS advanced incrementally increases the index for the parameters it sets, starting at zero. If you want to add additional parameters that use an index in their key, make sure you use a very high value as the index.

Configuring a Load Balancing Algorithm

The SAP Web Dispatcher provides several options for requesting load balancing. For more details about what those options are and how to set up and use them, see *Load balancing via SAP Web Dispatcher* in *Related Information* below.

Serving XS Classic with the XS Advanced Web Dispatcher

One use case for adding additional configuration to the XS advanced Web Dispatcher is related to XS classic, for example, to enable you to serve XS classic and XS advanced applications on the same TCP port. This is useful when running XS advanced in hostname-based routing mode, as you only need to open a single port for HTTPs communication in that case. The SSL settings will be taken from the XS advanced Web Dispatcher, which means you only need to configure the HTTP endpoint of XS classic, so that internal communication between the XS advanced Web Dispatcher and the XS classic Web Dispatcher can be established. To enable such a configuration, add the following line to your `sapwebdisp.template` file:

> **Sample Code**
>
> Connect the XS Advanced and XS Classic Web Dispatchers in `sapwebdisp.template`
>
> ```
> wdisp/system_999=NAME=ZZZ, SID=ZZZ, SRCURL=/, SRCVHOST=xsc.<default-
> domain>:<router-port>, SSL_ENCRYPT=0, EXTSRV=<xs-classic-http-url>
> ```

Values for the placeholders `<default-domain>` and `<router-port>` can be obtained from your XS advanced, system-parameter configuration, which is described in *XS Advanced System*

Configuration Parameters in *Related Information*. The value of `<xs-classic-http-url>http://127.0.0.1:80<instance-nr>`". After you have adapted the template file, the XS classic start page should be available at " should reflect the HTTP end point of the XS classic Web Dispatcher. If both XS advanced and XS classic are running on the same host (for example, within a single-host SAP HANA installation), the end point for the Web Dispatcher could be "`https://xsc.<default-domain>:<router-port>`".

Configuring the Platform Router with INI Parameters

Some settings of the XS advanced Web Dispatcher can be configured by means of parameters within the file `xscontroller.ini` as described in *XS Advanced System Configuration Parameters* in *Related Information*. For example, the property `router_https` determines whether the Web Dispatcher exposes HTTP or HTTPs end points.

Configuring TLS Versions and Cipher Suites

The TLS settings of the Web Dispatcher server ports used by the XS advanced platform can be configured by means of an INI parameter. The following INI settings are available in the `xscontroller.ini` file. If you are running a multi-host system, with multiple `xs_worker` roles configured, the settings also need to be added to the `xsexecagent.ini` file. This is because an additional Web Dispatcher is started by Execution Agents, which are not running on the XS advanced primary (controlling) host.

INI Parameter for Router Cipher-Suite Setting

Property Name	Section	Default	Description
Router.WebDispatcher.CipherSuites	router	545:PFS:HIGH::EC_X25519:EC_P256:EC_HIGH	The TLS cipher suite settings, provided on server ports exposed by the Web Dispatcher (enablesTLS 1.2 only)

The default cipher-suite setting restricts communication to TLS 1.2 only. If, however, clients are known to be incompatible with TLS 1.2, it is recommended to enable TLS 1.0 and TLS 1.1 in addition to TLS 1.2, for example, by changing the `Router.WebDispatcher.CipherSuites` parameter in the XS advanced controller's configuration file (`xscontroller.ini`) as illustrated in the following example:

Sample Code

Excerpt from the XS advanced controller configuration file (`xscontroller.ini`)

```
[router]
Router.WebDispatcher.CipherSuites =
135:PFS:HIGH:TLS_FALLBACK_SCSV::EC_P256:EC_HIGH
```

For more details about the format required for the cipher-suite parameters illustrated in the example above as well as the required sequence and meaning of the values used, see SAP Note 510007 *Setting up SSL on Application Server ABAP* in *Related Information* below.

Note

All internal HTTPs ports used by XS advanced are restricted to TLS 1.3 by default.

Logging Query Parameters

It is possible to enable the XS advanced platform router to record in the access logs all query parameters used in an application request, for example, using the `Router.WebDispatcher.LogQueryParameters` parameter in the XS advanced controller's INI file (`xscontroller.ini`). For security reasons, the default setting is "false" (disabled), and no query parameters are written to the access logs.

> ⚠ Caution
>
> Since the logged information could include details of a sensitive nature, it is not recommended to enable this feature other than perhaps temporarily in exceptional circumstances, for example, when troubleshooting security issues.

INI Parameter for Logging Query Parameters

Property Name	Section	Default	Description
Router.WebDispatcher.Log-QueryParameters	router	false	Write the query parameters used in a request to the plat-form-router's access logs

Related Information

XS Advanced System Configuration Parameters [page 1480]
SAP Web Dispatcher
Load balancing via SAP Web Dispatcher (SAP Community Wiki)
SAP Note 510007

14.2.11 Maintaining Single Sign-On for XS Advanced Applications

You can configure XS advanced applications to use single sign-on (SSO) authentication to confirm the logon credentials of a user calling an application service.

Single sign-on is a convenient way to authenticate a user against the XSUAA service using one or multiple available certificates. The XS advanced runtime enables you to use X.509 authentication as well as SPNEGO and Kerberos.

> ⓘ Note
>
> To prevent a login-logout loop with SSO, it is recommended to configure a custom logout page for the XS advanced application. For more information, see the reference documentation for the application router configuration syntax.

Related Information

Configure SSO with X.509 Authentication for XS Advanced Applications [page 1510]
XS Advanced Application Router Configuration Syntax

14.2.11.1 Configure SSO with X.509 Authentication for XS Advanced Applications

Enable SSO with X.509 certificates for user logon to applications in the XS advanced run-time environment.

Prerequisites

- You have the role USER_ADMIN on your SAP HANA database.
- You have the role XS_CONTROLLER_ADMIN on your XS advanced runtime.
- You have an SAP HANA administration tool.
- You have the root certificate of the X.509 user certificates.
- The parameters `uaa.oidc.enableoidc` and `uaa.oidc.enablex509` in the `xsuaaserver.ini` file are set to `true`.

Procedure

1. Add the trust certificate for client authentication.

   Add the root certificate of the X.509 user certificates in `PEM` format to your XS advanced runtime platform.

   Use the `xs trust-certificate` command with the --client-auth option, as illustrated in the following example:

   ```
   xs trust-certificate <ALIAS> --client-auth -c <PATH>
   ```

   You see a confirmation in your command prompt.

   ```
   Adding trusted certificate <ALIAS> as <user>...
   OK
   ```

2. Create a database user whose identity is defined in an X.509 certificate issued by your certificate authority (CA).
   a. Create a new user in the SAP HANA database based on the details specified in an existing X.509 certificate.

The following example shows how to use the SQL statement CREATE USER WITH IDENTITY to create the database user “MyUserName” and the corresponding X.509 certificate mapping:

```
CREATE USER MyUserName WITH IDENTITY 'CN=MyUserName, O=SAP-AG, C=DE'
ISSUER 'CN=SSO_CA, O=SAP-AG, C=DE' FOR X509
```

b. Configure the user in SAP HANA.

   To edit the user for whom you want to enable SSO, use the SAP HANA administration tool of your choice: select *X509* and *Configure* X.509 user certificates.

3. Use a Web browser to test the logon authentication settings for the XS advanced application.

   When you enter the URL for your application in the Web browser, the Web browser prompts you to select a certificate, which enables you to log on without supplying logon credentials manually.

Related Information

Maintaining Trust Certificates in XS Advanced [page 1321]

14.2.11.2 Configure SSO with SPNEGO and Kerberos for XS Advanced Applications

Enable SSO with SPNEGO and Kerberos for user logon to applications in the XS advanced run-time environment.

Prerequisites

- You have configured Kerberos and SPNEGO support in your SAP HANA database.
- You have an SAP HANA administration tool.

Context

In addition to configuring Kerberos in SAP HANA, see SAP Notes 1813724 and 1837331.

> **Note**
>
> For SPNEGO, the Kerberos keytab must contain an entry for Service Principal Name (SPN): *HTTP/<fully-qualified-host-name>*

Procedure

1. Open your SAP HANA adminstration tool and set the parameter `uaa.oidc.enablespnego` in the `xsuaaserver.ini` file to `true`.
2. Configure your SAP HANA user.

   Edit the user in your SAP HANA administration tool: select *Kerberos* and enter an *External ID*.
3. Test your configuration.

   With the Python script from SAP Note 1813724 you may verify, that your logon to SAP HANA with your Kerberos credentials works.

Related Information

Configure Kerberos for SAP HANA Database Hosts [page 558]

14.2.12 Multi-Host Setup with XS Advanced

Scale out inter-host communication and failover router with XS advanced in a multi-host setup.

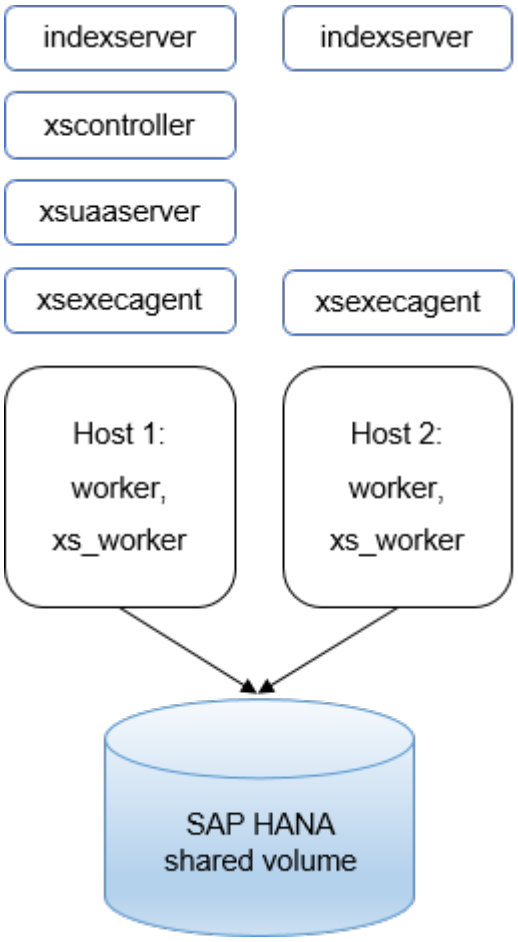
If you want to learn about how XS advanced integrates into an SAP HANA multi-host setup, see the *SAP HANA Installation Guide* in *Related Information*.

Scaling Out XS Advanced to Multiple Hosts

If you want to distribute XS advanced applications to more than one host, for example, in a scale-out scenario, you can assign the `xs_worker` role to additional SAP HANA hosts. Newly started application instances are distributed in a round-robin fashion among the existing hosts with the `xs_worker` role. You can also control which application is scheduled ro tun on which host by pinning an application to a specific host, as described in *Maintaining Host Pinning* in *Related Information* below.

Combining "worker" and "xs_worker" Roles on a Single Host

In order to scale-out XS advanced applications in an existing multi-host setup, you can add the `xs_worker` role to hosts that already have the `worker` role, for example, using the functionality provided by the `hdblcm` command, as shown in the following figure:



Combining xs_worker and worker Roles on the Same Host

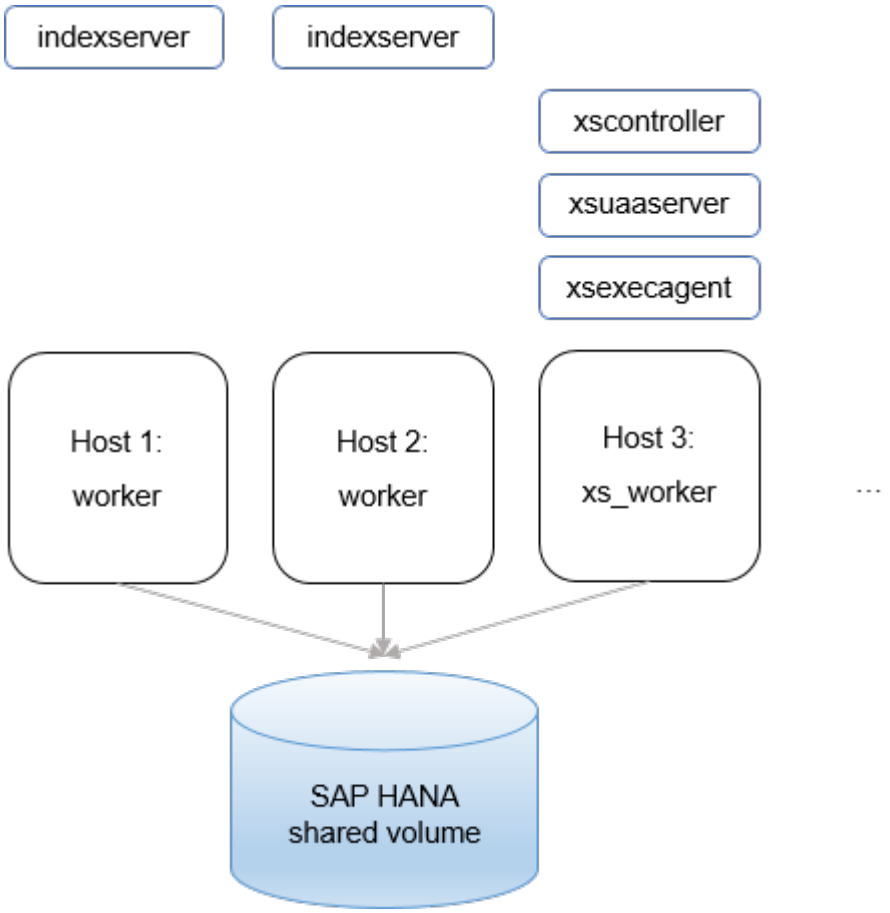
The first host assigned the `xs_worker` role runs the processes `xscontroller`, `xsuaaserver` and `xsexecagent`. Additional hosts with the `xs_worker` role only run `xsexecagent` services, which enables them to start application processes on the additional hosts as well.

For more information about assigning additional roles to hosts, see *Adding Host Roles*, in *Related Information* below.

Separating "worker" and "xs_worker" Roles Between Different Hosts

It is also possible to separate XS advanced application processes from SAP HANA database processes. If you add additional SAP HANA hosts and assign them only the `xs_worker` role, host resources (for

example, memory or CPU) are strictly separated between XS advanced application and database processes, as illustrated in the following diagram:



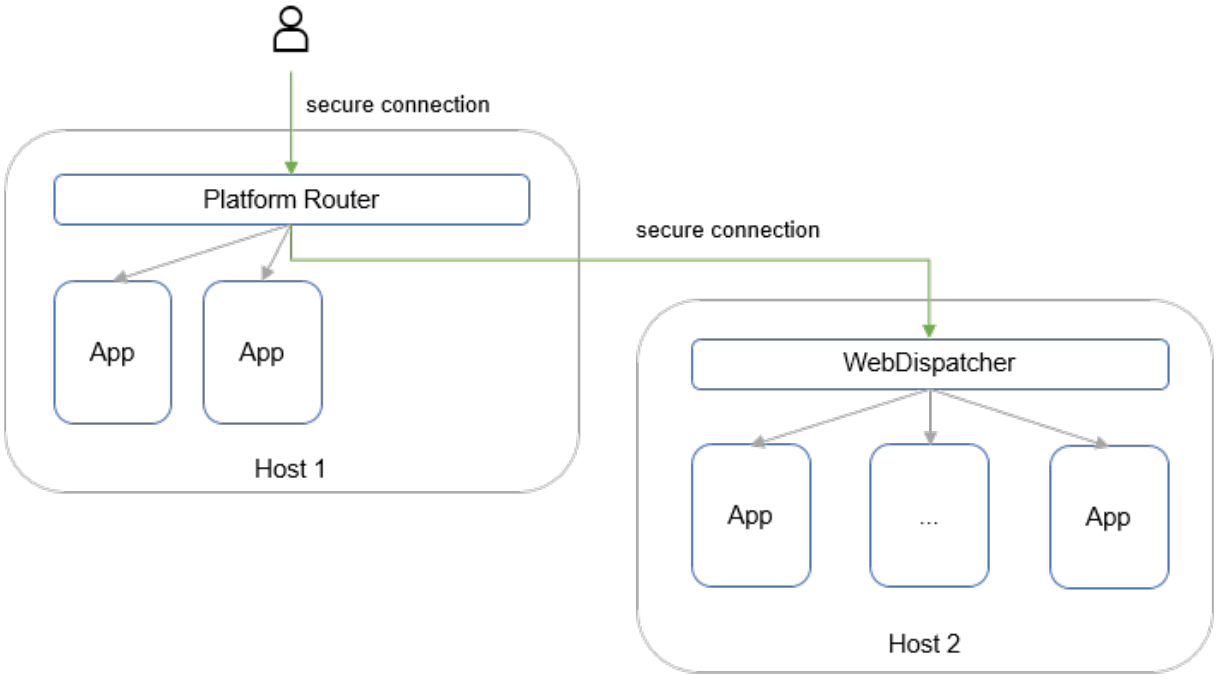
Separating worker and xs_worker Hosts

You can combine both setup variants in the same SAP HANA system, for example, using hosts with both the `worker` and `xs_worker` role alongside hosts with only the `xs_worker` role. This scenario might be useful if you want to isolate particular applications. For more information, see *Maintaining Host Pinning* in *Related Information*.

Inter-Host Communication

The XS advanced Platform Router is the central entry point for all requests to the XS advanced applications, even when XS advanced is scaled out across multiple hosts. To forward requests from the Platform Router to other hosts in a scale-out scenario, an extra SAP WebDispatcher process is automatically started and maintained by XS advanced on the scale-out host. For this reason, no additional custom configuration is

necessary. The Web Dispatcher receives forwarded requests on a secure TLS connection, terminates SSL, and forwards the requests to the appropriate applications instances:



Inter-Host Communication

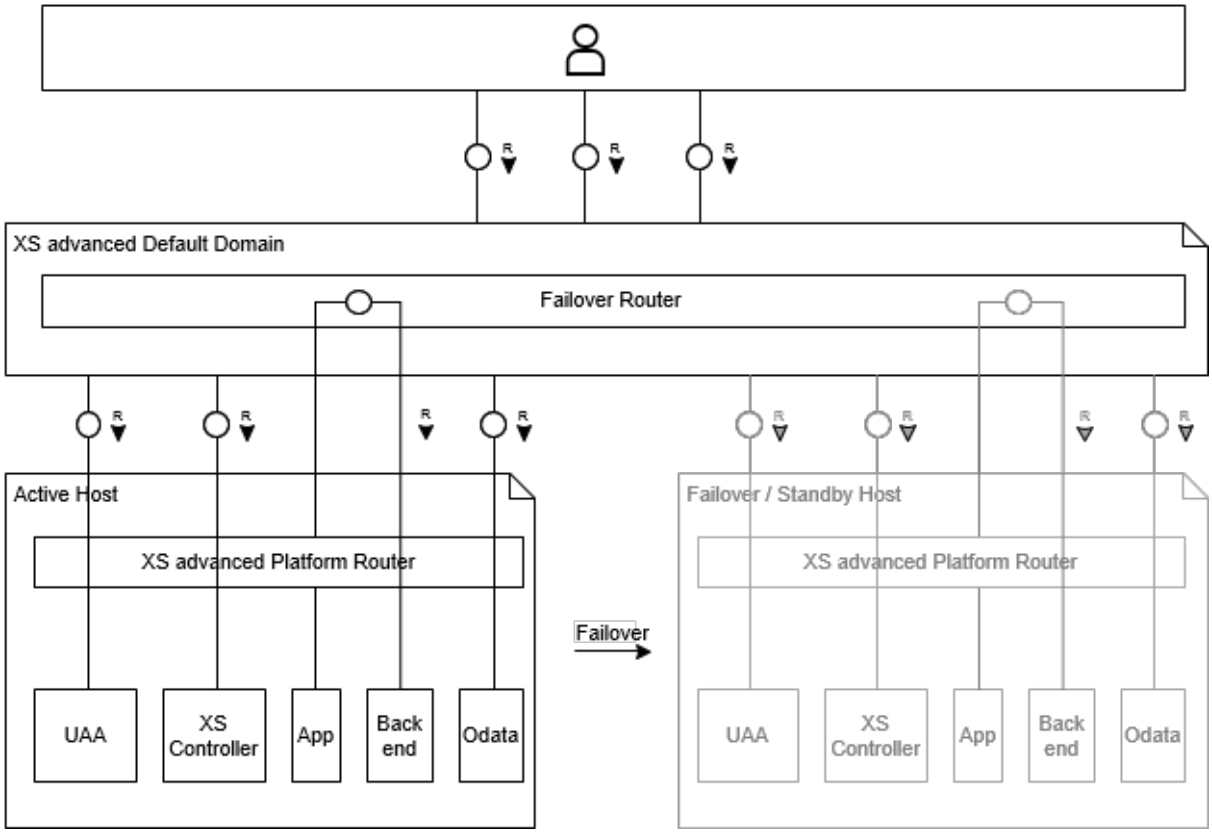
XS Advanced and Host Auto-Failover

In an SAP HANA multi-host system where standby or xs_standby host roles are assigned or where the xs_worker host role is assigned to the master host candidate, host auto-failover might occur with XS advanced. If a host fails, and either a standby host or a master candidate host takes over, you must make sure that requests to the XS advanced platform and any corresponding applications are routed to the correct host even after the failover.

> → Tip
>
> For more information regarding host auto-failover, see *Host Auto-Failover Setup with XS Advanced Run Time* in *Related Information* below.

The following options are available for routing requests:

- If no standby or xs_standby roles are assigned in the system, you can assign the xs_worker role to a host that is not a master candidate to prevent host auto-failover with XS advanced in general. For more information, see *Host Auto-Failover Parameters* in *Related Information* below.
- You can use virtual IP addresse or alias host names and make sure that after the takeover the failover hosts have the same route as the failing hosts.
- You can set up a failover router that ensures that requests to the XS advanced platform and applications are routed to the correct host even after the XS advanced platform router is moved to a different SAP HANA host, as illustrated in the following diagram:



Host Auto-Failover Setup for XS advanced

> → Tip
>
> For more information, see *Setting Up the XS Advanced Run Time Behind a Reverse Proxy* in *Related Information*.

Certificate Setup with a Failover Router

If you terminate SSL at the reverse proxy, the failover router must trust both the active and the failover (or standby) hosts. To configure this trust relationship, choose one of the following approaches:

- Create a certificate that includes the names of all the hosts in the SAP HANA system, in the *Subject Alt Names* section.
  Then, install it on the XS advanced Platform Router with `xs set-certificate`

  > → Tip
  >
  > For more information, see *Maintaining Domains in XS Advanced* in *Related Information* below.

- If you are sure that there are no security risks, you can consider turning off host-name validation in the failover router.

For more information about setting up certificates, see *Establishing the Trust Relationship with a Reverse Proxy* in *Related Information* below, and in particular, the sections *Certificate Setup without SSL Termination at Reverse Proxy* and *Certificate Setup with SSL Termination at Reverse Proxy*.

Shutdown in a Multi-Host Setup

It is recommended to use the `sapcontrol` command to shut down a multi-host SAP HANA system, as described in *Starting and Stopping Distributed SAP HANA Systems Using SAPControl* in *Related Information* below. If you want to stop each host individually, for example, with the command `HDB stop`, make sure that you shut down the active master host (the host currently running the master `nameserver` service) last, in order to prevent a master failover during shutdown. This is particularly important when XS advanced services are running on the active master host. During a shutdown in a multi-host scenario, XS advanced application instances switch hosts along with the `nameserver` service, which is an unnecessary operation during shutdown.

Related Information

Maintaining Host Pinning [page 1338]
Adding Host Roles [page 1105]
Setting Up the XS Advanced Runtime Behind a Reverse Proxy
Host Auto-Failover Parameters [page 901]
Maintaining Domains and Certificates in XS Advanced [page 1324]
Host Auto-Failover Setup with XS Advanced Run Time [page 902]
Starting and Stopping Distributed SAP HANA Systems Using SAPControl [page 1124]

15 Data Access

SAP HANA supports the integration of data from many data sources to enrich your applications and deliver in-depth analysis. These include federated queries, data replication, and processes to improve data quality.

This section provides an overview of the tools and technologies that are available with SAP HANA or supported by SAP HANA for data access and data virtualization. For more information about the administration of these technologies, as well as other operations topics, refer to the documentation indicated.

Native Capabilities for Data Access, Integration, and Quality in SAP HANA

Capability	Description	More Information
Data Federation with SAP HANA Smart Data Access (SDA)	SAP HANA SDA enables you to create virtual tables in SAP HANA that point to virtual tables on remote sources, such as SAP ASE, SAP IQ, Hadoop, and Teradata.	See SAP HANA Smart Data Access [page 1519] and SAP HANA Hadoop Integration [page 1676]
Data Replication and Transformation	SAP HANA smart data integration provides the architecture that supports all types of data delivery in SAP HANA: real-time, batch, and federation (SDA). It includes both data replication and data transformation services.	See the documentation for SAP HANA smart data integration and SAP HANA smart data quality option on SAP Help Portal at https://help.sap.com/docs/HANA_SMART_DATA_INTEGRATION.

Data Replication Technologies in the Extended SAP HANA Landscape

Capability	Description	More Information
Trigger-Based Replication	The trigger-based replication method uses the SAP Landscape Transformation (LT) Replication Server component to pass data from the source system to the SAP HANA database target system.	See the documentation for SAP HANA real-time replication on SAP Help Portal at https://help.sap.com/docs/SAP_LANDSCAPE_TRANSFORMATION_REPLICATION_SERVER
SAP HANA Direct Extractor Connection	SAP HANA Direct Extractor Connection (DXC) provides SAP HANA with out-of-the-box foundational data models based on SAP Business Suite entities, and is a data acquisition method as well.	See SAP HANA Direct Extractor Connection Implementation Guide

Capability	Description	More Information
Extraction Transformation Load-Based Replication	Extraction Transformation Load (ETL)-based data replication uses SAP Data Services (also called Data Services) to load relevant business data from SAP ERP to the SAP HANA database. This lets you read the business data on the application layer level.	See the documentation for SAP Data Services on SAP Help Portal at https://help.sap.com/docs/SAP_DATA_SERVICES
Log-Based Replication	SAP Replication Server (SRS) moves and synchronizes transactional data including DML and DDL across the enterprise, providing low impact, guaranteed data delivery, real-time business intelligence, and zero operational downtime.	See the documentation for SAP Replication Server on SAP Help Portal at https://help.sap.com/docs/SAP_REPLICATION_SERVER

Related Information

Important Disclaimer for Features in SAP HANA [page 1677]

15.1 SAP HANA Smart Data Access

SAP HANA smart data access lets you access remote data as if the data were stored in local tables in SAP HANA, without copying the data into SAP HANA. This capability lets you access and integrate data from multiple remote systems in real time.

In SAP HANA, you use linked database or create virtual tables, which point to remote tables in different remote databases, and then write SQL queries in SAP HANA that use these virtual tables. The SAP HANA query processor optimizes these queries by executing the relevant part of the query in the remote database, returning the results of the query to SAP HANA, and then completing the operation. Physical data movement is not supported by SAP HANA smart data access.

Because smart data access is part of the SAP HANA core system, no additional licensing is required to use it. Additional support packages and patches are available for download from the SAP Software Download Center, and are installed using the SAP HANA database lifecyle manager (HDBLCM).

For a list of supported remote source databases and versions, see SAP Note 2600176 - *SAP HANA Smart Data Access Supported Remote Sources*.

Related Information

SAP Note 2600176

15.1.1 Setting Up ODBC Drivers

The communication between SAP HANA and a remote data source is based on the ODBC protocol. To use the protocol, install the appropriate drivers for the databases you want to connect to using SAP HANA smart data access.

Supported Remote Source Databases

For a list of supported remote source databases and versions, see SAP Note 2600176 - *SAP HANA Smart Data Access Supported Remote Sources*.

.ODBC.INI File

SAP HANA smart data access requires that a `.odbc.ini` file exist in the administrator's home directory, even if empty. If using DSN references to create the remote source, create one entry in the `.odbc.ini` file for each remote source.

ODBC Driver Installation Location

Install ODBC driver library files in a location that is searched by the SAP HANA server. Libraries installed in the SAP HANA `exe` directory are found automatically. Libraries installed elsewhere require an entry in the `LD_LIBRARY_PATH` environment variable to point to this location. If SAP HANA is unable to locate the libraries, you may experience messages during SAP HANA smart data access queries stating the driver could not be loaded. To prevent possible interference with software maintenance of the SAP HANA product, we recommended that you install the ODBC files to a location other than the SAP HANA `exe` directory and add the path to the `LD_LIBRARY_PATH` environment variable.

In a scale-out landscape, install the driver on all hosts.

The `LD_LIBRARY_PATH` environment variable is configured by creating or modifying the `.customer.sh` file in the home directory of the SAP HANA administrator user to include the following entry:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/<path_to_driver_directory>
```

Since the `LD_LIBRARY_PATH` environment variable includes the directory path, the `'Driver='` entry in the DSN section of the `.odbc.ini`, or in the CREATE REMOTE SOURCE command only needs to contain the library name. Restart the SAP HANA system to apply the change to the `.customer.sh` file. To validate that the changes in `.customer.sh` have taken effect, logged in as the `<sid>`adm, execute:

```
echo $LD_LIBRARY_PATH
```

Example

If the IQ ODBC libraries are installed in `/usr/sap/sapiq/IQ-16_0/lib64`, the administrator's `$HOME/.customer.sh` file should include:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/sap/sapiq/IQ-16_0/lib64
```

This statement adds the directory `/usr/sap/IQ-16_0/lib64` to the end of the search path SAP HANA uses to find libraries.

HADOOP (ODBC) Driver

See the *SAP HANA Hadoop Integration* guide.

ODBC Driver Software Ownership

We recommend that you do not make the `<sid>`adm user the software the owner of the ODBC driver library files. Create a new user to become the software owner.

Remote Sources

The process to install, configure, and create remote sources depends on the ODBC driver. Refer to the specific driver for details.

Related Information

SAP HANA Hadoop Integration

15.1.1.1 SAP HANA ODBC Driver

SAP HANA smart data access uses the SAP HANA ODBC driver pre-installed with the SAP HANA server to connect to remote SAP HANA databases.

Since the SAP HANA ODBC driver is already installed on the SAP HANA server, you do not need to download or install the ODBC driver. No additional configuration of the driver is required.

Note

The SAP HANA ODBC driver is not available as a separate download.

Related Information

Create an SAP HANA On-Premise Remote Source [page 1563]
Create an SAP HANA Cloud, SAP HANA Database Remote Source [page 1553]

15.1.1.2 SAP IQ ODBC Driver

The SAP IQ ODBC driver (SYBASE IQ NETWORK CLIENT) is available for download from the SAP Software Download Center.

Prerequisites

If the SAP HANA SMART DATA ACCESS 2.0 package is already installed, you should uninstall it first. The official SAP IQ driver (SYBASE IQ NETWORK CLIENT) should be used instead of the drivers provided in the SAP HANA SMART DATA ACCESS 2.0 package.

To uninstall the SAP HANA SMART DATA ACCESS 2.0 package, execute the `uninstall.sh` script included in the package:

```
./uninstall.sh -s <SID>
```

Alternatively, copy the `uninstall.sh` file to the `federation` folder (for example, `/hana/shared/<SID>/federation/`) and then uninstall the package using the SAP HANA database lifecycle manager (HDBLCM).

Procedure

1. Download and extract the maintenance TAR file to a temporary location on the SAP HANA host.
2. Change to the directory where the SYBASE IQ NETWORK CLIENT installer is located.
3. Start the installer and follow the instructions:

   ```
   ./setup.bin
   ```

4. When installation is complete, use the `find` command to verify that the ODBC driver is installed. By default, the ODBC driver (`libdbodbc17_r.so` library file) is installed in `<installation_path>/IQ-<version>/lib64`. Note the full path as it is needed in future steps.

   ```
   find <installation_path> -name libdbodbc17_r.so
   ```

5. Switch to the `<sid>`adm $HOME directory.
6. Create or modify the `.customer.sh` file to add the following line:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
   ```

   where `<odbc_path>` is the path to the ODBC driver.

7. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

> **Note**
>
> As an alternative to steps 5 to 7, you can copy the ODBC driver, and the common crypto library if needed, to `/usr/sap/<SID>/SYS/global/hdb/federation` (create the directory if it does not already exist).

Related Information

SAP Software Download Center
Uninstalling SAP HANA Components
Create an SAP IQ Remote Source [page 1573]

15.1.1.3 SAP IQ ODBC Driver for SAP HANA Cloud, Data Lake Relational Engine

The SAP IQ ODBC driver for SAP HANA Cloud, data lake Relational Engine is contained in the SAP HANA Data Lake Client package, which is available for download from the SAP Software Download Center.

Procedure

1. In the SAP Software Download Center, enter *Data Lake* in the *Search in Software Downloads* field.

   You can also enter a more specific search query, for example, *SAP HANA Data Lake Client 2024 QRC3*.

   Choose the latest version of the SAP HANA Data Lake Client package.

   > **Note**
   >
   > SAP HANA Data Lake Client 2024 QRC3 (Build 17.1.243.4756) or later is required.

2. Download and extract the maintenance TAR file to a temporary location on the SAP HANA host.
3. Change to the directory where the HANA DATA LAKE CLIENT installer is located.
4. Start the installer and follow the instructions:

   ```
   ./setup.bin
   ```

5. When installation is complete, use the `find` command to verify that the ODBC driver is installed. By default, the ODBC driver (`libdbodbc17_r.so` library file) is installed in `<installation_path>/IQ-<version>/lib64`. Note the full path as it is needed in future steps.

   ```
   find <installation_path> -name libdbodbc17_r.so
   ```

6. Switch to the <sid>adm $HOME directory.
7. Create or modify the `.customer.sh` file to add the following line:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
```

   where <odbc_path> is the path to the ODBC driver.
8. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

SAP Software Download Center
Create an SAP HANA Cloud, Data Lake Relational Engine Remote Source [page 1560]

15.1.1.4 SAP Adaptive Server Enterprise ODBC Driver

The SAP Adaptive Server Enterprise (ASE) ODBC driver (SDK FOR SAP ASE) is available for download from the SAP Software Download Center.

Prerequisites

If the SAP HANA SMART DATA ACCESS 2.0 package is already installed, you should uninstall it first. The official SAP Adaptive Server Enterprise (ASE) ODBC driver (SDK FOR SAP ASE) should be used instead of the drivers provided in the SAP HANA SMART DATA ACCESS 2.0 package.

To uninstall the SAP HANA SMART DATA ACCESS 2.0 package, execute the `uninstall.sh` script included in the package:

```
./uninstall.sh -s <SID>
```

Alternatively, copy the `uninstall.sh` file to the `federation` folder (for example, `/hana/shared/<SID>/federation/`) and then uninstall the package using the SAP HANA database lifecycle manager (HDBLCM).

Procedure

1. Download and extract the maintenance TAR file to a temporary location on the SAP HANA host.
2. Change to the directory where the SDK FOR SAP ASE installer is located.
3. Start the installer and follow the instructions:

```
./setup.bin
```

4. When installation is complete, use the `find` command to verify that the ODBC driver is installed. By default, the ODBC driver (`libsybdrvodb-sqllen8` or `libsybdrvodb.so` library file) is installed in `<installation_path>/DataAccess64/ODBC/lib`. Note the full path as it is needed in future steps.

   ```
   find <installation_path> -name libsybdrvodb-sqllen8.so
   ```

   If the `libsybdrvodb-sqllen8.so` library file does not exist in the installed directory, create a symbolic link:

   ```
   ln -sf libsybdrvodb.so libsybdrvodb-sqllen8.so
   ```

5. Switch to the `<sid>`adm $HOME directory.
6. Create or modify the `.customer.sh` file to add the following line:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
   ```

   where `<odbc_path>` is the path to the ODBC driver.
7. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

> **Note**
>
> As an alternative to steps 5 to 7, you can copy the ODBC driver, and the common crypto library if needed, to `/usr/sap/<SID>/SYS/global/hdb/federation` (create the directory if it does not already exist).

Related Information

SAP Software Download Center
Uninstalling SAP HANA Components
SAP Note 2310113
Create an SAP Adaptive Server Enterprise Remote Source [page 1577]

15.1.1.5 SAP Event Stream Processor ODBC Driver

SAP Event Stream Processor (ESP) on Linux ODBC driver is included in the ESP software package, and is available for download from the SAP Software Download Center.

Prerequisites

Java is installed on the SAP HANA host.

Context

The ESP ODBC driver can be installed using the *Custom* option in the full ESP installer, or from the `.../archives/esp_odbc` folder where you extracted the ESP package. If Java is not installed, and you use the full installer, the installation appears to run successfully, but the ODBC driver is not installed. If you install from the `esp_odbc` folder, an error message appears and the installation fails if Java is not installed.

Procedure

1. Download and extract the ESP package to a temporary location on the SAP HANA host.
2. Download and install the unixODBC driver manager, which is included in the SAP Event Stream Processor installation package or available from your Linux provider.
3. Logged on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `location where you extracted the ESP package.`
4. Change to the `../archives/esp_odbc` folder and type:

   ```
   ./setup.bin
   ```

5. When installation is complete, use the `find` command to verify that the ODBC driver is installed. By default, the ODBC driver (`streamingpsqlodbca_lib.so`) is installed to `/opt/sybase/lib`. Note the full path as it is needed in future steps.

   ```
   find /opt/sybase/lib -name libstreamingpsqlodbca_lib.so
   ```

6. Switch to the `<sid>`adm $HOME directory.
7. Create or modify the `.customer.sh` file to add the following lines:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>/lib
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>/STREAMING-1_0/odbc
   ```

   where `<odbc_path>` is the path to the ODBC driver.
8. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

SAP Software Download Center
Create an SAP Event Stream Processor Remote Source [page 1581]

15.1.1.6 SAP HANA Streaming Analytics ODBC Driver

SAP HANA streaming analytics ODBC driver is installed as part of the streaming analytics installation or the streaming analytics client, and is available for download from the SAP Software Download Center.

Context

You can create a remote source to a streaming analytics system running on another host or on the current host. If the remote source points to the current host, the streaming analytics ODBC driver is already installed as part the streaming analytics instance. If the remote source points to a different host, install the streaming analytics client. The version of the client must be the same as the current host, not the remote host. For example, the current host is running SAP HANA 2.0 SPS 00, while the remote host is running SAP HANA 1.0 SPS 12. You would install the 2.0 client, not the SPS 12 client.

Procedure

1. If required, install the SAP HANA streaming analytics client, which includes the ODBC driver.
2. Download and install the unixODBC driver manager, which is included in the streaming analytics installation package or available from your Linux provider.
3. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm).
4. Switch to the `<sid>`adm $HOME directory.
5. Create or modify the `.customer.sh` file to add the following lines.

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$STREAMING_HOME/odbc
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$STREAMING_HOME/lib
   ```

   > **Note**
   >
   > If streaming analytics is already installed, these entires may already exist.

6. Restart the SAP HANA instance to apply the changes to the `.customer.sh` file.

Related Information

SAP Software Download Center
Create an SAP HANA Streaming Analytics Remote Source [page 1584]

15.1.1.7 SAP MaxDB ODBC Driver

The SAP MaxDB ODBC driver is included in the SAP MaxDB LinuxDB server installation package, which is available for download from the SAP Software Download Center.

Procedure

1. Download and extract the client installation files to a temporary location on the SAP HANA host.
2. On the SAP HANA host, logged on as the SAP HANA software owner (`<sid>`adm), switch to the location of the extracted files.
3. Type:

   ```
   ./SDBINST
   ```

4. When prompted for the software components to install, select *ODBC only*.
5. When installation is complete, use the `find` command to verify that the ODBC driver is installed. By default the ODBC driver (`libsdbodbcw.so` library file) is installed to `/sapdb/clients/MAXDB`.

   ```
   find /sapdb/ -name libsdbodbcw.so
   ```

6. Change to the `<sid>`adm $HOME directory.
7. Add the following line to the `.customer.sh` file.

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
   ```

   where `<odbc_path>` is the path to the ODBC driver.
8. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

SAP Software Download Center
Create an SAP MaxDB Remote Source [page 1587]

15.1.1.8 ODBC Driver for ABAP

The ODBC driver for ABAP is available for download from the SAP Software Download Center.

Procedure

1. In the SAP Software Download Center, search for the ODBC DRIVER FOR ABAP 1.0 component and download the appropriate SAR file.

   Make sure that you download the correct SAR file for your Linux operating system.
2. Extract the SAR file to a directory of your choice (for example, `/home/<myuser>/ODBC_driver_for_ABAP`).

   The directory should now contain at least the following files:
   - `ODBC_driver_for_ABAP.so`
   - `libicudata65.so`
   - `libicudecnumber.so`
   - `libicui18n65.so`
   - `libicuuc65.so`
3. Switch to the `<sid>`adm $HOME directory.
4. Create or modify the `.customer.sh` file to add the following line:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
   ```

   where `<odbc_path>` is the path to the ODBC driver for ABAP.
5. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

SAP Software Download Center
Create an ABAP Cloud Remote Source [page 1555]
Create an ABAP On-Premise Remote Source [page 1568]

15.1.1.9 Teradata ODBC Driver

Teradata drivers are packaged and distributed as a Linux RPM file. You can obtain these RPMs from Teradata.

Procedure

1. Download and extract the files to a temporary location on the SAP HANA host.
2. On the SAP HANA host, logged on as the SAP HANA software owner (`<sid>`adm), install the extracted files.
   - For version 13, installation of `TeraGSS_redhatlinux-i386__linux_i386.13.10.00.06-1.tar.gz` normally fails. While not required for the ODBC installation, the installation attempt prevents `sudo rpm -i tdodbc-13.10.00.04-1.noarch.rpm` from failing with missing dependencies, which is required as part of the driver installation.
   - For version 14, run `sudo rpm -i tdicu-14.10.00.04-1.noarch` to prevent a dependency error.
3. Change the default Kerberos 5 setup.

   The Teradata driver loads GSS API libraries from the OS folders, which conflict with the version of libraries loaded by SAP HANA during installation. Since SAP HANA does not support single sign-on for Teradata remote sources, you can safely disable the Kerberos 5 mechanism.

   a. Edit the `/<installaton_path>/teragss/site/TdgssUserConfigFile.xml` file and add:

   ```
   <Mechanism Name="KRB5">
       <MechanismProperties MechanismEnabled="no" />
   </Mechanism>
   ```

   b. Remove the `/<installaton_path>/teragss/site/linux-x8664/<version>/TdgssUserConfigFile.xml` file, if it exists.

   c. As sudo rmp, execute:

   ```
   /opt/teradata/teragss/linux-x8664/client/bin/run_tdgssconfig
   ```

Related Information

Create a Teradata Remote Source [page 1590]

15.1.1.10 SQL Server ODBC Driver

Remote sources for SQL Server databases require the Microsoft ODBC driver for SQL Server.

Procedure

1. Download and extract the Microsoft ODBC driver package to a temporary location on the SAP HANA host.
2. Download and install the unixODBC driver manger as specified by SQL Server.
3. On the SAP HANA host, logged on as the SAP HANA software owner (`<sid>`adm), install the Microsoft ODBC driver.
   The driver installation creates the following directory: `/opt/microsoft/msodbcsql/lib64`.
4. Change to the `<sid>`adm $HOME directory.
5. Add the following line to the `.customer.sh` file:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
   ```

   where `<odbc_path>` is the path to the ODBC driver.

   Depending on your setup, you might also need to add `/usr/local/lib64` to the SAP HANA administrator user's `PATH` variable.
6. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

Create an SQL Server Remote Source [page 1592]

15.1.1.11 IBM DB2 Driver

IBM DB2 drivers are available on the IBM website.

Procedure

1. Download and extract the IBM DB2 ODBC package to a temporary location on the SAP HANA host.
2. Download and install the unixODBC driver manager as specified by IBM DB2.
3. Log on to the SAP HANA system as the root user.
4. Create the following directory: `/opt/ibm/db2` and extract the ODBC package there.
5. Change to directory: `/opt/ibm/db2/odbc_cli/clidrriver/cfg`.
6. Add a new entry to the `db2cli.ini` file, replacing the applicable information for your system.

Sample Code

```
[TEST_DB2]
 Database=<database_name>
 Protocol=TCPIP
 Port=50010
 Hostname=<machine_name>
```

7. Use the `find` command to verify the ODBC driver is installed. By default, the ODBC driver (`libdb2o.so.1` library file) is installed to `/opt/ibm/db2/odbc_cli/clidriver/lib`. Note the full path as it is needed in future steps.

   ```
   find /opt/ibm -name libdb2o.so.1
   ```

8. Log on as the `<sid>`adm user and switch to the `<sid>`adm $HOME directory.
9. Create or modify the `.customer.sh` file to add the following line:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
   ```

   where `<odbc_path>` is the path to the ODBC driver.
10. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

Create an IBM DB2 Remote Source [page 1595]

15.1.1.12 Oracle Database ODBC Driver

Remote sources for Oracle databases require the ODBC driver for Oracle.

Context

If you are configuring SAP HANA Smart Data Access (SDA) to Oracle SuperCluster, the unixODBC driver also needs to be configured as described in SAP Knowledge Base Article 2501150 - *HANA SDA connection to Oracle fails with ORA-12504*.

Procedure

1. Download and extract the Oracle ODBC driver package to a temporary location on the SAP HANA host.
2. Download and install the unixODBC driver manager as specified by Oracle.
3. Install the Oracle ODBC driver package.

4. Change to `<install_path>/client64/lib` and verify it contains the ODBC driver `libsqora.so.<version>`.
5. Switch to the SAP HANA software owner (`<sid>adm`), then change to the `<sid>adm` $HOME directory.
6. Create a new `tnsnames.ora` file in the home folder. For more information, see *Create an Oracle Remote Source*. For details on the `tnsnames.ora` file, refer to the Oracle documentation.
7. Create or modify the `.customer.sh` file to add the following lines:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/<install_path>/client64/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
export TNS_ADMIN=~/
```

8. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

SAP Knowledge Base Article 2501150
Create an Oracle Remote Source [page 1597]

15.1.1.13 IBM Netezza Driver

IBM Netezza drivers are available on the IBM website.

Procedure

1. Download and extract the IBM Netezza ODBC package to a temporary location on the SAP HANA host.
2. Download and install the unixODBC driver manager as specified by IBM Netezza.
3. Log on to the SAP HANA system as the root user.
4. Create a directory: `/usr/local/nz` and extract the ODBC package.
5. Change to the `linux64` folder within the path of the extracted package and type:

```
./unpack
```

6. Use the `find` command to verify the ODBC driver is installed. By default, the ODBC driver (`libnzodbc.so` library file) is installed to `usr/local/nz/lib64`. Note the full path as it is needed in future steps.

```
find /usr/local/nz -name libnzodbc.so
```

7. Log on as the `<sid>adm` user and switch to the $HOME directory.
8. Create or modify the `.customer.sh` file to add the following line:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<odbc_path>
```

where `<odbc_path>` is the path to the ODBC driver.

9. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

Create an IBM Netezza Remote Source [page 1600]

15.1.1.14 Google BigQuery ODBC Driver

Remote sources for Google BigQuery databases require the ODBC driver for Google BigQuery.

Procedure

1. Download the Linux 64-bit Simba ODBC driver for Google BigQuery from the Google Cloud Platform site.
2. Download and install the unixODBC driver manager as specified by Google BigQuery.
3. On the SAP HANA host, logged on as the SAP HANA software owner (`<sid>`adm), extract the file using the --directory=`/<install_path>` option.
4. Change to the folder you extracted the file to and edit the `simba.googlebigqueryodbc.ini` file. Change the following parameters:
   - Set DriverManagerEncoding to use UTF-16. You will be unable to connect to the Google BigQuery database if this parameter is not set.
   - Set ErrorMessagesPath to point to `/<install_path>/simba/googlebigqueryodbc/ErrorMessages`
   - Add `ODBCInstLib=libiodbcinst.so`
5. Create or modify the `customer.sh` file to add the following lines:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/<install_path>/simba/
googlebigqueryodbc/lib/64
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<path_to_unixODBC_library>
```

6. Restart the SAP HANA system to apply the change to the `.customer.sh` file.

Related Information

Create a Google BigQuery Remote Source [page 1603]

15.1.2 Managing Privileges

Various privileges are required to manage remote sources, virtual tables, and linked database.

Remote Source

- Creating a remote source requires the CREATE REMOTE SOURCE system privilege.
- Managing a remote source created by another user requires additional object level privileges on the remote source. No privileges are required to manage your own remote sources.
- Managing other user's credentials on the remote source requires the CREDENTIAL ADMIN system privilege. No privileges are required to manage your own credentials.

Virtual Tables

- Creating virtual tables requires the CREATE VIRTUAL TABLE object level privilege on the remote source.
- Managing virtual tables owned by other users requires additional object level privileges (for example, INSERT, UPDATE, DELETE). No privileges are required to manage your own virtual tables.

Linked Database

Using and managing linked database requires the LINKED DATABASE object-level privilege on the remote source, regardless of who created the remote source. This requirement allows ad-hoc linked database access using a three part name (`<remote_source>.<schema_name>.<remote_table_name>`).

This privilege, validated on the SAP HANA side, controls access to the linked database feature. Granular access control checks are performed on the remote database system to avoid duplication of access control mechanisms. While a technical user can be used for linked database, it is recommended that secondary credentials be used for any linked database operations. Use CREDENTIAL management commands to manage the remote database user performing DML operations.

Related Information

GRANT Statement (Access Control)

15.1.3 Managing Credentials

The following credential types (or credential modes) are supported for accessing a remote source: technical user, secondary credentials, and single sign-on with JSON Web Tokens.

Credential Type	Description
Technical user	A valid user and password in the remote database. This valid user is used by anyone using the remote source.
Secondary credentials (preferred option)	A unique access credential on the remote source assigned to a specific user.
Single sign-on with JSON Web Tokens	A JSON Web Token (JWT) is used to authenticate the user on the remote SAP HANA system. Users do not need to provide a password (SSO mode).

Both the technical user and secondary credentials types can be set up with either a password or Kerberos single sign-on.

Related Information

Managing Secondary Credentials [page 1536]
Managing Single Sign-On (SSO) with JSON Web Tokens [page 1545]
Managing Single Sign-On (SSO) with Kerberos [page 1541]
Change the Credential Type of a Remote Source [page 1548]

15.1.3.1 Managing Secondary Credentials

Secondary credentials let you assign different credentials to different users when using a remote source.

Accessing a remote database requires valid credentials on the database. All actions performed on the remote database are executed using these credentials and the privileges associated with them.

When creating a remote source, you can define a remote user name and password, called the technical user. With this configuration, all users using the remote source use the same technical user credentials to access to the remote database. If a user with secondary credentials accesses a remote source with a technical user defined, then the technical user credentials are used and the secondary credentials are ignored.

If you want to associate different remote credentials with individual users, then configure the remote source to use secondary credentials. With this configuration, users without secondary credentials can't access the remote source.

Dropping a remote source automatically drops all credentials, including secondary credentials associated with the remote source.

No privileges are required to manage your own credentials, but the CREDENTIAL ADMIN privilege is required to manage other credentials. Management of technical user credentials can be done by the owner of the remote source, or any user with the CREATE REMOTE SOURCE or CREDENTIAL ADMIN privilege.

15.1.3.1.1 Create Secondary Credentials for a User

Create secondary credentials for use with a remote source.

Prerequisites

- Requires the CREDENTIAL ADMIN privilege to manage other credentials. No privileges are required to manage your own credentials.

Context

Though not mandatory, creating the secondary credentials before creating the remote source speeds the process. Secondary credentials can only be created using SQL.

Procedure

In an SQL console window on the local host, create secondary credentials for each user to access the remote source. The specified remote user must exist on the remote source.

```
CREATE CREDENTIAL FOR USER <HANA_user> COMPONENT 'SAPHANAFEDERATION' PURPOSE
    '<remote__source_name>' TYPE 'PASSWORD' USING
    'user=<remote_user_name>;password=<remote_user_password>';
```

Example

This example creates secondary credentials for user1 to access remote source HANA2. User1 is assigned the credentials of user2 on the remote source HANA2.

```
CREATE CREDENTIAL FOR USER user1 COMPONENT 'SAPHANAFEDERATION' PURPOSE 'HANA2'
  TYPE 'PASSWORD' USING 'user=user2;password=password1234';
```

Related Information

CREATE CREDENTIAL Statement (Access Control)

15.1.3.1.2 Delete Secondary Credentials for a User

Delete the secondary credentials assigned for use with a remote source.

Prerequisites

- Requires the CREDENTIAL ADMIN privilege to manage other credentials. No privileges are required to manage your own credentials.

Context

Secondary credentials can only be deleted using SQL.

Procedure

In an SQL console window on the local host, delete the secondary credentials for a specific user and remote source:

```
DROP CREDENTIAL FOR USER <HANA_user> COMPONENT 'SAPHANAFEDERATION' PURPOSE
    '<remote__source_name>' TYPE 'PASSWORD';
```

Example

This example deletes the secondary credentials assigned to user1 for the remote source HANA2:

```
DROP CREDENTIAL FOR USER user1 COMPONENT 'SAPHANAFEDERATION' PURPOSE 'HANA2'
  TYPE 'PASSWORD';
```

Related Information

DROP CREDENTIAL Statement (Access Control)

15.1.3.1.3 Create a Remote Source Using Secondary Credentials

Create a remote source that uses secondary credentials to control access at the user level.

Prerequisites

- Requires the CREATE REMOTE SOURCE system privilege to manage remote sources.

Create a Remote Source Using SQL Syntax

Context

Secondary credentials do not have to exist before you create the remote source without credentials, but you won't be able to connect to the remote source until they are created.

Procedure

In an SQL console window on the local host, create a remote source without credentials. The CONFIGURATION syntax to create a remote source depends upon the connection mode and ODBC driver.

```
CREATE REMOTE SOURCE <remote_source_name> ADAPTER <ODDBC_driver>
CONFIGURATION <driver_specific_syntax>;
```

Create a Remote Source Using the SAP HANA Database Explorer

Context

When creating the remote source, if secondary credentials for at least one user do not already exist, you can't save the remote source with the credential mode set to secondary credentials. Save the remote source with the credential mode set to *None*, create credentials for at least one user, and then modify the remote source to use secondary credentials.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select the applicable adapter.
5. Enter the properties for the specified adapter. For adapter-specific properties, see *Managing Remote Sources*.
6. In the *Credentials Mode* dropdown list, select *Secondary Credentials*.
7. Choose *Create*.

Related Information

Creating Remote Sources [page 1550]

15.1.3.1.4 Convert an Existing Remote Source to Use Secondary Credentials

Change the credentials mode of an existing remote source to use secondary credentials.

Prerequisites

- Requires the CREATE REMOTE SOURCE system privilege to manage remote sources.

Convert a Remote Source Using SQL Syntax

Procedure

Execute:

```
ALTER REMOTE SOURCE <remote__source_name> DROP CREDENTIAL TYPE 'PASSWORD';
```

Convert a Remote Source Using the SAP HANA Database Explorer

Context

When converting the remote source, if secondary credentials for at least one user do not already exist, then you cannot save the remote source with the credential mode set to secondary credentials. Save the remote source with the credential mode set to *None*, create credentials for at least one user, and then modify the remote source to use secondary credentials.

Procedure

1. Expand the *Catalog*.
2. Select *Remote Sources*.
   A list of remote sources appears in the catalog browser item list.
3. Select your remote source.
   The remote source editor opens.
4. Choose *Edit*.
5. In the *Credentials Mode* dropdown list, select *Secondary Credentials*.
6. Choose *Save*.

15.1.3.2 Managing Single Sign-On (SSO) with Kerberos

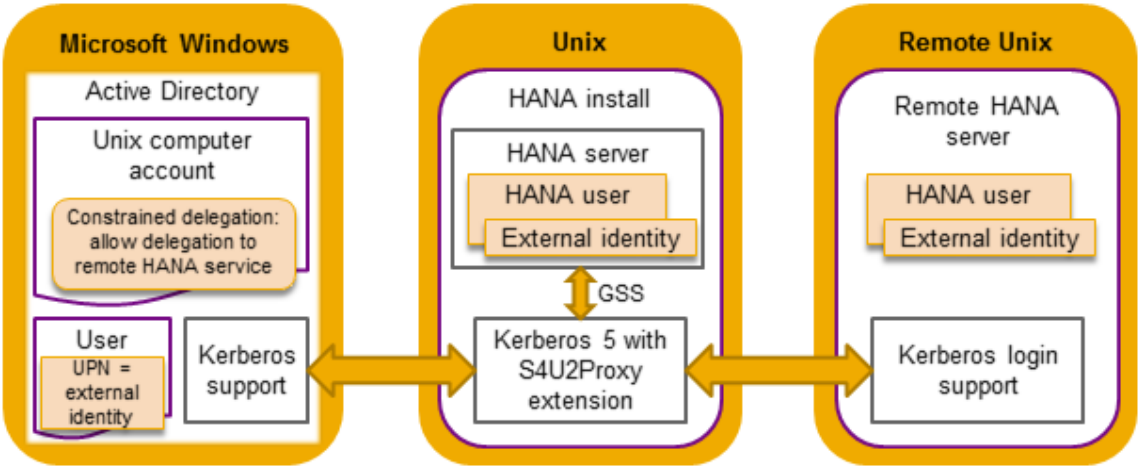
SAP HANA smart data access supports single sign-on with Kerberos for connections to SAP HANA on-premise remote sources. It is not supported for remote sources to SAP HANA databases in SAP HANA Cloud.
Using Kerberos constrained delegation and protocol transition allows SAP HANA users to be authenticated automatically on Microsoft Windows Active Directory, without having to provide a password (SSO mode).

Prerequisites

Microsoft Windows Server, version 2003 or later.

Architecture Overview

The Kerberos platform architecture used in SSO authentication for connections to SAP HANA remote sources is shown below. Protocol transition is assured by Kerberos 5's S4U2Proxy extension:



1. The local SAP HANA server is authenticated only once as a computer using the `krb5_host.keytab`.
2. Users sign on to the local SAP HANA server using an authentication protocol.
3. The local SAP HANA server requests a constrained delegation ticket in its name for the SAP HANA user external identity.
4. The connection to the remote SAP HANA server is authenticated with the constrained delegation ticket.
5. In the authentication validation process, the constrained delegation ticket is validated against the remote SAP HANA service SPN.

Kerberos 5 is installed automatically together with SAP HANA. It contains the S4U (Service for User) extension needed for user impersonation and constrained delegation. Constrained delegation means that delegation can be done only to a predefined set of services. For the purposes of protocol transition, the computer on which the server is installed needs to be entrusted by the Microsoft Windows Active Directory for delegation.

Note that the Kerberos platform is used in SAP HANA for authentication only and not for session management.

Configuration

The Kerberos configuration is defined in the following configuration files:

Configuration File	Description
`<sidadm home>/etc/krb5_hdb.conf`	Configuration of the Kerberos realm to be used with the SAP HANA server installed under `<SID>`adm.

Configuration File	Description
`<sidadm home>/etc/krb5_hdb.keytab`	List of service keys required to authenticate the services on the Kerberos server.
`<sidadm home>/etc/krb5_host.keytab`	One entry only to authenticate the host on the Kerberos server for the purpose of delegation.

If the files are present in the `<sidadm home>/etc` folder, the configuration is automatically taken from there, otherwise the default OS configuration in `/etc/krb5.conf` and `/etc/krb5.keytab` is used instead.

For a custom setup of Kerberos, you can overwrite the following variables in `/usr/sap/<SID>/home/.customer.sh`: `KRB5_CONFIG`, `KRB5_KTNAME`, `KRB5_CLIENT_KTNAME`. For example:

Sample Code

```
export KRB5_CONFIG=<conf file>
export KRB5_KTNAME=<hdb keytab file>
export KRB5_CLIENT_KTNAME=<host keytab file>
```

15.1.3.2.1 Configure Kerboros On Local SAP HANA

On the local SAP HANA server, configure Kerberos to support constrained delegation.

Procedure

1. Configure the Kerberos realm to be used with the SAP HANA server and enable delegation by setting the `forwardable` parameter for Kerberos service tickets to `true` in the `krb5_hdb.conf` file.
2. On the Microsoft Windows Active Directory server, create a Windows Domain account for the SAP HANA server computer and map a host service principal name (SPN) to it.
3. Add the `hdb` service of a remote SAP HANA server to a Microsoft Windows Active Directory account in order to be able to log in to the remote SAP HANA server using Kerberos. Enable constrained delegation and protocol transition for your remote SAP HANA server in the Active Directory Users and Computers application.
4. Add a keytab entry for the `hdb` service. The keytab stores the keys needed by the SAP HANA server to take part in the authentication protocol.

Results

For more information about how to set up SSO for SAP HANA smart data access using Kerberos and Microsoft Windows Active Directory, see the **SAP HANA Smart Data Access Single Sign-On Guide** attached to SAP Note 2303807.

15.1.3.2.2 Create a Remote Source Using Kerberos Authentication

Connect users to an SAP HANA remote source using single sign-on (SSO) with Kerberos.

Prerequisites

- Requires the CREATE REMOTE SOURCE system privilege to manage remote sources.

Context

Kerberos authentication is primarily used as an alternative to the technical user credential type. You can declare it as either a global credential type for the remote source or as the individual type for a given user. If a user has user level credentials defined and the remote source has global credentials defined, the global credentials are used; the user level credentials are ignored on the remote source.

Procedure

1. Do one of:

To:	Execute:
Create global credentials	`CREATE CREDENTIAL FOR COMPONENT 'SAPHANAFEDERATION' PURPOSE <remote_source_name> TYPE 'KERBEROS';`
Create user level credentials	`CREATE CREDENTIAL FOR USER <user_name> COMPONENT 'SAPHANAFEDERATION' PURPOSE <remote_source_name> TYPE 'KERBEROS';`

2. Create a remote source using the credential type KERBEROS.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)

15.1.3.3 Managing Single Sign-On (SSO) with JSON Web Tokens

SAP HANA smart data access supports single sign-on with JSON Web Tokens (JWT) for connections to SAP HANA on-premise and SAP HANA Cloud, SAP HANA database remote sources. JWT allows SAP HANA users to be authenticated automatically on the remote SAP HANA server, without having to provide a password (SSO mode).

To use JWT in an SAP HANA smart data access scenario, the local SAP HANA system is registered as a JWT identity provider in the remote SAP HANA system. The local SAP HANA system then generates the JWTs that are required to authenticate the users accessing the remote SAP HANA system.

The JWT single sign-on process involves the following high-level steps:

1. Users sign on to the local SAP HANA system.
2. Before creating a connection to the remote SAP HANA server, SAP HANA smart data access (SDA) requests a JWT token for the user in the local SAP HANA system.
3. A JWT is issued in the local SAP HANA system and passed to SDA.
4. SDA creates a connection to the remote SAP HANA server using the JWT.
5. The connection is authenticated by the remote SAP HANA server using the JWT.

Related Information

Set Up the JWT SSO Environment [page 1545]
Create a Remote Source with JWT Authentication [page 1547]
Change the Credential Type of a Remote Source [page 1548]
Single Sign-On Using JSON Web Tokens

15.1.3.3.1 Set Up the JWT SSO Environment

On the local and remote SAP HANA servers, configure a JSON Web Token (JWT) single sign-on (SSO) environment for smart data access.

Context

The local SAP HANA system requires a private key and certificate stored in a dedicated trust store (PSE) to be able to sign the JWTs. The remote SAP HANA server also requires a PSE for JWT authentication.

The local SAP HANA system is registered as a JWT provider in the remote SAP HANA system. The name of the claim in the JWT token to be used for mapping the SAP HANA user to an external user name is defined when creating the JWT provider in the remote SAP HANA system.

The database users in the remote SAP HANA system are created with an external identity. During the authentication process, the name of the database user in the local SAP HANA system is matched with the external identity of the remote user.

Note that the standard SAP HANA JWT SSO feature is used in the remote SAP HANA server. For more information, see *Single Sign-On Using JSON Web Tokens*.

Procedure

1. On the local SAP HANA server, set up JWT SSO as follows:
   a. Create a PSE for signing the JWT:

```
CREATE PSE <pse_name>;
ALTER PSE <pse_name> SET OWN CERTIFICATE '
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
... (same as the certificate in the remote database)
-----END CERTIFICATE-----';
SET PSE <pse_name> PURPOSE JWT;
```

   Note that you can create the private key and certificate using OpenSSL:

   Sample Code

```
openssl genrsa -out jwtkey.pem 2048
openssl req -new -x509 -key jwtkey.pem -out jwtpub.crt -days 3650 -subj
"/CN=JWT SignTest PublicKey"
```

2. On the remote SAP HANA server, set up JWT SSO as follows:
   a. Create a certificate from the public certificate of the local SAP HANA database:

```
CREATE CERTIFICATE FROM '
-----BEGIN CERTIFICATE-----
... (same as the certificate in the local database)
-----END CERTIFICATE-----
' COMMENT <comment>;
```

   - COMMENT: A string literal. For example, `COMMENT 'JWT_SSO_PUB'`.

   The created certificate is now contained in the CERTIFICATES system view and can be added to a PSE store.

   b. Create a JWT identity provider for the local SAP HANA database:

```
CREATE JWT PROVIDER <jwt_provider_name> WITH ISSUER 'hana://<instance
name>/<database name>' CLAIM 'user' AS EXTERNAL IDENTITY;
```

   - ISSUER: The local SAP HANA database. You can find the instance name and database name in the M_DATABASE monitoring view, in the SYSTEM_ID and DATABASE_NAME columns.
     Example: 'hana://HD1/HD1'
   - CLAIM: The claim name is 'user'. Note that the claim name is not freely selectable, it must be 'user'. It is used as a key in the JWT body. Its value in the JWT is the name of the database user in the local SAP HANA system.

c. Create a PSE for JWT authentication and add the certificate created above to it. Then set the purpose of the newly created PSE to JWT:

```
CREATE PSE <pse_name>;

SELECT CERTIFICATE_ID FROM SYS.CERTIFICATES WHERE COMMENT=<comment>;
ALTER PSE <pse_name> ADD CERTIFICATE <certificate_id>;

SET PSE <pse_name> PURPOSE JWT FOR PROVIDER <jwt_provider_name>;
```

d. Create a user and enable the JWT authentication mechanism for it:

```
CREATE USER <user_name> PASSWORD <password>;
ALTER USER <user_name> ADD IDENTITY <mapped_user_name> FOR JWT PROVIDER
<jwt_provider_name>;
ALTER USER <user_name> ENABLE JWT;
```

- (EXTERNAL) IDENTITY: A JWT provider-user mapping. The mapped JWT user name should be the name of the user in the local SAP HANA database who creates the virtual tables.

Related Information

CREATE CERTIFICATE Statement (System Management)
CREATE PSE Statement (System Management)
ALTER PSE Statement (System Management)
SET PSE Statement (System Management)
CREATE JWT PROVIDER Statement (Access Control)
ALTER USER Statement (Access Control)
Single Sign-On Using JSON Web Tokens

15.1.3.3.2 Create a Remote Source with JWT Authentication

Connect users to an SAP HANA remote source using single sign-on (SSO) with JSON Web Tokens (JWT).

Prerequisites

Requires the CREATE REMOTE SOURCE system privilege to manage remote sources.

Procedure

As the administrator, create a remote source to the remote SAP HANA server, with the credential type JWT:

```
CREATE REMOTE SOURCE <remote_source_name> ADAPTER "hanaodbc"
```

```
    CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=<remote_server_name>:3<remote_instance_number>15
;'
    WITH CREDENTIAL TYPE 'JWT';
```

Next Steps

Create users and assign them the CREATE VIRTUAL TABLE object-level privilege on the remote source:

```
CREATE USER <user_name> PASSWORD <password>;
GRANT CREATE VIRTUAL TABLE ON REMOTE SOURCE <remote_source_name> TO <user_name>;
```

When a user tries to create a virtual table using this remote source, the connection to the remote source is created using a JWT token for authentication.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)

15.1.3.4 Change the Credential Type of a Remote Source

Change the credential type of an existing remote source to use a different credential type.

Prerequisites

- Requires the CREATE REMOTE SOURCE system privilege to manage remote sources.
- For secondary credentials, you have created at least one set of secondary credentials.
- For JWT authentication, you have set up the JWT authentication environment.
- For Kerberos authentication, you have set up the Kerberos authentication environment.

Procedure

1. Drop the defined credential type if the remote source has one:

   ```
   ALTER REMOTE SOURCE <remote_source_name> DROP CREDENTIAL TYPE
   '<credential_type>';
   ```

   The credential types are as follows:

- Technical user credential type: PASSWORD
- JWT credential type: JWT
- Kerberos credential type: KERBEROS

2. Assign the new credential type to the remote source:
   - Technical user credential type

     ```
     ALTER REMOTE SOURCE <remote_source_name> WITH CREDENTIAL TYPE 'PASSWORD'
     USING 'user=<user_name>;password=<password>';
     ```

   - JWT credential type

     ```
     ALTER REMOTE SOURCE <remote_source_name> WITH CREDENTIAL TYPE 'JWT';
     ```

   - Kerberos credential type

     ```
     ALTER REMOTE SOURCE <remote_source_name> WITH CREDENTIAL TYPE 'KERBEROS';
     ```

   - Secondary credentials
     To use secondary credentials, do not specifiy a credential type. Secondary credentials must be created for individual users for use with the remote source. For more information, see *Create Secondary Credentials for a User*.

Example

This example drops the technical user credential type for the HANA1 remote source and then assigns the JWT credential type to it:

```
ALTER REMOTE SOURCE HANA1 DROP CREDENTIAL TYPE 'PASSWORD';
ALTER REMOTE SOURCE HANA1 WITH CREDENTIAL TYPE 'JWT';
```

This example drops the Kerberos credential type for the HANA2 remote source and then assigns the technical user credential type to it:

```
ALTER REMOTE SOURCE HANA2 DROP CREDENTIAL TYPE 'KERBEROS';
ALTER REMOTE SOURCE HANA2 WITH CREDENTIAL TYPE 'PASSWORD' USING
'user="user1";password="Password1"';
```

Related Information

Create Secondary Credentials for a User [page 1537]
Managing Single Sign-On (SSO) with JSON Web Tokens [page 1545]
Managing Single Sign-On (SSO) with Kerberos [page 1541]

15.1.4 Managing Remote Sources

Create, modify, and delete remote sources.

15.1.4.1 Creating Remote Sources

Create a new remote source using the SAP HANA database explorer, or SQL syntax. The parameters required vary by adapter.

A remote source can be used to create virtual tables or to use the linked database feature.

For a list of supported databases and versions, see SAP Note 2600176 - *SAP HANA Smart Data Access Supported Remote Sources*.

> **Note**
>
> If the SAP HANA database is configured with a high isolation level, you cannot create a remote source using a DSN entry. See SAP Note 2867216 - *Login Failure to Remote Source From MDC Configured With High Isolation Level*.

15.1.4.1.1 Import Certificates for SSL Connections to Remote Sources

Connections to an SAP HANA database in SAP HANA Cloud, to an SAP HANA Cloud, data lake Relational Engine, or to an ABAP system require an SSL certificate. An SSL certificate can optionally be used for connections to an SAP HANA on-premise system.

Prerequisites

You have downloaded the required certificates:

Remote Source	Certificates

SAP HANA Cloud, SAP HANA database
SAP HANA Cloud, data lake Relational Engine
	Download the DigiCert SSL certificate:
	DigiCert Global Root CA:
	`curl -O https://dl.cacerts.digicert.com/DigiCertGlobalRootCA.crt.pem`
	DigiCert TLS RSA4096 Root G5:
	`curl -O https://dl.cacerts.digicert.com/DigiCertTLSRSA4096RootG5.crt.pem`

ABAP systems	Download the DigiCert SSL certificates:
	DigiCert Global Root CA:
	`curl -O https://dl.cacerts.digicert.com/DigiCertGlobalRootCA.crt.pem`
	DigiCert Global Root G2:
	`curl -O https://dl.cacerts.digicert.com/DigiCertGlobalRootG2.crt.pem`
	DigiCert TLS RSA4096 Root G5:
	`curl -O https://cacerts.digicert.com/DigiCertTLSRSA4096RootG5.crt.pem`

For more information about DigiCert certificates, see SAP Note 3327214, SAP Note 3399573, and SAP Note 3397584.

Procedure

1. Create a certificate collection:

```
CREATE PSE <pse_name>;
```

2. Create a certificate and insert the content of the downloaded certificate.

   If you require more than one SSL certificate, create a separate certificate for each one.

```
CREATE CERTIFICATE <certificate_name> FROM '
-----BEGIN CERTIFICATE-----
<content in certificate>
-----END CERTIFICATE-----';
```

   The created certificate is now contained in the CERTIFICATES system view and can be added to a certificate collection.

3. Add the certificate(s) created above to the certificate collection:

```
ALTER PSE <pse_name> ADD CERTIFICATE <certificate_name>;
```

4. Set the purpose of the newly created PSE to REMOTE SOURCE:

```
SET PSE <pse_name> PURPOSE REMOTE SOURCE;
```

   This makes the usage of the certificates in the PSE store available to all remote sources.

> **Note**
>
> You can restrict the usage of the certificates to a certain remote source as follows (the remote source must already exist):
>
> ```
> SET PSE <pse_name> PURPOSE REMOTE SOURCE FOR REMOTE SOURCE
> <remote_source_name>;
> ```

Related Information

CREATE CERTIFICATE Statement (System Management)
CREATE PSE Statement (System Management)
ALTER PSE Statement (System Management)
SET PSE Statement (System Management)

15.1.4.1.2 Create an SAP HANA Cloud, SAP HANA Database Remote Source

Create a remote source to an SAP HANA database in an SAP HANA Cloud instance.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.
- You have downloaded and imported the SSL certificate signed by DigiCert. See Import Certificates for SSL Connections to Remote Sources [page 1551].

Context

You can use an SAP HANA Cloud, SAP HANA database remote source with virtual tables or the linked database feature.

The following syntax example assumes the remote source is configured to use technical user credentials. To use secondary credentials or other credential types, see *Managing Credentials*.

Create a Remote Source Using SQL Syntax

Procedure

In an SQL console, connected to the local database, execute a CREATE REMOTE SOURCE command specifying all properties.

The following syntax example assumes the remote source is configured to use technical user credentials.

Sample Code

```
CREATE REMOTE SOURCE <remote_source_name> ADAPTER "hanaodbc"
    CONFIGURATION 'Driver=libodbcHDB.so;ServerNode=<HANA_Cloud_endpoint>;
    encrypt=TRUE;'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

To set session-specific connection information during creation, include the sessionVariable in the ServerNode property:

Sample Code

```
ServerNode=<HANA_Cloud_endpoint>;encrypt=TRUE;
```

```
sessionVariable:<session_variable_name>=?
```

For more information, see *Session-Specific Information for Connections*.

Create a Remote Source Using the SAP HANA Database Explorer

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *HANA (ODBC)*.
5. Enter the required information:

Adapter Properties

Property	Value
Server	Server of the remote source. Use the remote source URL endpoint excluding the port number.
Port	Port number of the remote source. Use the port number contained in the URL endpoint.
DML Mode	Specifies if the remote source is readwrite (default) or readonly
Extra Adapter Properties	Additional properties of the remote source. Separate properties by a semicolon. • encrypt=TRUE • (To enable the optional feature on the remote source) sessionVariable:<session_variable_name>=? For more information, see *Session-Specific Information for Connections*.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
7. Choose *Create*.

Related Information

Import Certificates for SSL Connections to Remote Sources [page 1551]
CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
SAP HANA ODBC Driver [page 1521]
Session-Specific Information for Connections [page 1655]
Verify a Remote Source [page 1607]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.3 Create an ABAP Cloud Remote Source

Create a remote source to the ABAP SQL service of an ABAP cloud system, such as SAP S/4HANA Cloud or SAP BTP ABAP environment.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.
- You have downloaded and imported the SSL certificate signed by DigiCert. See Import Certificates for SSL Connections to Remote Sources [page 1551].

Context

An ABAP cloud remote source lets you access ABAP CDS view entities and parameterized ABAP CDS view entities in the remote ABAP system. It can be used with virtual tables or the linked database feature.

Note the following limitations:

- Only the technical user credential type is supported.
- An ABAP remote source is read only. This means that the WITH REMOTE option in the CREATE VIRTUAL TABLE statement is not supported.
- Join relocation does not work, since the remote source is read only.

For troubleshooting purposes, traces for the ODBC driver for ABAP can be configured and managed within the SAP HANA trace framework.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.
3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[ABAP1]
Driver=ODBC_driver_for_ABAP.so
HOST=<abap_host_name>
PORT=<abap_port_number>
CLIENT=000
LANGUAGE=EN
TYPEMAP=semantic
SERVICEPATH=/sap/bc/sql/sql1/sap/s_privileged
UidType=uid
TYPEMAP=semantic
CryptoLibrary=/usr/sap/[sid]/SYS/exe/hdb/libsapcrypto.so
Uid=<user_name>
Pwd=<user_password>
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:

- If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file:

  Sample Code

  ```
  CREATE REMOTE SOURCE <remote_source_name> ADAPTER "abapodbc"
      CONFIGURATION 'DSN=ABAP1'
      WITH CREDENTIAL TYPE 'PASSWORD' USING
  'user=<user_name>;password=<password>';
  ```

- If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties:

  Sample Code

  ```
  CREATE REMOTE SOURCE <remote_source_name> ADAPTER "abapodbc"
      CONFIGURATION
  'uidtype=alias;driver=ODBC_driver_for_ABAP.so;servicepath=/sap/bc/sql/
  sql1/sap/s_privileged;
      host=<abap_host_name>;port=<abap_port_number>;
      language=EN;typemap=semantic'
      WITH CREDENTIAL TYPE 'PASSWORD' USING
  'user=<username>;password=<password>';
  ```

Property	Value
ADAPTER	ABAP adapter name: abapodbc
driver	ODBC driver for ABAP file name: `ODBC_driver_for_ABAP.so`
host	Host address of the ABAP cloud system
port	Port number of the ABAP cloud system: 443
servicepath	Service path to the ABAP SQL service. The path name is formed as follows: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system including name space>` For example: • For services delivered by SAP: `/sap/bc/sql/sql1/sap/<name of SQL service as defined in ABAP system>` • For customer-defined services: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system>`
client	Logon client; this must not be specified for a cloud system
user	ID or alias name of the ABAP user, depending on the value of the `uidtype` property

Property	Value
password	Password of the ABAP user
uidtype (optional)	Either `uid` (default) or `alias` • **`uid`**: The user name in the credential is the ABAP user ID. • **`alias`**: The user name in the credential is the alias name of the ABAP user.
language (optional)	ABAP language code with two characters. Default: EN
typemap (optional)	Either `semantic`, `native`, or `semanticDatsTimsAsWchar` Specifies how ABAP data types are mapped to ODBC SQL data types.

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *ABAP (ODBC)*.
5. Select the connection mode.
6. Depending on the connection mode, enter the required information:

Adapter Properties

Property	Value
Host	Host address of the ABAP cloud system
Port	Port number of the ABAP cloud system: 443
Driver	ODBC driver for ABAP file name: `ODBC_driver_for_ABAP.so`

Adapter Properties

Property	Value
Service Path	Service path to the ABAP SQL service. The path name is formed as follows: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system including name space>` For example: For services delivered by SAP: `/sap/bc/sql/sql1/sap/<name of SQL service as defined in ABAP system>` For customer-defined services: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system>`
Client	Logon client; this must not be specified for a cloud system
Language	ABAP language code with two characters. Default: EN
Type Map	Either `semantic`, `native`Service path to the ABAP SQL service. The path name is formed as follows:, or `semanticDatsTimsAsWchar` Specifies how ABAP data types are mapped to ODBC SQL data types.
Uid Type	Either **`alias`** (default) or **`uid`** • **`alias`**: The user name in the *Credentials* section is the alias name of the ABAP user. • **`uid`**: The user name in the *Credentials* section is the ABAP user ID.
Extra Adapter Properties	Additional properties of the remote source. Separate properties by a semicolon.

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.

7. Specify the credentials mode for the remote source:

Option	Description
Technical User	Specify a valid user and password to connect to the remote source: • *User Name*: ID or alias name of the ABAP user, depending on the value of *Uid Type* • *Password*: Password of the ABAP user All connections to the remote source share the same credentials for the remote source.
None	The remote source is created without credentials. You cannot connect to the remote source until credentials are assigned.

8. Choose *Create*.

Related Information

Verify a Remote Source [page 1607]
ODBC Driver for ABAP [page 1529]
Safe Mode for ODBC Connections [page 1653]

15.1.4.1.4 Create an SAP HANA Cloud, Data Lake Relational Engine Remote Source

Create a remote source to an SAP HANA Cloud, data lake Relational Engine server.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.
- You have downloaded and imported the SSL certificate signed by DigiCert. See Import Certificates for SSL Connections to Remote Sources [page 1551].
- For secondary credentials, you have created at least one set of secondary credentials. For more information, see Managing Secondary Credentials [page 1536].

Context

The following syntax example assumes the remote source is configured to use technical user credentials.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (adm), change to the adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.
3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[HDLRE1]
Driver= libdbodbc17_r.so
Host=<data lake Relational Engine endpoint>
ENC=TLS(direct=yes)
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file:

     Sample Code

     ```
     CREATE REMOTE SOURCE <remote_source_name> ADAPTER "iqodbc"
         CONFIGURATION 'DSN=HDLRE1'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties:

     Sample Code

     ```
     CREATE REMOTE SOURCE <remote_source_name> ADAPTER "iqodbc"
         CONFIGURATION 'Driver=libdbodbc17_r.so;host=<data lake Relational
     Engine endpoint>;
         ENC=TLS(direct=yes)'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

For secondary credentials or no credentials, a credential type is not specified:

```
CREATE REMOTE SOURCE <remote_source_name> ADAPTER "iqodbc"
    CONFIGURATION 'Driver=libdbodbc17_r.so;host=<data lake Relational
Engine endpoint>;
    ENC=TLS(direct=yes)';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *IQ (ODBC)*.
5. Select the connection mode.
6. Depending on the connection mode, enter the required information:

Adapter Properties

Property	Description
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

Extra Adapter Properties	Specifies the additional properties to complete the remote connection, in the format:
	`Driver=libdbodbc17_r.so;`
	`Host=<Data Lake Relational Engine endpoint>;`
	`ENC=TLS(direct=yes)`
	For example:
	`Driver=libdbodbc17_r.so;`
	`host=xxxxxxxx-xxxx-xxxx-xxxx-`
	`xxxxxxxxxxxx.iq.hdl.xxxxxxxxxxxxxxxxxxx.`
	`xxxxxxx.hanacloud.ondemand.com:443;`
	`ENC=TLS(direct=yes)`

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

7. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
8. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
Manage Read-Write and Read-Only Access to a Remote Source [page 1622]
SAP IQ ODBC Driver for SAP HANA Cloud, Data Lake Relational Engine [page 1523]

15.1.4.1.5 Create an SAP HANA On-Premise Remote Source

Create a remote source to another SAP HANA database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.
- To use an SSL connection, you have downloaded and imported the SSL certificate signed by DigiCert. See Import Certificates for SSL Connections to Remote Sources [page 1551].

Context

You can use an SAP HANA remote source with virtual tables or the linked database feature.

An SAP HANA remote source supports failover.

The remote source cannot point to the local SAP HANA instance. This scenario, which is called a loopback, can return unexpected query results and is not a supported scenario.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials or other credential types, see *Managing Credentials*.

Create DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.
3. Define one entry in the `.odbc.ini` file for each remote source. For each entry, specify the driver for the remote source adapter.

   ```
   Driver=libodbcHDB.so
   ```
4. Specify the remote server.

   ```
   ServerNode=<remote_server_name>:3<remote_instance_number>15
   ```
5. To use an SSL connection, enter:

   ```
   encrypt=TRUE
   ```
6. To enable failover, add the failover server name to the `ServerNode` property, separated by a comma.

   ```
   ServerNode=<remote_server_name>:3<remote_instance_number>15,<failover_server_name>:3<failover_instance_number>15
   ```
7. To set session specific connection information, add the `sessionVariable` property.

   ```
   sessionVariable:<session_variable_name>=?
   ```

For more information, see *Session-Specific Information for Connections*.

8. To enable optimized mode, add the `linkeddatabase_mode` property.

```
linkeddatabase_mode=optimized
```

9. Restart the SAP HANA system to apply the changes to the `.INI` file.

Example

```
[HANA1]
Driver=libodbcHDB.so
ServerNode=my_server:30115,my_failover:30215
sessionVariable:APPLICATIONUSER=?
linkeddatabase_mode=optimized
```

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

     Sample Code

     ```
     CREATE REMOTE SOURCE MY_HANA ADAPTER "hanaodbc"
         CONFIGURATION 'DSN=HANA1'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

     To enable failover, define session variables or enable optimized mode for linked database, add the parameters to the serverNode property in the DSN entry in the `.odbc.ini` file.
   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties.

     Sample Code

     ```
     CREATE REMOTE SOURCE <HANA1> ADAPTER "hanaodbc"
     ```

```
    CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=<remote_server_name>:3<remote_instance_
number>15;'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

- To use an SSL connection, enter:

```
encrypt=TRUE
```

Sample Code

```
CREATE REMOTE SOURCE <HANA1> ADAPTER "hanaodbc"
CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=<remote_server_name>:3<remote_instance_
number>15;encrypt=TRUE;'
WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

- To enable failover during creation, specify the failover server information in the ServerNode property.

Sample Code

```
ServerNode=<remote_server_name>:3<remote_instance_number>15,
    <failover_server_name>:3<failover_instance_number>15;
```

- To enable optimized mode for linked database or to set session-specific connection information during creation, include the sessionVariable or lindkeddatabase_mode parameter in the ServerNode property.

Sample Code

```
ServerNode=<remote_server_name>:3<remote_instance_number>15;

sessionVariable:<session_variable_name>=?;linkeddatabase_mode=optimized;
```

For more information, see *Session-Specific Information for Connections*.

- For an existing remote source, to enable failover, set session specific connection information, or enable optimized mode, use ALTER REMOTE SOURCE to add the properties.

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *HANA (ODBC)*.
5. Select the connection mode.
6. Depending on the connection selected, enter the required information:

Adapter Properties

Property	Description
Server	Specifies the server of the remote source. Use the machine name or IP address. To enable failover, include the failover remote source server, separated by a comma.
Port	Specifies the port number of the remote source. Use 3`<instance_number>`15, where `<instance_number>` is the instance number of the remote source. To enable failover, include the failover remote source port number, separated by a comma.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.
Extra Adapter Properties	Specifies additional properties for the remote source. Separate properties by a semicolon. • (To enable the optional SSL connection) encrypt=TRUE • (To enable the optional feature on the remote source) sessionVariable:`<session_variable_name>`=? For more information, see *Session-Specific Information for Connections*. • (To enable the optional feature on the remote source) linkeddatabase_mode=optimized

Data Source Name

Property	Description
To enable failover, session connection information, or optimized mode, add the properties to the DSN section of the `.odbc.ini` file before you create the remote source.	
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

7. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
   - SSO (KERBEROS) - All connections to the remote source are authenticated through Kerberos single sign-on (SSO).

8. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
SAP HANA ODBC Driver [page 1521]
Import Certificates for SSL Connections to Remote Sources [page 1551]
Session-Specific Information for Connections [page 1655]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.6 Create an ABAP On-Premise Remote Source

Create a remote source to the ABAP SQL service of an on-premise ABAP system, such as SAP S/4HANA.

Prerequisites

- The remote ABAP system is based on ABAP Platform 2021 FPS01 or higher, where the ABAP SQL service is available.
- The remote source is reachable by the network from the computer you are using.
- You have the CREATE REMOTE SOURCE system privilege.
- You have downloaded and imported the SSL certificate signed by DigiCert. See Import Certificates for SSL Connections to Remote Sources [page 1551].

Context

An ABAP on-premise remote source lets you access ABAP CDS view entities and parameterized ABAP CDS view entities in the remote ABAP system. It can be used with virtual tables or the linked database feature.

For troubleshooting purposes, traces for the ODBC driver for ABAP can be configured and managed within the SAP HANA trace framework.

Note the following limitations:

- Only the technical user credential type is supported.
- An ABAP remote source is read only. This means that the WITH REMOTE option in the CREATE VIRTUAL TABLE statement is not supported.
- Join relocation does not work, since the remote source is read only.

For troubleshooting purposes, traces for the ODBC driver for ABAP can be configured and managed within the SAP HANA trace framework.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.
3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[ABAP1]
Driver=ODBC_driver_for_ABAP.so
HOST=<abap_host_name>
PORT=<abap_port_number>
CLIENT=000
LANGUAGE=EN
TYPEMAP=semantic
SERVICEPATH=/sap/bc/sql/sql1/sap/s_privileged
UidType=uid
TYPEMAP=semantic
CryptoLibrary=/usr/sap/[sid]/SYS/exe/hdb/libsapcrypto.so
Uid=<user_name>
Pwd=<user_password>
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file:

     Sample Code

     ```
     CREATE REMOTE SOURCE <remote_source_name> ADAPTER "abapodbc"
         CONFIGURATION 'DSN=ABAP1'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties:

     Sample Code

     ```
     CREATE REMOTE SOURCE <remote_source_name> ADAPTER "abapodbc"
         CONFIGURATION 'HOST=<abap_host_name>;PORT=<abap_port_number>;
         driver=ODBC_driver_for_ABAP.so;
         TYPEMAP=semantic;SERVICEPATH=/sap/bc/sql/sql1/sap/
     s_privileged;CLIENT=000;LANG=EN;UidType=uid;'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

Property	Value
ADAPTER	ABAP adapter name: abapodbc
driver	ODBC driver for ABAP file name: `ODBC_driver_for_ABAP.so`
HOST	Host address of the ABAP system
PORT	Port number of the ABAP system

Property	Value
SERVICEPATH	Service path to the ABAP SQL service. The path name is formed as follows: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system including name space>` For example: • For services delivered by SAP: `/sap/bc/sql/sql1/sap/<name of SQL service as defined in ABAP system>` • For customer-defined services: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system>`
Client	Logon client
UidType (optional)	Either `uid` (default) or `alias` • **`uid`**: The user name in the credential is the ABAP user ID. • **`alias`**: The user name in the credential is the alias name of the ABAP user.
TYPEMAP (optional)	Either `semantic`, `native`, or `semanticDatsTimsAsWchar` Specifies how ABAP data types are mapped to ODBC SQL data types.
language (optional)	ABAP language code with two characters. Default: EN
user	ID or alias name of the ABAP user, depending on the value of the `uidtype` property
password	Password of the ABAP user

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *ABAP (ODBC)*.
5. Select the connection mode.
6. Depending on the connection mode, enter the required information:

Adapter Properties

Property	Value
Host	Host address of the ABAP system
Port	Port number of the ABAP system
Driver	ODBC driver for ABAP file name: `ODBC_driver_for_ABAP.so`
Service Path	Service path to the ABAP SQL service. The path name is formed as follows: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system including name space>` For example: For services delivered by SAP: `/sap/bc/sql/sql1/sap/<name of SQL service as defined in ABAP system>` For customer-defined services: `/sap/bc/sql/sql1/<name of SQL service as defined in ABAP system>`
Client	Logon client
Language	ABAP language code with two characters. Default: EN
Type Map	Either `semantic`, `native`, or `semanticDatsTimsAsWchar` Specifies how ABAP data types are mapped to ODBC SQL data types.
Uid Type	Either **`alias`** (default) or **`uid`** • **`alias`**: The user name in the *Credentials* section is the alias name of the ABAP user. • **`uid`**: The user name in the *Credentials* section is the ABAP user ID.
Extra Adapter Properties	Additional properties of the remote source. Separate properties by a semicolon.

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.

7. Specify the credentials mode for the remote source:

Option	Description
Technical User	Specify a valid user and password to connect to the remote source: • *User Name*: ID or alias name of the ABAP user, depending on the value of *Uid Type* • *Password*: Password of the ABAP user All connections to the remote source share the same credentials for the remote source.
None	The remote source is created without credentials. You cannot connect to the remote source until credentials are assigned.

8. Choose *Create*.

Related Information

Verify a Remote Source [page 1607]
ODBC Driver for ABAP [page 1529]
Safe Mode for ODBC Connections [page 1653]

15.1.4.1.7 Create an SAP IQ Remote Source

Create a remote source to an SAP IQ database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An SAP IQ remote source can be used with virtual tables or the linked database feature.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.
3. Define one entry in the `.odbc.ini` file for each remote source. For example:

   ```
   [IQ1]
   Driver= libdbodbc17_r.so
   ServerName=<iq_server_name>
   CommLinks=tcpip(host= <iq_machine_name>;port=<IQ_port>)
   ```
4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

Sample Code

```
CREATE REMOTE SOURCE MY_IQ1 ADAPTER "iqodbc"
    CONFIGURATION 'DSN=IQ1;'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

- If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties.

Sample Code

```
CREATE REMOTE SOURCE MY_IQ2 ADAPTER "iqodbc"
    CONFIGURATION 'Driver=libdbodbc17_r.so;ServerName=<iq_server_name>;
        CommLinks=tcpip(host=<iq_machine_name>;port=<IQ_port>);'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *IQ (ODBC)*.
5. Select the connection mode.
6. Depending on the connection mode, enter the required information:

Adapter Properties

Property	Description
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

Adapter Properties

Property	Description

Extra Adapter Properties	Specifies the additional properties to complete the remote connection, in the format:
	`ServerName=<iq_machine_name>;`
	`CommLinks=tcpip(host=<IQ_host>;port=<IQ_port>`
	For example, the additional properties to connect to the demo database would be:
	ServerName=<iq_machine_name>_iqdemo;
	CommLinks=tcpip(host=<iq_machine_name>;port=2638)

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

7. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
8. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
SAP IQ ODBC Driver [page 1522]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.8 Create an SAP Adaptive Server Enterprise Remote Source

Create a remote source to an SAP Adaptive Server Enterprise (ASE) database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An SAP Adaptive Server Enterprise (AES) remote source can be used with virtual tables or the linked database feature. An ASE remote source also supports failover.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.
3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[ASE1]
Server=<ase_machine_name>
Port=<ase_port>
Driver= libsybdrvodb-sqllen8.so
```

```
Database=<ase_database_name>
```

4. To enable failover in the DSN entry, add the `HASession` and `AlternativeServers` property. For example:

```
[ASE1]
Server=<ase_machine_name>
Port=<ase_port>
Driver= libsybdrvodb-sqllen8.so
Database=<ase_database_name>
HASession=1
AlternateServers=<failover_machine_name>:<failover_port_number>
```

5. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

     Sample Code

     ```
     CREATE REMOTE SOURCE My_ASE1 ADAPTER "aseodbc"
         CONFIGURATION 'DSN=ASE1;'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties.

     Sample Code

     ```
     CREATE REMOTE SOURCE My_ASE2 ADAPTER "aseodbc"
         CONFIGURATION
     'server=<ase_machine_name>;port=<ase_port>;Driver=libsybdrvodb-
     sqllen8.so;Database=<ase_database_name>;'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

   - To enable failover, add the HASession=1 and AlternateServers properties to the command.

Sample Code

```
CREATE REMOTE SOURCE My_ASE2 ADAPTER "aseodbc"
    CONFIGURATION
'server=<ase_machine_name>;port=<ase_port>;Driver=libsybdrvodb-
sqllen8.so;Database=<ase_database_name>;

HASession=1;AlternateServers=<failover_machine_name>:<failover_port_numb
er>;'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

- Use ALTER REMOTE SOURCE to enable failover on an existing ASE remote source.

Sample Code

```
CREATE REMOTE SOURCE My_ASE1 ADAPTER "aseodbc"
    CONFIGURATION 'DSN=ASE1;'
HASession=1;AlternateServers=<failover_machine_name>:<failover_port_numb
er>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *ASE (ODBC)*.
5. Select the connection mode.
6. Depending on the connection mode, enter the required information:

Adapter Properties

Property	Description
Server	Specifies the server of the ASE server.
Port	Specifies the port number of the ASE server.
Database Name	Specifies the name of the ASE server.

Adapter Properties

Property	Description
DML Mode	Specifies if the remote source is readwrite (default) or readonly.
Extra Adapter Properties	(For failover only) Enables automatic failover for the remote source. Enter: `HASession=1;AlternateServers=<failover_server>:<failover_port_number>`

Data Source Name

Property	Description
To enable failover, add the properties to the DSN section of the `.odbc.ini` file before you create the remote source.	
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

7. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
8. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
SAP Adaptive Server Enterprise ODBC Driver [page 1524]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.9 Create an SAP Event Stream Processor Remote Source

Create a remote source to an Event Stream Processor (ESP) window.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An SAP Event Stream Processor (ESP) remote source can be used with virtual tables or the linked database feature.

SAP HANA smart data access does not currently support the BOOLEAN and BINARY data types that exist in ESP. Therefore, any virtual tables created over ESP windows containing these column types either fail or produce incorrect data.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is used.

3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[ESP1]
Driver=libstreamingpsqlodbca_lib.so
Database=<esp_workspace_name>/<esp_project_name>
ServerName=<esp_machine_name>
Port=<esp_port_number>
SSLMode=disable
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

     Sample Code

     ```
     CREATE REMOTE SOURCE MY_ESP1 ADAPTER "odbc"
         CONFIGURATION FILE 'property_esp.ini'  CONFIGURATION 'DSN=ESP1'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties.

     Sample Code

     ```
     CREATE REMOTE SOURCE MY_ESP2 ADAPTER "odbc"
         CONFIGURATION FILE 'property_esp.ini'
         CONFIGURATION
     'ServerName=<esp_machine_name>;port=<esp_port_number>;Driver=libstreamin
     glibpsqlodbca_lib.so;
         Database='<esp_workspace_name>/<esp_project_name>;SSLmode=disable'
         WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>;
     ```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *ODBC (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Configuration File	Specify `property_esp.ini`
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]

SAP Event Stream Processor ODBC Driver [page 1525]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.10 Create an SAP HANA Streaming Analytics Remote Source

Create a remote source to a streaming analytics window.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An SAP HANA streaming analytics remote source can be used with virtual tables or the linked database feature.

SAP HANA smart data access does not currently support the BOOLEAN and BINARY data types that exist in streaming analytics. Therefore, any virtual tables created over streaming analytics windows containing these column types would either fail or produce incorrect data.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.

2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is used.

3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[STREAMING1]
Driver=libstreamingpsqlodbca_lib.so
Database=<streaming_workspace_name>/<streaming_project_name>
ServerNname=<streaming_machine_name>
Port=3<streaming_instance_number>16
SSLMode=enable
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- If you're planing to create the remote source using a DSN entry, it already exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

     Sample Code

```
CREATE REMOTE SOURCE MY_STREAMING1 ADAPTER "odbc"
    CONFIGURATION FILE 'property_esp.ini'  CONFIGURATION
'DSN=STREAMING1'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties.

     Sample Code

```
CREATE REMOTE SOURCE MY_STREAMING2 ADAPTER "odbc"
    CONFIGURATION FILE 'property_esp.ini'
    CONFIGURATION
'ServerName=<streaming_machine_name>;port=3<streaming_instance_number>16
;
    Driver=libstreaminglibpsqlodbca_lib.so'
    Database='<streaming_workspace_name>/
<streaming_project_name>;SSLmode=enable'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>;
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *ODBC (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Configuration File	Specify `property_esp.ini`
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]

SAP HANA Streaming Analytics ODBC Driver [page 1527]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.11 Create an SAP MaxDB Remote Source

Create a remote source to an SAP MaxDB database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An SAP MaxDB remote source can be used with virtual tables or the linked database feature. Enter all passwords in uppercase only, even if it contains lower or mixed case characters. The connection fails when lower or mixed case values are supplied.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.

   The `.odbc.ini` must exist, even if empty, regardless of whether a DSN entry is being used.

3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[MaxDB1]
Driver=/opt/MaxDB/lib/libsdbodbcw.so
ServerNode=<maxdb_machine_name>
ServerDB=MAXDB
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Context

> **Note**
>
> These steps are only required if you plan to use a DSN entry to create your remote source.

Procedure

1. In a SQL console, connect to the tenant database.
2. Do one of:
   - If using a DSN entry, execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

     Sample Code

     ```
     CREATE REMOTE SOURCE MY_MAXDB1 ADAPTER "maxdb"
         CONFIGURATION 'DSN=MaxDB1'
        WITH CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

   - If not using a DSN entry, execute a CREATE REMOTE SOURCE command specifying all driver properties.

     Sample Code

     ```
     CREATE REMOTE SOURCE MY_MaxDB2 ADAPTER "maxdb"
         CONFIGURATION
     'Driver=libsdbodbcw.so;ServerNode=<maxdb_machine_name>;ServerDB=MAXDB'
         with CREDENTIAL TYPE 'PASSWORD' USING
     'user=<user_name>;password=<password>';
     ```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *MaxDB (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
SAP MaxDB ODBC Driver [page 1528]

15.1.4.1.12 Create a Teradata Remote Source

Create a remote source to a Teradata database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

A Teradata remote source can be used with virtual tables or the linked database feature.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.
3. Add the following entries to the `.odbc.ini` file. For example:

```
[ODBC]
InstallDir=/<installation_path>/client/ODBC_64
[ODBC Data Sources]
default=tdata.so
TD=tdata.so
```

4. Also in the `.odbc.ini` file, define one entry for each remote source. For example:

```
[TDATA1]
Driver=/<installation_path>/client/ODBC_64/lib/tdata.so
DBCName=<server.com>
CharacterSet=UTF8
```

5. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

Sample Code

```
CREATE REMOTE SOURCE MY_TDATA1 ADAPTER "tdodbc"
    CONFIGURATION 'DSN=TDATA1'
    with CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *TERADATA (ODBC)*.
5. Select a connection mode.
6. Enter the required connection information:

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

7. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
8. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
Teradata ODBC Driver [page 1530]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.13 Create an SQL Server Remote Source

Create a remote source to an SQL Server database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An SQL Server remote source can be used with virtual tables or the linked database feature.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.
3. Define one entry in the `.odbc.ini` file for each remote source. For example:

```
[MSSQL1]
Driver= /opt/microsoft/msodbcsql/lib64/libmsodbcsql-18.0.so.1.1
Server=<sql_server_name>,<sql_port>
Database=<sql_database_name>
```

> **Note**
>
> From Microsoft ODBC Driver 18, the `encrypt` property is enabled by default. If the remote SQL Server is using self-signed certificates, the following parameter needs to be added to the DSN entry: `TrustServerCertificate=Yes`

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

Sample Code

```
CREATE REMOTE SOURCE MY_MSSQL1 ADAPTER "mssql"
    CONFIGURATION 'DSN=MSSQL1'
   with CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *MSSQL (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
SQL Server ODBC Driver [page 1531]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.14 Create an IBM DB2 Remote Source

Create a remote source to a DB2 database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An IBM DB2 remote source can be used with virtual tables or the linked database feature.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist.
3. Define a DSN entry in `.odbc.ini` file for each remote source. For example:

```
[DB2_1]
Driver=/opt/ibm/db2/odbc_cli/clidriver/lib/libdb2o.so.1
Description=TEST_DB2
```

4. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

Sample Code

```
CREATE REMOTE SOURCE MY_DB2_11 ADAPTER "db2"
    CONFIGURATION 'DSN=DB2_1'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *DB2 (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.

Data Source Name

Property	Description
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.
7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
IBM DB2 Driver [page 1531]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.15 Create an Oracle Remote Source

Create a remote source to an Oracle database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

An Oracle remote source can be used with virtual tables or the linked database feature.

Oracle remote sources support failover. Refer to your Oracle product documentation for information on configuring failover.

Oracle remote sources do not support empty strings. Values inserted into a virtual table that are generated from an Oracle remote source are transformed into NULL values if they are empty strings. This behavior also impacts some of the smart data access optimization techniques (for example, join relocation).

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. In the <sid>adm $HOME folder, create a `tnsnames.ora` file if it doesn't already exist. Define one entry for each remote source. For example:

   ```
   ORCL=
   (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP) (HOST = <oracle_hostname>) (PORT = 1521) )
   (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME =
   <oracle_machine_name>) ) )
   ```

3. Also in the <sid>adm $HOME folder, create a `.odbc.ini` file if it doesn't already exist and define one entry for each remote source. For example:

   ```
   [ORA12C_1]
   Driver=<install_oracle_driver_folder>/instantclient_12_1/libsqora.so.12.1
   ServerName=ORCL
   ```

4. To enable failover, add the `HASession` and `AlternateServers` properties to the command. For example:

   ```
   [ORA12C_1]
   Driver=<install_oracle_driver_folder>/instantclient_12_1/libsqora.so.12.1
   ServerName=ORCL
   HASession=1;AlternateServers=<failover_machine_name>:<failover_port_number>'
   ```

5. Restart the SAP HANA system to apply the changes to the files.

Create a Remote Source Using SQL Syntax

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Execute the CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file.

Sample Code

```
CREATE REMOTE SOURCE MY_ORA12C_1 ADAPTER "oracle"
    CONFIGURATION 'DSN=ORA12C_1'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *ORACLE (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
To enable failover, add the properties to the DSN section of the `.odbc.ini` file before you create the remote source.	
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.

7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
Oracle Database ODBC Driver [page 1532]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.16 Create an IBM Netezza Remote Source

Create a remote source to an IBM Netezza database.

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Context

An IBM Netezza remote source can be used with virtual tables or the linked database feature.

The following syntax examples assume the remote source is configured to use technical user credentials. To use secondary credentials, see *Managing Credentials*.

Create a DSN Entry

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (`<sid>`adm), change to the `<sid>`adm $HOME directory.

2. Add the following entry to the `odbcinst.ini` file:

```
[ODBC Drivers]
NetezzaSQL = Installed
[NetezzaSQL]
Driver           = /usr/local/nz/lib64/libnzodbc.so
Setup            = /usr/local/nz/lib64/libnzodbc.so
APILevel         = 1
ConnectFunctions = YYN
Description      = Netezza ODBC driver
DriverODBCVer    = 03.51
DebugLogging     = false
LogPath          = /tmp
UnicodeTranslationOption = utf16
CharacterTranslationOption = all
PreFetch         = 25600
Socket           = 16384
```

> ### Note
>
> Set the PreFetch property to a large value since IBM Netezza does not support multicursor on same connection.

3. Create a `.odbc.ini` file if it doesn't already exist.
4. Define a DSN entry in the `.odbc.ini` file for each remote source. For example:

```
[NTZ1]
Driver = /usr/local/nz/lib64/libnzodbc.so
Servername =  <ibm_machine_name>
Port = 5480
database = <ibm_database_name>
Username = <username>
Password = <password>
UnicodeTranslationOption = utf16
CharacterTranslationOption = all
```

> ### Note
>
> The UnicodeTranslationOption entry is required for HANA to connect successfully to an IBM Netezza remote source.

5. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.
2. Execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file. For example:

Sample Code

```
CREATE REMOTE SOURCE MY_NTZ1 ADAPTER "netezza"
    CONFIGURATION  'DSN=NTZ1'
    WITH CREDENTIAL TYPE 'PASSWORD' USING 'user=<username>;password=
<password>';
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *NETEZZA (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly.

6. Specify the credentials mode for the remote source.
   - Technical user – Specify a valid user and password to connect to the remote source. All connections to the remote source share the same credential for the remote source.
   - Secondary credentials – Create one credential per user per remote source. At least one secondary credential should exist before you create the remote source. If no secondary credentials exist, the credential mode is set to None, but once a secondary credential is created, the credential mode automatically switches to secondary credentials.

7. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)
Managing Credentials [page 1536]
IBM Netezza Driver [page 1533]
Security Aspects of SAP HANA Smart Data Access

15.1.4.1.17 Create a Google BigQuery Remote Source

Create a remote source to a Google BigQuery database.

Prerequisites

- You have the CREATE REMOTE SOURCE system privilege.
- The remote source is reachable by the network from the computer you are using.

Context

A Google BigQuery remote source can be used with virtual tables or the linked database feature.

A Google BigQuery remote source does not support authentication using a technical user or secondary credentials. Authentication is done using either a certificate key file or OAuth, which are defined in the DSN entry in the `.odbc.ini` file.

> **Note**
>
> For certificate key files, from Simba ODBC Driver 3.0.0, only JSON key files are supported.

Procedure

Enable safe mode for Google BigQuery.

a. If the instance does not yet have a scriptserver, connect to SYSTEMDB and in a console window, execute:

```
ALTER DATABASE <tenant_db_name> ADD 'scriptserver'.
```

b. Connect to the tenant database. Change the value of `odbc_adapters_in_scriptserver` parameter in `indexserver.ini`.

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM')
    SET ('smart_data_access', 'odbc_adapters_in_scriptserver') = 'bigquery'
WITH RECONFIGURE;
```

Create a DSN Entry

Procedure

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm), change to the <sid>adm $HOME directory.
2. Create a `.odbc.ini` file if it doesn't already exist and define one entry for each remote source. Use this template as a guideline.

```
[GoogleBQ]
# Description: DSN Description.
# This key is not necessary and is only to give a description of the data
source.
Description=Simba ODBC Driver for Google BigQuery (64-bit) DSN

# Driver: The location where the ODBC driver is installed to.
Driver=/opt/simba/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so

# These values can be set here, or on the connection string.
# Catalog: The catalog to connect to. This is a required setting.
Catalog=

# SQLDialect: The SQL Dialect to use.  There are two SQL dialects:
# 0 = BigQuery Legacy SQL
# 1 = BigQuery Standard SQL (SQL 11)
SQLDialect=1

# OAuth Mechanism: The OAuth mechanism to use.  There are two choices:
# 0 = Service Authentication
# 1 = User Authentication
#
# This is a required setting.
OAuthMechanism=0

# RefreshToken: The Refresh Token used. This can be generated from the
Windows connection dialog.
# It can also be generated by executing the following steps:
# 1. Get an Authentication by logging into Google from the following URL:
# https://accounts.google.com/o/oauth2/auth?scope=https://www.googleapis.com/
auth/bigquery&response_type=
```

```
#
code&redirect_uri=urn:ietf:wg:oauth:2.0:oob&client_id=977385342095.apps.google
usercontent.com&hl=en
# &from_login=1&as=76356ac9e8ce640b&pli=1&authuser=0
# 2. Run the get_refresh_token.sh shell script and pass in the Authentication
Token received in step 1.
# 3. Copy the Refresh Token (the text on the right-side of the colon, without
the trailing or leading spaces)
# from the output of the script. This is a required setting.
#RefreshToken=

# Email: For Service Authentication, this is a required setting. It is your
GENERATED service account email
# (not a typical Gmail account).
# It is unique and associated with at least one public/private key pair.
Email=

# KeyFile Path: For Service Authentication, this is a required setting.  This
is the path to the stored keyfile (.p12 or .json).
KeyFilePath=

# Used to specify the full path of the PEM formatted file containing trusted
SSL CA certificates.
# If an empty string is passed in for the configuration, the driver expects
the trusted SSL CA
# certificates can be found in the file named cacerts.pem located in the same
directory as the
# driver's shared library.
#TrustedCerts=

# AllowLargeResults: When set to 1, the driver allows for result sets in
responses to be larger than 128 MB.
AllowLargeResults=0

# LargeResultsDataSetId: DatasetId to store temporary tables created.  This
is a required setting if
# AllowLargeResults is set to 1.
LargeResultsDataSetId=_bqodbc_temp_tables

# LargeResultsTempTableExpirationTime: Time in milliseconds before the
temporary tables created expire.
# This is a required setting if AllowLargeResults is set to 1.
LargeResultsTempTableExpirationTime=3600000
```

3. Restart the SAP HANA system to apply the changes to the `.INI` file.

Create a Remote Source Using SQL Syntax

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. In a SQL console, connect to the tenant database.

2. Execute a CREATE REMOTE SOURCE command, referencing the DSN entry in the `.odbc.ini` file. For example:

Sample Code

```
CREATE REMOTE SOURCE BigQ ADAPTER "bigquery"
    CONFIGURATION 'DSN=GoogleBQ'
```

Create a Remote Source Using the SAP HANA Database Explorer

Prerequisites

- A DSN entry exists in the `.odbc.ini` file.

Procedure

1. Expand the *Catalog*.
2. In the context menu of the *Remote Sources* object, choose *Add Remote Source*.
3. Enter a remote source name.
4. In the *Adapter Name* dropdown list, select *BIGQUERY (GENERIC ODBC)*.
5. Enter the required connection information:

Data Source Name

Property	Description
Data Source Name	Specifies the DSN as defined in the `.odbc.ini` file.
DML Mode	Specifies if the remote source is readwrite (default) or readonly. Only readonly is supported for a Google BigQuery remote source.

6. Choose *Create*.

Related Information

CREATE REMOTE SOURCE Statement (Access Control)

Google BigQuery ODBC Driver [page 1534]
Security Aspects of SAP HANA Smart Data Access
SAP Note 3509345
SAP Note 3507638

15.1.4.2 Verify a Remote Source

You can use the CHECK_REMOTE_SOURCE procedure to verify that a remote source is correctly configured.

Context

When a remote source is created using the CREATE REMOTE SOURCE statement or modified using the ALTER REMOTE SOURCE statement, no errors are thrown if the configuration is not valid. The remote source is not validated until an attempt is made to establish a remote connection to it.

Procedure

To verify that a remote source is correctly configured, execute the following command:

```
CALL CHECK_REMOTE_SOURCE('<remote_source_name>');
```

If the check is not successful, an error message gives the reasons for the failure, such as an invalid host name.

15.1.4.3 Connect to a Remote Source As a Restricted User

Configure the restricted user in an SAP HANA remote source.

Prerequisites

Use of the restricted user is supported for SAP HANA remote sources only. The SAP HANA remote source must be running one of the following versions:

- SAP HANA 1.0 SPS 12 revision 122.20 or later
- SAP HANA 2.0 SPS 02 revision 24.05 or later
- SAP HANA 2.0 SPS 03 revision 033 or later

This functionality is not supported for remote sources running SAP HANA 2.0 SPS 00 or 01 (any revision). These limitations do not apply to the SAP HANA local source.

Context

Restricted users are intended for provisioning users who access SAP HANA through client applications and who are not intended to have full SQL access via an SQL console. A restricted user can be granted in a remote source for SDA access (JDBC/ODBC) only and provides more access control compared to standard database users.

Procedure

1. On the SAP HANA remote source, create the restricted user:

```
CREATE RESTRICTED USER <user_name> PASSWORD <password> NO
FORCE_FIRST_PASSWORD_CHANGE;
```

2. Grant ODBC and JDBC access to the restricted user:

```
GRANT RESTRICTED_USER_ODBC_ACCESS, RESTRICTED_USER_JDBC_ACCESS to <user_name>;
ALTER USER <user_name> ENABLE CLIENT CONNECT;
```

3. Grant the SDA role RESTRICTED_USER_SDA_ENABLE to the restricted user on the remote source:

```
GRANT RESTRICTED_USER_SDA_ENABLE TO <user_name>;
```

4. On the local SAP HANA, create the remote source. Specify the restricted user credentials for the technical user:

Related Information

User Types
Create a Restricted Database User
CREATE USER Statement (Access Control)

15.1.4.4 Modify a Remote Source

Modify an existing remote source.

Prerequisites

One of the following:

- You created the remote source.
- You have the CREATE REMOTE SOURCE system privilege.

Modify a Remote Source Using SQL Syntax

Procedure

In an SQL console, connected to the tenant database, execute:

```
ALTER REMOTE SOURCE <remote_source_name> <adapter_clause>
   [ <credential_clause> ];
```

Example

This example creates remote source, REMOTE1, and then changes the port for the remote source from `30115` to `30315`.

Sample Code

```
CREATE REMOTE SOURCE REMOTE1 ADAPTER "hanaodbc"
    CONFIGURATION 'Driver=libodbcHDB.so;ServerNode=my_machine:30115'
    WITH CREDENTIAL TYPE 'PASSWORD' USING
'user=<user_name>;password=<password>';
```

```
ALTER REMOTE SOURCE REMOTE1 ADAPTER "hanaodbc"
    CONFIGURATION 'Driver=libodbcHDB.so;ServerNode=<machine_name>:30315';
```

Related Information

ALTER REMOTE SOURCE Statement (Access Control)

Modify a Remote Source Using the SAP HANA Database Explorer

Procedure

1. Expand the *Catalog*.
2. Select *Remote Sources*.
   A list of remote sources appears in the catalog browser item list.
3. Select your remote source to open the remote source editor.
4. Choose *Edit*.
5. Make the changes and choose *Save*.

15.1.4.5 Drop a Remote Source

Remove an existing remote source.

Prerequisites

One of the following:

- You created the remote source.
- You have the CREATE REMOTE SOURCE system privilege.

Drop a Remote Source Using SQL Syntax

Procedure

In an SQL console, while you are connected to the tenant database, execute:

```
DROP REMOTE SOURCE <remote_source_name> CASCADE;
```

Drop a Remote Source Using the SAP HANA Database Explorer

Procedure

1. Expand the *Catalog*.
2. Select *Remote Sources*.
   A list of remote sources appears in the catalog browser item list.
3. In the context menu of the remote source you want to drop, choose *Delete*.
4. Select *Drop the remote source and any other objects that depend on it* and choose *Delete*.

15.1.4.6 List Remote Sources

Provides a list of remote sources you have privilege to.

Procedure

In an SQL console, while you are connected to the tenant database, execute:

```
SELECT * FROM "SYS"."REMOTE_SOURCES";
```

15.1.4.7 Customizing the Behavior of a Remote Source

The supported behaviors of a remote source may not be the same as those of the local SAP HANA instance. Smart data access provides a set of customizable properties, capabilities, functions, and data types to help address these differences.

Customizations are on a per remote source basis. If you have multiple remote sources using the same adapter, each remote source can have its own customizations.

15.1.4.7.1 Configure Remote Source Properties

Customize the behavior of properties for your remote source.

Prerequisites

You need the CREATE REMOTE SOURCE system privilege.

Context

Properties on a remote source define how to do something on the remote database. For example, what syntax is used to obtain a list of tables or how to create temporary tables. Properties are set using string values.

Each `<property_name>` specified must be in the form PROP_`<property_name>`. All properties are disabled by default.

Use the ALTER REMOTE SOURCE statement to set and unset capabilities. For the full syntax for setting and unsetting remote source properties, see *ALTER REMOTE SOURCE Statement* in the *SAP HANA SQL and System Views Reference*.

Procedure

1. Connected to the tenant database, open an SQL console window.
2. To set a new property value, execute:

```
ALTER REMOTE SOURCE <remote_source_name> SET PROPERTY '<property_name>' =
'<value>'
    [, '<property_name>' = '<value>' [,...];
```

3. To revert a property value to its default, execute:

```
ALTER REMOTE SOURCE <remote_source_name> UNSET PROPERTY { '<property_name>'
[, '<property_name>' [,...] | ALL };
```

Example

This example sets the value of the property PROP_MAX_NCHAR_STRING_LENGTH to 5000 for remote source RS1.

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'PROP_MAX_NCHAR_STRING_LENGTH' = '5000';
```

This example resets the property PROP_MAX_NCHAR_STRING_LENGTH to its default value for remote source RS1.

```
ALTER REMOTE SOURCE RS1 UNSET PROPERTY 'PROP_MAX_NCHAR_STRING_LENGTH';
```

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
Join Relocation for Multiple Remote Sources [page 1619]
List Current Properties for a Remote Source [page 1618]

15.1.4.7.2 Configure Remote Source Capabilities

Customize the behavior of capabilities for your remote source.

Prerequisites

You need the CREATE REMOTE SOURCE system privilege.

Context

Capabilities on a remote source define what types of functionality are supported by a remote database. For example, which join types are supported or what built-in functions are supported. Capabilities are set as true or false values.

Each `<capability_property_name>` specified must be in the form CAP_`<property_name>`. All capabilities are disabled by default.

Use the ALTER REMOTE SOURCE statement to set and unset capabilities. For the full syntax for setting and unsetting remote source properties, see *ALTER REMOTE SOURCE Statement* in the *SAP HANA SQL and System Views Reference*.

Procedure

1. Connected to the tenant database, open an SQL console window.
2. To set a new capability value, execute:

```
ALTER REMOTE SOURCE <remote_source_name> SET PROPERTY
'<capability_property_name>' = '<value>'
    [, '<capability_property_name>' = '<value>' [,...];
```

3. To revert a capability value to its default, execute:

```
ALTER REMOTE SOURCE <remote_source_name> UNSET PROPERTY
{ '<capability_property_name>' [, '<capability_property_name>' [,...] | ALL };
```

Example

This example sets the capability CAP_LIMIT to false for remote source RS1.

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'CAP_LIMIT' = 'FALSE';
```

This example resets the capability CAP_LIMIT to its default value for remote source RS1.

```
ALTER REMOTE SOURCE RS1 UNSET PROPERTY 'CAP_LIMIT';
```

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
List Current Properties for a Remote Source [page 1618]

15.1.4.7.3 Configure Remote Source Function Pushdown

Customize the behavior of SAP HANA functions pushed down to the remote source.

Prerequisites

You need the CREATE REMOTE SOURCE system privilege.

Context

When you execute a query, if a built-in function does not exist on both SAP HANA and a remote database, then unexpected behavior may result when the function is pushed to the remote source. For these occurrences, you can create mappings of which remote function to use when the SAP HANA function is referenced.

Each `<function_property_name>` must be specified in the form FUNC_`<function_name>`, followed by an argument type list bounded by brackets (FUNC_abs[INT]) or an argument count in braces (FUNC_abs{1}).

Specifying false for the property value marks the function as not supported by the remote database and is never pushed down to the remote source.

Each parameter passed to the SAP HANA SQL function is mapped to a placeholder. The first parameter is mapped to $1, the second to $2, and so on up to a maximum of 99 parameters. $* is supported as a wildcard, which causes a comma-separated list of parameters to be inserted into the translation string.

Parameter variations are supported as follows:

Syntax	Description	Comments
FUNC_abs = ABS()	Map a function (no arguments)	
FUNC_to_date = TO_DATE($*)	Map a function which has variable number of arguments	When TO_DATE is used, all arguments are inserted as a comma-separated list at the $*.
FUNC_substr{#2} = SUBSTRING($1, $2)	Map a function where the number of arguments is enforced	When SUBSTR is used with two arguments, the SUBSTRING expression is used.
FUNC_locate{#2} = LOCATE($2, $1)	Map a function where the argument order is changed	When LOCATE is used with two arguments, the first argument is substituted for $1 and the second argument for $2.
FUNC_locate{#2} = POSITION($1, $2)	Map to different function name	When LOCATE is used with two arguments, it is remapped to use POSITION on the remote system.

Syntax	Description	Comments
FUNC_locate{#2} = LOCATE($2 IN $1)	Map a function where the argument order is changed, and additional syntax is used	When LOCATE is used with two arguments, the first argument is substituted for $1 and the second for $2. The remote system requires that the syntax use IN as a keyword.
FUNC_abs[INT] = ABS($1)	Map a function where the type of arguments is specified	This function can have only one parameter and it must be of type INT. The parameter is substituted for $1.
FUNC_cast = CAST($1 AS NUMBER)	Cast arguments	Using Oracle for an example, the remote system casts the parameter to an Oracle NUMBER data type prior to calling SIGN.
FUNC_cast{#3} = CAST ($1 AS $2($3)), FUNC_cast{#4} = CAST ($1 AS $2($3,$4))	Specify a different mapping depending on the number of arguments to the function	When CAST is present with three arguments, the first of these two is used. If it has four arguments, the second is used.
FUNC_abs = false	Stop a function from being pushed down	Remapping ABS causes the query to fail.
FUNC_locate{#2} = LOCATE($1, $2), FUNC_locate{#3} = false	Stop a function from being pushed down only for certain number of parameters	When locate is used with two arguments, the remap succeeds. When LOCATE is used with three arguments, the remap fails.

Use the ALTER REMOTE SOURCE statement to set and unset capabilities. For the full syntax for setting and unsetting remote source properties, see *ALTER REMOTE SOURCE Statement* in the *SAP HANA SQL and System Views Reference*.

Procedure

1. Connected to the tenant database, open an SQL console window.
2. To set a new function push down value, execute:

```
ALTER REMOTE SOURCE <remote_source_name> SET PROPERTY
'<function_property_name>' = '<value>'
    [, '<function_property_name>' = '<value>' [,...];
```

3. To revert a function push down value to its default, execute:

```
ALTER REMOTE SOURCE <remote_source_name> UNSET PROPERTY
{ '<function_property_name>' [, '<function_property_name>' [,...] | ALL };
```

Example

This example sets the SAP HANA function ADD_DAYS to use the function DATEADD(DAY) on remote source RS1.

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'FUNC_add_days' = 'DATEADD(DAY)';
```

This example resets the function FUN_add_days to its default value on remote source RS1.

```
ALTER REMOTE SOURCE RS1 UNSET PROPERTY 'FUNC_add_days';
```

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
List Current Properties for a Remote Source [page 1618]

15.1.4.7.4 Configure Remote Source Data Type Mappings

Customize the mapping of SAP HANA data types for your remote source.

Prerequisites

You need the CREATE REMOTE SOURCE system privilege.

Context

If a data type does not exist on both SAP HANA and a remote database, then unexpected behavior may result when used on the remote source. For these occurrences, you can create mappings of which remote data type to use when the SAP HANA data type is referenced. For example, you map the SAP HANA data type FLOAT to DOUBLE on the remote source.

Each `<data_type_property_name>` specified must be in the form MAPPING_SQL_`<property_name>`.

Smart data access (SDA) supports a subset of the data types supported by SAP HANA core. Core data types outside the subset are automatically converted to one of the subset values when accessing a remote database.

TINYINT	DATETIME	VARBINARY
SMALLINT	FLOAT	LONGBINARY
INT	REAL	NCHAR

BIGINT	CHAR	NVARCHAR
DECIMAL	VARCHAR	LONGNCHAR
DATE	LONGCHAR	ST_GEOMETRY
TIME	BINARY	BOOLEAN

There are two kinds of data type mappings that can be customized: ODBC to SDA type mappings, and SDA to remote data base type mappings.

ODBC to SDA type mappings convert between ODBC SQL type and an SDA type. It is used when creating a virtual table to determine which HANA data type to use for each column. The standard ODBC SQL types have default mappings that should suffice but can be customized if needed. When customizing an ODBC SQL type, the property name begins with MAPPING_`<odbc_type_name>` and the value must be specified as TYPE_`<sda_data_type>`. Both `<odbc_type_name>` and `<sda_data_type>` are specified as string values.

SDA to remote data base type mappings define the syntax SDA uses when creating a remote table on the remote database, such as when creating a temporary remote table for join relocation. These values are predefined for SAP HANA supported remote database types, but can be customized as needed. For user defined adapters, a set of default data type mappings is obtained from the driver, however not all types may be customized. When customizing an SDA to remote data base type, the property name begins with TYPE_`<sda_data_type>`. Both `<sda_data_type>` and the property value are expressed as a string.

Use the ALTER REMOTE SOURCE statement to set and unset data type mappings. For the full syntax for setting and unsetting remote source properties, see *ALTER REMOTE SOURCE Statement* in the *SAP HANA SQL and System Views Reference*.

Procedure

1. Connected to the tenant database, open an SQL console window.
2. To set a new data type mapping value, execute:

```
ALTER REMOTE SOURCE <remote_source_name> SET PROPERTY
'<data_type_property_name>' = '<value>'
    [, '<data_type_property_name>' = '<value>' [,...];
```

3. To revert a data type mapping value to its default, execute:

```
ALTER REMOTE SOURCE <remote_source_name> UNSET PROPERTY
{ '<data_type_property_name>' [, '<data_type_property_name>' [,...] | ALL };
```

Example

This example maps the SDA data type VARCHAR to NVARCHAR on remote source RS1.

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'MAPPING_SQL_VARCHAR' = 'TYPE_NVARCHAR';
```

This example resets the SAP HANA data type property MAPPING_SQL_VARCHAR' to its default value on remote source RS1.

```
ALTER REMOTE SOURCE RS1 UNSET PROPERTY 'MAPPING_SQL_VARCHAR';
```

This example maps the SDA data type BOOLEAN to bit on remote source RS1.

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'TYPE_BOOLEAN' = 'bit';
```

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
List Current Properties for a Remote Source [page 1618]

15.1.4.7.5 List Current Properties for a Remote Source

List the customizable properties, capabilities, functions, and data types for a specific remote source.

Prerequisites

You need the EXECUTE privilege on the GET_REMOTE_SOURCE_PROPERTIES procedure.

Context

For a list of all customizable properties, capabilities, functions, and data types, call the procedure. In the results list, properties begin with PROP_, capabilities with CAP_, functions with FUNC_, and data types with MAPPING.

Procedure

In an SQL console, connected to the tenant database, execute the procedure:

Example

```
CALL GET_REMOTE_SOURCE_PROPERTIES ('<remote_source_name>', ?)
```

15.1.4.7.6 Migrate Adapter Property Files

Manually migrate customizations in the property_<adapter_type>.ini file.

Prerequisites

- You need the CREATE REMOTE SOURCE system privilege.

Context

With SAP HANA 2.0 SP S04 and later, settings for properties, capabilities, functions and data types are stored in the database. The property_<adapter_name>.ini file is no longer used.

If you updated your SAP HANA 2.0 system from a pre-SPS04 version and you want to continue to use customized property values, you must re-enter the customizations using the ALTER REMOTE SOURCE...SET PROPERTY statement.

If you are unsure which values were customized, compare the pre-update values in the properties_<adapter_name>.ini file with the post-update default values for the remote source.

To review the pre--update property_<adapter_name>.ini files:

1. Log on to the SAP HANA host as the SAP HANA software owner (<sid>adm).
2. Change to `/usr/sap/<sid>/SYS/exe/hdb/config` folder.
3. Open the property_<adapter_name>.ini file.

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
List Current Properties for a Remote Source [page 1618]
Configure Remote Source Capabilities [page 1612]
Configure Remote Source Function Pushdown [page 1614]
Configure Remote Source Data Type Mappings [page 1616]
Configure Remote Source Properties [page 1611]

15.1.4.7.7 Join Relocation for Multiple Remote Sources

The remote source property PROP_JOIN_RELOCATION_BETWEEN_SOURCES is used when tables from different remote sources need to be joined. It allows a join to be relocated so that it occurs directly in one of the remote sources instead of in the local SAP HANA system.

When two remote tables from two remote sources (one table per remote source) are joined, both tables are pulled to the local SAP HANA system, where the join is then performed. If one of these remote tables is small

and the expected join result is also small, it can be beneficial to send the smaller table to the other remote source and perform the join in the remote source with the bigger table. This approach can be effective when both of the following conditions apply:

- The smaller remote table is far smaller than the other bigger table.
- The join result is far smaller than the bigger remote table.

Because two remote sources cannot communicate directly with each other, the smaller table is first pulled to the local SAP HANA system and then sent to the other remote source.

Prerequisites

Join relocation is only available for remote sources based on SAP HANA smart data access built-in ODBC adapters. In addition, the remote sources must support the CAP_CRT_TEMP_TABLES capability, which allows temporary tables to be created. The following remote sources support temporary table creation:

- SAP HANA
- SAP IQ
- SAP Adaptive Server Enterprise
- Teradata
- Oracle
- Microsoft SQL Server
- IBM DB2

For more information about the supported versions of these remote sources, see SAP Note 2600176 - *SAP HANA Smart Data Access Supported Remote Sources*.

Join Relocation Behavior

Join relocation is enabled by default. Since it is a form of query optimization, it is chosen in a cost-based manner. This means that the SQL optimizer decides whether join relocation should be applied based on the estimated cost. Therefore, even if join relocation is enabled, it does not always result in a query plan with join relocation. By disabling join relocation, you can override the cost-based decision made by the SQL optimizer.

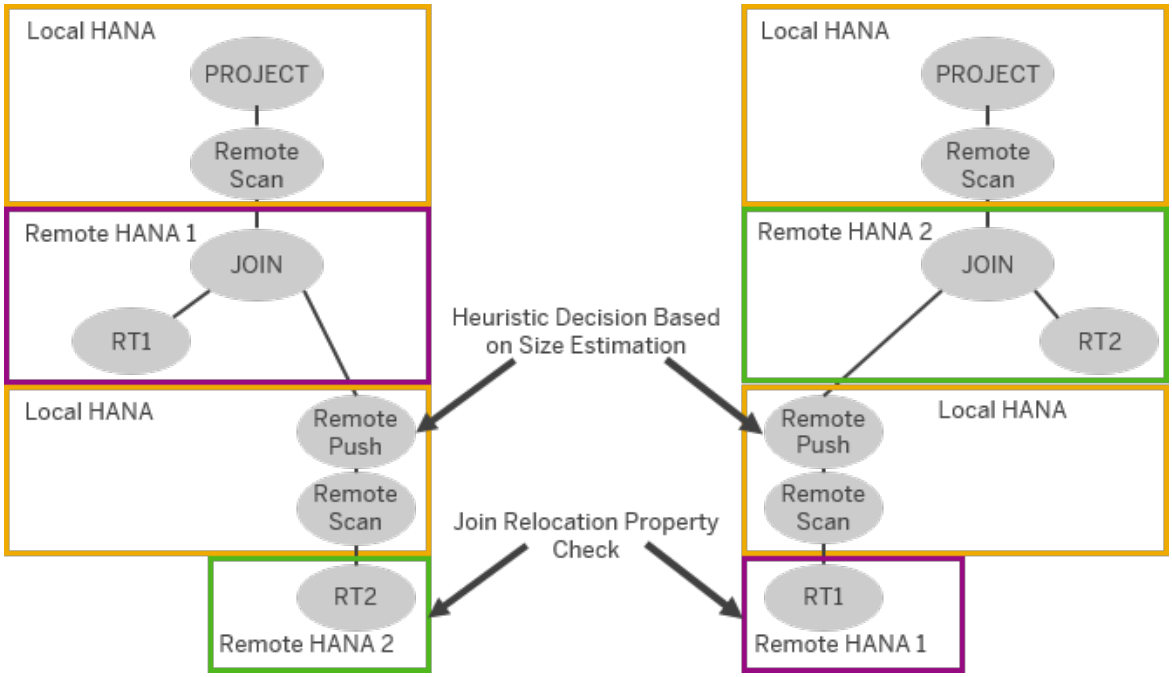
Note the following:

- The SQL optimizer chooses the direction of the remote push for the join relocation based on the estimated sizes of the remote tables. It does not change the direction of the remote push once it has been determined.
- The join relocation property is checked only for the remote source of the remote push target direction.

Example

The remote tables RT1 and RT2 in the remote sources remote HANA 1 and remote HANA 2 are to be joined. The remote table RT2 is much larger than the remote table RT1.

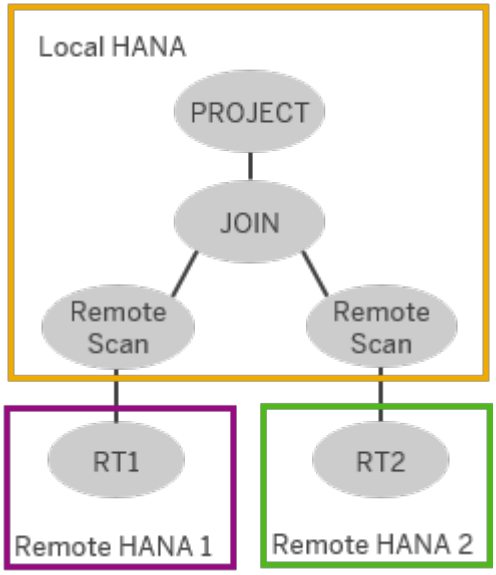
1. Join relocation is enabled (default)
   Based on the cost estimation, the SQL optimizer decides to apply join relocation.
   The remote push direction shown below on the right is chosen so that the join occurs in remote HANA 2:



2. Disable join relocation
   To override the SQL optimizer's decision, you now disable join relocation for remote HANA 1 (RS1):

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'PROP_JOIN_RELOCATION_BETWEEN_SOURCES' =
'false';
```

   When you run the query again, the join now occurs in the local SAP HANA system:



3. Enable join relocation

To be able to benefit from the optimized query plan again, you enable join relocation for remote HANA 1 (RS1):

```
ALTER REMOTE SOURCE RS1 SET PROPERTY 'PROP_JOIN_RELOCATION_BETWEEN_SOURCES' =
'true';
```

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
List Current Properties for a Remote Source [page 1618]
Supported Data Types for Remote Join Pushdown and Remote Join Relocation [page 1650]

15.1.4.8 Manage Read-Write and Read-Only Access to a Remote Source

The DML mode property specifies whether read-write or read-only access to the remote source is allowed. The DML mode, however, cannot override the restrictions of a remote source itself. For example, a Google BigQuery remote source is read-only and only the DML mode `readonly` is allowed.

Prerequisites

- Requires the ALTER object privilege on the remote source.

Context

You can set the DML mode property using the SQL console or the SAP HANA database explorer. When a remote source is created with the SQL console, the default setting of the DML mode is `readwrite`. Therefore, if the DML mode is not explicitly specified as either `readwrite` or `readonly`, the remote source is created in read-write mode by default. When using the SAP HANA database explorer, the default selection for the DML mode is also `readwrite`.

To enable write operations on the remote source using federation, the connection property DML_MODE must be set to READWRITE (or the default setting) in the remote source definition. Certain remote source types may have limited write capabilities. The available write operations are represented by the corresponding adapter capabilities, such as CAP_INSERT, CAP_UPDATE, CAP_DELETE, and so on.

INSERT, UPDATE, and DELETE operations can be executed on virtual tables only when the DML mode property is set to `readwrite` on a remote source.

Note the following points:

- INSERT, UPDATE, and DELETE operations are supported on virtual tables to provide more flexibility. However, the supported scope is very limited compared with DML operations executed on normal tables. More specifically, only DML statements with simple predicates and simple data types are supported and only on a single remote source. For example, LOB data types, correlated queries, cross-source operations such as join relocation, semi-join reduction, and so on are not supported. Cross-source DML operations (including remote and local sources) are not allowed.
- Distributed transaction control is not fully supported across the local SAP HANA database and the involved remote systems. Therefore, consistency cannot be guaranteed across the SAP HANA database and the remote systems in write scenarios.
- The read-only option improves performance for SELECT queries.
  The `readonly` setting does not affect features like remote join relocation, which internally requires temporary table creation. Creation and manipulation of temporary tables on a remote source are not considered data updates.

Enable Read-Write or Read-Only Access Using SQL Syntax

Procedure

1. To display the current value of the DML mode property, retrieve the remote connection details from the M_REMOTE_CONNECTIONS monitoring view, which provides information about the remote connections active in the database:
   a. To ensure that the status shown in the M_REMOTE_CONNECTIONS view is up to date, execute:

   ```
   CALL CHECK_REMOTE_SOURCE('<remote_source_name>');
   ```

   b. To retrieve the details about the remote connection, execute:

   ```
   SELECT * FROM SYS.M_REMOTE_CONNECTIONS WHERE
   CONNECTION_ID=CURRENT_CONNECTION;
   ```

   The DETAILS column shows whether a remote connection was created as `readonly` or `readwrite`. There might be multiple entries if the DML mode is `readonly`.
2. To modify the DML mode option, execute the following:
   - To enable read-write access:

     Sample Code

     ```
     ALTER REMOTE SOURCE <remote_source_name> ADAPTER "hanaodbc"
         CONFIGURATION
     'Driver=libodbcHDB.so;ServerNode=<remote_server_name>:3<remote_instance_
     number>15;
         dml_mode=readwrite';
     ```

   - To enable read-only access:

     Sample Code

     ```
     ALTER REMOTE SOURCE <remote_source_name> ADAPTER "hanaodbc"
     ```

```
    CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=<remote_server_name>:3<remote_instance_
number>15;
    dml_mode=readonly';
```

Enable Read-Write or Read-Only Access Using the SAP HANA Database Explorer

Procedure

1. Expand the *Catalog*.
2. Select *Remote Sources*.
   A list of remote sources appears in the catalog browser item list.
3. Select your remote source to open the remote source editor.
4. Choose *Edit*.
5. Modify the *DML Mode* field and choose *Save*.

Related Information

ALTER REMOTE SOURCE Statement (Access Control)
M_REMOTE_CONNECTIONS System View

15.1.5 Using Linked Database

Linked database allows DML queries on remote sources without the need to first create virtual tables for each table referenced in a query before executing the query.

This feature makes ad-hoc access to remote data much more convenient.

Linked database uses a three-part namespace to directly identify the remote database, schema, and table name. The three-part name syntax is defined as:

```
<remote_source>.<remote_schema>.<remote_table>
```

Queries between tenant databases, known as cross-database access, also use this three-part namespace. It is possible for the name of a remote source and a tenant database to be the same. If this situation occurs, then the database name resolves to the tenant database as a cross-database query, not the remote source.

The linked database feature supports all smart data access remote sources. The LINKED DATABASE object-level privilege on the remote source is required to use the feature.

When a query is executed, the system locates the referenced remote table and then checks to see if it has been used in previous queries. If it hasn't, then the system automatically creates an internal virtual table, which is used to process the query. Subsequent queries to the same remote table are automatically redirected to the internal virtual table. Users cannot directly access these internal virtual tables. However, they can be administered for refreshing metadata and housekeeping.

> ⓘ Note
>
> - Management of linked database can only be performed using SQL syntax.
> - Managing linked database is not applicable when optimized mode is enabled.

Example

The following statement accesses the remote table TABLE1 on the remote source HANA1:

```
SELECT * FROM HANA1.MYSCHEMA.TABLE1;
```

The following statement inserts a row into the remote table TABLE1 on the remote source HANA1:

```
INSERT INTO HANA1.MYSCHEMA.TABLE1 VALUES(09,'Mousepad',4);
```

The following statement deletes a row from the remote table TABLE1 on the remote source HANA1 where the PO column is equal to 09:

```
DELETE FROM HANA1.MYSCHEMA.TABLE1 WHERE PO = 09;
```

The internally generated virtual table can be seen as follows:

```
SELECT "SCHEMA_NAME", "TABLE_NAME", "REMOTE_SOURCE_NAME", "REMOTE_OBJECT_NAME"
FROM "SYS"."VIRTUAL_TABLES"
WHERE REMOTE_OBJECT_NAME LIKE '%TABLE1%';
```

For example:

SCHEMA_NAME	TABLE_NAME	REMOTE_SOURCE_NAME	REMOTE_OBJECT_NAME
_SYS_LDB	LT0000000000000002	HANA1	TABLE1

Linked Database Optimized Mode

With optimized mode, when a query is executed, the metadata for referenced remote objects is cached locally. Internal virtual tables are no longer created. This metadata is automatically refreshed. No housekeeping tasks are required. If the remote object referenced is a view, then the metadata of the tables, which are part of the view definition, are also cached locally. You don't need to create or refresh statistics for remote objects. They are fetched automatically when needed. The global optimizer may generate a better query plan, resulting in better query performance.

While linked database is supported for all smart data access remote sources, optimized mode is only available for SAP HANA to SAP HANA workflows.

Optimized mode is disabled by default. To enable it, when you create a remote source, set the `linkeddatabase_mode` property to *optimized*. For example:

```
CREATE REMOTE SOURCE MY_HANA1 ADAPTER "hanaodbc"
    CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=myserver:30115;linkeddatabase_mode=optimized'
    WITH CREDENTIAL TYPE 'PASSWORD' USING 'user=user1;password=Test1234';
```

Linked database optimized mode has some function restrictions. For more information, see SAP Note 2605574 - *Linked Database Optimized Mode Functional Restrictions*.

Related Information

SAP Note 2605574
Refresh Linked Database [page 1626]
Drop Linked Tables [page 1627]
Synonyms [page 1652]

15.1.5.1 Refresh Linked Database

Refresh metadata for a single linked table or all linked objects using the remote source.

Prerequisites

- If linked database optimized mode is disabled, you require the LINKED DATABASE object level privilege on the remote source, regardless of who created the remote source.
- If database optimized mode is enabled, no additional privilege is required.

Context

The REFRESH LINKED OBJECTS clause refreshes metadata for all linked objects. Definitions of any tables changed on the remote source are updated. Refreshing metadata could potentially take some time to execute. It is recommended that you schedule the execution within an appropriate maintenance window. The REFRESH LINKED TABLE clause refreshes metadata for a single specified table. Use this clause to perform an immediate refresh of a changed table when it would be difficult to execute the refresh on the entire linked database.

When refreshing a linked table, `<table_name>` is the name of the table on the remote source.

Procedure

To refresh metadata on all linked objects or for a single linked table, execute the following in the local SAP HANA database:

```
ALTER REMOTE SOURCE <remote_source> { REFRESH LINKED OBJECTS
    | REFRESH LINKED TABLE <remote_source>.<schema_name>.<table_name> };
```

Example

To refresh metadata of all linked objects associated with remote source `MYREMOTESYS`, execute the following in the local SAP HANA database:

Sample Code

```
ALTER REMOTE SOURCE MYREMOTESYS REFRESH LINKED OBJECTS;
```

To refresh metadata for remote table `MYREMOTESYS.MYTABLE` on remote source `MYREMOTESYS`, execute the following in the local SAP HANA database:

Sample Code

```
ALTER REMOTE SOURCE MYREMOTESYS REFRESH LINKED TABLE
MYREMOTESYS.MYSCHEMA.MYTABLE;
```

15.1.5.2 Drop Linked Tables

Drop all internally generated objects associated with linked database by using the remote source.

Prerequisites

- For linked database, you require the LINKED DATABASE object level privilege on the remote source, regardless of who created the remote source.

Context

The CASCADE option drops all the linked tables and dependent objects associated with linked tables. The RESTRICT option drops linked tables only if there are no dependencies on any of the linked tables. If this option is used and there are dependent objects on a linked table, then an error is raised, and all linked tables are

retained. If no drop option is specified, then all internally generated linked tables with no dependencies are dropped. Linked tables with dependencies are retained.

Procedure

To clear generated objects, execute the following in the local SAP HANA database:

```
ALTER REMOTE SOURCE <remote_source_name> DROP LINKED OBJECTS [CASCADE |
RESTRICT];
```

Example

To clear linked objects associated with the remote source `MYREMOTESYS` that have no current references to a linked object, execute the following in the local SAP HANA database:

Sample Code

```
ALTER REMOTE SOURCE MYREMOTESYS DROP LINKED OBJECTS;
```

To clear linked objects associated with the remote source `MYREMOTESYS` and drop any dependent objects that reference the linked object, execute the following in the local SAP HANA database:

Sample Code

```
ALTER REMOTE SOURCE MYREMOTESYS DROP LINKED OBJECTS CASCADE;
```

15.1.6 Managing Virtual Tables

Virtual tables point to remote tables or views in a remote source. When SQL queries are executed on a virtual table, they access the remote data as if it were stored locally.

You can create virtual tables that point to the following object types:

- Tables
- SQL views
- Calculation views
- SAP HANA CDS views (generated SQL views)
- Synonyms for the above object types

The following restrictions apply:

- You can't create virtual tables based on tables with a flexible schema.
- You can't create virtual tables based on SAP HANA CDS views with input parameters.

> **Note**
>
> - You can use the SAP HANA database explorer or SQL syntax to manage virtual tables.
> - Some management tasks are not available in all management tools.
> - Managing virtual tables is not applicable when using linked database.

15.1.6.1 Managing Virtual Tables Using SQL Syntax

Create, view, refresh, and delete virtual tables using SQL syntax.

15.1.6.1.1 Create a Virtual Table

Create a virtual table using SQL syntax.

Prerequisites

- One of the following applies:
  - You created the remote source.
  - You have the CREATE VIRTUAL TABLE object privilege on the remote source.
- To use the WITH REMOTE clause you also need the REMOTE TABLE ADMIN object privilege.

Context

When creating a virtual table, the remote table does not need to already exist. Including the `<table_contents_source>` and WITH REMOTE clauses in the CREATE VIRTUAL TABLE statement first creates the remote table on the remote source and then creates the associated virtual table. For the full syntax for this feature, see the CREATE VIRTUAL TABLE statement in the SAP HANA SQL Reference Guide.

When using the WITH REMOTE clause, if the credential type of the remote source is PASSWORD, then only the owner of the remote sources who set the technical user credentials has the authorization to use the WITH REMOTE clause. For all other credential types, authorization is a function of the privileges of the user on the remote source. This functionality is supported for the following remote sources:

- SAP HANA
- SAP IQ
- SAP Adaptive Service Enterprise (ASE)
- ORACLE
- Microsoft SQL Server

- IBM DB2
- Teradata
- IBM NETEZZA
- Google BigQuery

Procedure

1. Open an SQL console.
2. Execute the command specifying the schema and name of the virtual table and the name of the remote source:

```
CREATE VIRTUAL TABLE [<schema_name>.]<virtual_table_name> [ (
<table_contents_source> ) ]
AT
"<remote_source>"."<database_name>"."<remote_schema_name>"."<remote_table_name
>" [ WITH REMOTE ];
```

Example

The following statement creates virtual table HANA1_T1 from the remote source HANA1. The remote database name is DT1 and the schema of T1 is MYSCHEMA2:

```
CREATE VIRTUAL TABLE MYSCHEM1.HANA1_T1 AT "HANA1"."DT1"."MYSCHEMA2"."T1";
```

This example first creates the remote table T2 containing integer columns a2 and b2, and then creates the virtual table VIRTUAL_T2 from table T2.

```
CREATE VIRTUAL TABLE MYSCHEM1.VIRTUAL_T2 (a INT, b INT) AT
"HANA1"."DT1"."MYSCHEMA2"."T2" WITH REMOTE;
```

Related Information

CREATE VIRTUAL TABLE Statement (Data Definition)

15.1.6.1.2 List Virtual Tables by Schema

Display the virtual tables of a remote source by schema using SQL syntax.

Context

You cannot change the definition or contents while viewing.

Procedure

1. Open an SQL console.
2. Execute the SELECT statement specifying the schema name of the virtual tables to list.

   ```
   SELECT * FROM "SYS"."VIRTUAL_TABLES" WHERE SCHEMA_NAME='<schema_name>';
   ```

   Tables, which you have permissions on, appear for the specified schema. The list includes both local and virtual tables.

Example

The following statement lists the tables for schema `user1`:

```
SELECT * FROM SYS.VIRTUAL_TABLES WHERE SCHEMA_NAME='user1';
```

Related Information

SELECT Statement (Data Manipulation)

15.1.6.1.3 Refresh a Virtual Table

Update a virtual table to reflect metadata changes in the corresponding remote source table using SQL syntax.

Prerequisites

Requires the CREATE VIRTUAL TABLE object privilege.

Context

When changes to the metadata in a remote table are made, the changes are not automatically reflected in the corresponding virtual table. Manually update the virtual table to reflect the changes.

Procedure

To refresh the virtual table, execute:

```
ALTER VIRTUAL TABLE [<schema_name>.]"<virtual_table_name>" REFRESH DEFINITION;
```

Example

The following statement refreshes the content of virtual table `HANA1_T1` in schema `user1`:

```
ALTER VIRTUAL TABLE user1."HANA1_T1" REFRESH DEFINITION;
```

Related Information

ALTER VIRTUAL TABLE Statement (Data Definition)

15.1.6.1.4 Delete a Virtual Table

Remove an existing virtual table from the target system using SQL syntax.

Prerequisites

- One of the following applies:
  - You created the virtual table.
  - You have the DROP privilege on the virtual table.
- To use the WITH REMOTE clause you also need the REMOTE TABLE ADMIN object privilege.

Context

If access to a virtual table has been granted to multiple users, then deleting the table removes it for all users, not just the user deleting it.

When dropping a virtual table, if you include the WITH REMOTE clause, both the corresponding table on the remote source is also dropped. For the full syntax for this feature, see the DROP TABLE statement in the SAP HANA SQL Reference Guide.

Procedure

1. Open an SQL console.
2. Execute:

```
DROP TABLE [<schema_name>]."<virtual_table_name>" CASCADE [ WITH REMOTE ];
```

Example

The following statement deletes virtual table `HANA1_T1` in the schema `user1`.

```
DROP TABLE user1."HANA1_T1" CASCADE;
```

The following statement deletes virtual table `HANA1_T2` in the schema `user1` from the local source and the corresponding table on the remote source.

```
DROP TABLE user1."HANA1_T2" CASCADE WITH REMOTE;
```

Related Information

DROP TABLE Statement (Data Definition)

15.1.6.2 Managing Virtual Tables Using the SAP HANA Database Explorer

Create virtual tables, view table content and definition, and delete virtual tables using the SAP HANA database explorer.

15.1.6.2.1 Create a Virtual Table

Create a virtual table from the remote object of a remote source.

Prerequisites

One of the following:

- You created the remote source.
- You have the CREATE VIRTUAL TABLE object privilege on the object created by another user.

Procedure

1. In the SAP HANA database explorer, expand the *Catalog*.
2. Select *Remote Sources*.
   A list of remote sources appears in the catalog browser item list.
3. Select a remote source.
   The remote source editor opens.
4. Filter for the applicable schema and type of table and then choose *Search*.
   A list of tables on the remote source for the specified schema appears.
5. Select one or more objects and choose *Create Virtual Object(s)*.
   The *Create Virtual Object* dialog opens. If multiple objects are selected, you are prompted to provide a prefix that is then applied to each virtual table created. If only one object is selected, the virtual table is the same as the remote table name. SAP recommends you add a prefix to any virtual table name. This ensures local and virtual table names are unique and makes it easier to identify virtual tables.
6. Choose *Create*.
   A list of tables for the specified schema appears in the catalog browser. The list includes both local and virtual tables. Virtual tables are those with the green, remote sources icon ( ).

15.1.6.2.2 List Virtual Tables By Schema

Display the virtual tables of a remote source by schema using the SAP HANA database explorer.

Context

You cannot change the definition or contents while viewing a virtual table.

Procedure

1. In the SAP HANA database explorer, expand the *Catalog*.
2. Select *Tables*.

Results

Tables, which you have permissions to for the specified schema, appear in the catalog browser. The list includes both local and virtual tables. Virtual tables are those with the green, remote sources icon ( ). To change the schema, choose the (*Choose schema*) icon beside the schema name and select a new schema. If multiple schemas are selected, the schema name appears with each table listed.

15.1.6.2.3 Delete a Virtual Table

Delete an existing virtual table from your schema using the SAP HANA database explorer.

Prerequisites

One of the following:

- You created the virtual table.
- You have the DROP privilege on the object created by another user.
- To use the WITH REMOTE clause you also need the REMOTE TABLE ADMIN object privilege.

Context

If access to a virtual table has been granted to multiple users, deleting the table removes it for all users, not just the user deleting it.

Procedure

1. In the SAP HANA database explorer, expand the *Catalog*.
2. Select *Tables*.
3. In the context menu of the virtual table, choose *Delete*.
4. Specify how to handle any dependencies on the virtual table being dropped.

Results

The virtual table is deleted from the schema.

15.1.6.3 List All Virtual Tables

Provides a list of all virtual tables you have privilege to.

Procedure

Execute:

```
SELECT * FROM "SYS"."VIRTUAL_TABLES"
```

15.1.6.4 Check Virtual Table Definitions

The virtual table check is a procedure available in the SAP HANA database that checks for mismatches between the definition of a virtual table in the local system and the definition of the remote table it points to.

Prerequisites

- To run the check specifying both the schema and table, you must have either of the following:
  - The CREATE VIRTUAL TABLE object privilege on the remote source to which the virtual table belongs
  - The CATALOG READ system privilege

Context

It's natural that a virtual table's definition is slightly different from the target remote table's definition since columns might be converted or normalized to fit into more widely compatible column types.

The virtual table and its target remote table are regarded as consistent if the definition of the virtual table stored in the local system is identical to the table definition retrieved from the remote system at the time of the procedure call. Note that the mismatches found are not necessarily harmful.

Procedure

In an SQL console, execute the following:

```
CALL CHECK_VIRTUAL_TABLES('<action-name>', '<schema-name>' | NULL, '<virtual-
table-name>' | NULL )
```

Parameters:

- `action-name`: The only action name that is currently valid is **`CHECK`**.
- `schema-name`: The schema to check. Enter **`NULL`** to check all schemas.
- `virtual-table-name`: The table to check. Enter **`NULL`** to check all tables in the schema given by `schema-name`.

Results

The check results give, for example, the mismatches found in column definitions, primary key definitions, and table or column privileges.

The output columns are as follows:

- SCHEMA_NAME: The relevant schema name for the result row.
- TABLE_NAME: The relevant table name for the result row.
- COLUMN_NAME: The relevant column name for the result row, if any.
- RESULT_CODE: One of the following:

Result Code	Description
COLUMN_ADDED	Column added to remote table
COLUMN_REMOVED	Column removed from remote table
COLUMN_TYPE_CHANGED	Column type code changed
COLUMN_SIZE_CHANGED	Column size changed while type remained the same
COLUMN_SCALE_CHANGED	Column precision changed while type remained the same
COLUMN_NULLABILITY_CHANGED	Nullability (NULL or NOT NULL) changed
PK_ADDED	Primary key added to remote table
PK_REMOVED	Primary key removed from remote table
PK_COLUMN_ADDED	Column added to primary key
PK_COLUMN_REMOVED	Column removed from primary key
TABLE_DEFINITION_CHANGED	Definition value string of table changed
RETRIEVAL_FAILED	Retrieval of remote metadata failed

- RESULT_MESSAGE: Details about the inconsistency, for example, `column C1 changed nullability from NULL to NOT NULL`.
- SEVERITY: A mismatch can cause errors when SQL statements are executed on a virtual table with a mismatch. The SEVERITY indicates the probable outcome based on the RESULT_CODE. For example, a column added (COLUMN_ADDED) to a remote table does not cause any errors (an added column in the result can be ignored) whereas a column that has been removed (COLUMN_REMOVED) does cause an error (according to the local definition, a required column is missing).
  The severity levels are as follows:

Severity	Description
INFO	Mismatch does not lead to SQL errors
WARNING	Mismatch may lead to SQL errors
ERROR	Mismatch is likely to lead to SQL errors

Example

The following statement checks all tables in all accessible schemas:

```
CALL CHECK_VIRTUAL_TABLES('CHECK', NULL, NULL)
```

An example result is shown below:

```
call SYS.CHECK_VIRTUAL_TABLES('CHECK', null, null)
```

	SCHAME_NAME	TABLE_NAME	COLUMN_NAME	RESULT_CODE	RESULT_MESSAGE	SEVERITY
1	SYSTEM	VTAB2	C1	COLUMN_NULLABILITY_CHANGED	column C1 in remote table changed nullability from NULL to NOT NULL	INFO
2	SYSTEM	VTAB2		PK_ADDED	primary key (C1) added to remote table	INFO
3	SYSTEM	VTAB1	C2	COLUMN_REMOVED	column C2 removed from remote table	WARNING
4	SYSTEM	VTAB1	C3	COLUMN_ADDED	column C3 added to remote table	INFO
5	SYSTEM	VTAB1	C5	COLUMN_TYPE_CHANGED	column C5 in remote table changed type from INTEGER to DOUBLE	WARNING
6	SYSTEM	VTAB1	C6	COLUMN_TYPE_CHANGED	column C6 in remote table changed type from DECIMAL(size: 3, scal...	WARNING
7	SYSTEM	VTAB1	C7	COLUMN_SIZE_CHANGED	column C7 in remote table changed size from 20 to 50	WARNING
8	SYSTEM	VTAB1	C8	COLUMN_SCALE_CHANGED	column C8 in remote table changed scale from 2 to 3	WARNING
9	SYSTEM	VTAB1	C8	COLUMN_SIZE_CHANGED	column C8 in remote table changed size from 3 to 4	WARNING
10	SYSTEM	VTAB1	C2	PK_COLUMN_REMOVED	column C2 removed from primary key (C1,C3) of remote table	INFO
11	SYSTEM	VTAB1	C3	PK_COLUMN_ADDED	column C3 added to primary key (C1,C3) of remote table	INFO
12	SYSTEM	VTAB3	C1	COLUMN_NULLABILITY_CHANGED	column C1 in remote table changed nullability from NOT NULL to NULL	INFO
13	SYSTEM	VTAB3	C2	COLUMN_NULLABILITY_CHANGED	column C2 in remote table changed nullability from NOT NULL to NULL	INFO
14	SYSTEM	VTAB3		PK_REMOVED	primary key (C1,C2) removed from remote table	INFO

Next Steps

If you have found inconsistencies in the table definitions, you can resolve them by refreshing the corresponding virtual table.

Issues with severity level ERROR cannot generally be resolved by simply refreshing the virtual table definition. This applies as follows:

- Cases where the errors returned cannot be resolved by refreshing the virtual table definition:
  - Remote table does not exist
  - Could not reach the remote source (network error, wrong endpoint, and so on)
  - SAP HANA internal error such as out-of-memory
- Error that can be resolved by refreshing the virtual table definition:
  - Virtual table is pointing to a remote parameterized view or table user defined function, but the remote object does not have any parameters anymore
    When the virtual table definition is refreshed, the flag indicating whether the remote object is a parameterized SQL view is updated and the parameter metadata of the virtual table is truncated.

Note that for all four cases the result code is RETRIEVAL_FAILED.

Related Information

Refresh a Virtual Table [page 1631]

15.1.6.5 EXPORT/IMPORT Virtual Tables Between SAP HANA Systems

Export virtual tables from one system and import them into another.

Prerequisites

- A remote source must exist on the target system using the same name as the source system.
- You have the system privileges IMPORT, EXPORT, and INSERT.

Context

You can export some or all of the existing virtual tables in the source system to a new system, specifying a new schema or database name during the process.

Both the source and target systems must be running SAP HANA 2.0 SPS 01 or later, but do not have to be running the same version.

Ensure you save the export file to a location available to the target system.

EXPORT/IMPORT When the Schema Name Is the Same on the Target and Source Systems

Procedure

1. On the source system, export part or all of the schema to a temporary directory.

   ```
   EXPORT <schema_name>.[ <table_name> | "*" ] AS CSV INTO '<temporary_dir>';
   ```

2. Create the remote source on the target system using the same name as the source system.
3. On the target system, import the schema.

   ```
   IMPORT "<schema_name>"."[<table_name> | * ]" AS CSV FROM '<temporary_dir>';
   ```

Example

Export all tables in `MYSCHEMA` to remote source `RS`, using a temporary directory `/export_hold`.

```
EXPORT MYSCHEMA."*" AS CSV INTO '/export_hold';
```

```
CREATE REMOTE SOURCE RS ADAPTER "hanaodbc" CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=mymachine:30115'
    WITH CREDENTIAL TYPE 'PASSWORD' USING 'user=<user_name>;password=<password>';
IMPORT "MYSCHEMA"."*" AS CSV FROM '/export_hold';
```

Export only table VT in MYSCHEMA to remote source RS, using a temporary directory /export_hold.

```
EXPORT MYSCHEMA.VT AS CSV INTO '/export_hold';
CREATE REMOTE SOURCE RS ADAPTER "hanaodbc" CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=mymachine:30115'
    WITH CREDENTIAL TYPE 'PASSWORD' USING 'user=<user_name>;password=<password>';
IMPORT "MYSCHEMA"."VT" AS CSV FROM '/export_hold';
```

Rename the Remote Object During Transport

Procedure

1. On the source system, export part or all of the schema to a temporary directory.

   ```
   EXPORT <schema_name>.[ <table_name> | "*" ] AS CSV INTO '<temporary_dir>';
   ```

2. Create the remote source on the target system using the same name as the source system.
3. On the target system, import the schema, specifying the <original_database_name> and <original_schema_name> and the <new_database_name> and <new_schema_name>.

   ```
   IMPORT "<schema_name>"."[<table_name> | * ]" AS CSV FROM '<temporary_dir>'
       WITH RENAME REMOTE OBJECT
   <remote_source>.<original_database_name>.<original_source_schema>.TBL
       TO <remote_source>.<new_database_name>.<new_source_schema>.TBL;
   ```

Example

Export table VT in MYSCHEMA to remote source RS, with a database name ADMIN and a schema name RS_SCHEMA.

```
EXPORT MYSCHEMA."VT" AS CSV INTO '/export_hold';
CREATE REMOTE SOURCE RS ADAPTER "hanaodbc" CONFIGURATION
'Driver=libodbcHDB.so;ServerNode=mymachine:30115'
    WITH CREDENTIAL TYPE 'PASSWORD' USING 'user=<user_name>;password=<password>';
IMPORT "MYSCHEMA"."VT" AS CSV FROM '/tmp' WITH RENAME REMOTE OBJECT
RS.ADMIN.RS_SCHEMA.TBL TO RS.NEW_ADMIN.NEW_RS_SCHEMA.TBL;
```

15.1.7 Virtualizing Parameterized Views

In the SAP HANA on-premise system, you can create virtual parameterized views that point to remote parameterized SQL views in another SAP HANA on-premise system or in an SAP HANA Cloud, SAP HANA

database. You can also create virtual parameterized views that point to remote parameterized ABAP CDS view entities in a remote ABAP system.

A virtual parameterized view lets you pass parameters to the corresponding remote parameterized view (SQL view or ABAP CDS view entity) to retrieve data from the remote system. By passing parameter values when querying the view, you can avoid the WHERE clause that you might otherwise need in the SELECT statement.

Prerequisites

- For remote parameterized SQL views, the remote source is based on the `hanaodbc` adapter.
- For remote parameterized ABAP CDS view entities, the remote source is based on the `abapodbc` adapter.
- The remote parameterized view exists on the remote source.

Creating a Virtual Parameterized View

You can create a virtual parameterized view using the SQL statement CREATE VIRTUAL TABLE, specifying the schema (optional) and name of the virtual parameterized view and the name of the remote parameterized view on the remote source:

```
CREATE VIRTUAL TABLE [<schema_name>.]<virtual_table_name>
AT
"<remote_source>"."<database_name>"."<remote_schema_name>"."<remote_view_name>";
```

> **Note**
>
> The WITH REMOTE clause is not supported in the CREATE VIRTUAL TABLE statement.

A virtual parameterized view can be refreshed or deleted using the same SQL syntax as for virtual tables. For more information, see *Refresh a Virtual Table*, *Delete a Virtual Table*, and *Check Virtual Table Definitions*.

Example: Remote SAP HANA Parameterized SQL View

On the remote SAP HANA database, the following statement is executed to create the parameterized SQL view PSV1:

```
CREATE VIEW PSV1 (IN STR1 NVARCHAR(10), IN STR2 NVARCHAR(10) DEFAULT '', IN I3
INT DEFAULT 0) AS (
    SELECT CONCAT(:STR1, :STR2) AS C1, :I3 + 100 AS C2 FROM DUMMY
);
```

In the local on-premise SAP HANA database, the virtual parameterized SQL view VPSV1 is created as follows based on the remote view PSV1 on the remote source HANA1. The schema of PSV1 is MYSCHEMA2:

```
CREATE VIRTUAL TABLE VPSV1 AT "HANA1"."<NULL>"." MYSCHEMA2"."PSV1";
```

The virtual parameterized SQL view PSV1 is called specifying all parameters:

```
SELECT * FROM VPSV1('PARAM', 'VIEW', 5);
```

The virtual parameterized SQL view PSV1 is called specifying one parameter and using default values for the others:

```
SELECT * FROM VPSV1('QUERY');
```

Example: Remote Parameterized ABAP CDS View Entity

In the local on-premise SAP HANA database, you create a virtual table and a virtual parameterized view from the CDS view entities ORDERS (no parameters) and ORDERS_P (with parameters) on the remote source ABAP1.

- Virtual table based on the remote CDS view entity ORDERS in the schema TAPC1/SQL_TPCH_1:

  ```
  CREATE VIRTUAL TABLE VT_ORDERS AT "ABAP1"."<NULL>"."/ITAPC1/
  SQL_TPCH_1"."ORDERS";
  ```

- Virtual parameterized view based on the remote parameterized CDS view entity ORDERS_P in the schema TAPC1/SQL_TPCH_1:

  ```
  CREATE VIRTUAL TABLE VT_ORDERS_P AT "ABAP1"."<NULL>"."/ITAPC1/
  SQL_TPCH_1"."ORDERS_P";
  ```

You check that the virtual table and view have been created corectly:

```
SELECT * FROM VIRTUAL_TABLES WHERE remote_source_name='ABAP1';
```

Result (not all columns are shown below):

SCHEMA_NAME	TABLE_NAME	REMOTE_SOURCE_NAME	REMOTE_DB_NAME	REMOTE_OWNER_NAME	REMOTE_OBJECT_NAME	HAS_PARAMETERS
SYSTEM	VT_ORDERS	ABAP1	<NULL>	TAPC1/SQL_TPCH_1	ORDERS	FALSE
SYSTEM	VT_ORDERS_P	ABAP1	<NULL>	TAPC1/SQL_TPCH_1	ORDERS_P	TRUE

You run the following query on the non-parameterized CDS view entity, using a WHERE clause:

```
SELECT * FROM VT_ORDERS WHERE O_ORDERKEY=1;
```

You run the following query on the parameterized CDS view entity, using a named parameter:

```
SELECT * FROM VT_ORDERS_P(P_ORDERKEY => 1);
```

Monitoring

You can find information about virtual parameterized views in the following system views:

View	Description
VIRTUAL_TABLES System View	Provides information about virtual tables. The HAS_PARAMETERS column indicates whether the virtual table has parameters.
VIRTUAL_TABLE_PARAMETERS System View	Provides information about the parameters of virtual tables

Limitations

- Default parameter values defined in remote parameterized views are supported in virtual parameterized views. However, the HAS_DEFAULT_VALUE column in the VIRTUAL_TABLE_PARAMETERS system view shows FALSE even if a virtual parameterized view has default parameter values.
- Exporting catalog and remote data is not supported for virtual parameterized views. Virtual parameterized views are implicitly skipped in both schema export and object export (an error is not thrown).

Related Information

Refresh a Virtual Table [page 1631]
Delete a Virtual Table [page 1632]
Check Virtual Table Definitions [page 1637]
CREATE VIEW Statement (Data Definition)

15.1.8 Virtualizing Table User-Defined Functions

In the local SAP HANA system, you can create virtual table user-defined functions (TUDFs) that point to remote table user-defined functions in another SAP HANA on-premise system or in an SAP HANA Cloud, SAP HANA database.

A virtual table user-defined function lets you pass parameters to the corresponding remote table user-defined function to retrieve data from the remote database, which is returned as a table. Virtual table user-defined functions are read only and can only be used with SELECT statements. They cannot be used with other data manipulation statements such as INSERT, DELETE, UPDATE, REPLACE, or MERGE INTO.

Limitations

- Virtual scalar user-defined functions (SUDFs) that point to remote scalar user-defined functions are not supported.
- Default parameter values defined in remote user-defined functions are supported in virtual user-defined functions. However, the HAS_DEFAULT_VALUE column in the VIRTUAL_FUNCTION_PARAMETERS system view shows FALSE even if a virtual user-defined function has default parameter values.
- The linked database feature is not supported for virtual table user-defined functions.
- Exporting catalog and remote data is not supported for virtual functions. Virtual functions are implicitly skipped in both schema export and object export (an error is not thrown).

Related Information

Create a Virtual Table User-Defined Function [page 1645]
Refresh a Virtual Table User-Defined Function [page 1646]
Delete a Virtual Table User-Defined Function [page 1647]
System Views for Monitoring Virtual Table User-Defined Functions [page 1648]

15.1.8.1 Create a Virtual Table User-Defined Function

Create a virtual table user-defined function (TUDF) that points to a remote table user-defined function in another SAP HANA on-premise system or in an SAP HANA Cloud, SAP HANA database.

Prerequisites

- The remote source is based on the `hanaodbc` adapter.
- The remote table user-defined function exists on the remote source. It must not have the following types of input parameters:
  - Table variable
  - Array
  - Ciphertext
- One of the following applies:
  - You created the remote source.
  - You have the CREATE VIRTUAL FUNCTION object privilege on the remote source.

Procedure

In an SQL console, execute the following command specifying the schema (optional) and name of the virtual function and the name of the remote function on the remote source:

```
CREATE VIRTUAL FUNCTION [<schema_name>.]<virtual_function_name>
AT
"<remote_source>"."<database_name>"."<remote_schema_name>"."<remote_function_name
>";
```

Example

On the remote SAP HANA database, the following statement is executed to create the table user-defined function TUDF1:

```
CREATE FUNCTION TUDF1 (IN P1 INT, P2 NVARCHAR(100), P3 DECIMAL, P4 DECIMAL(38,2))
RETURNS TABLE (C1 INT, C2 NVARCHAR(200), C3 DECIMAL, C4 DECIMAL(38,2))
LANGUAGE SQLSCRIPT AS
BEGIN
    RETURN SELECT :P1+1 AS C1, CONCAT(:P2,'+ADDED') AS C2, :P3*3 AS C3, :P4/2 AS
C4 FROM DUMMY;
END;
```

In the local SAP HANA system, the virtual function VTUDF1 is created as follows based on the remote function TUDF1 on the remote source HANA1. The remote database name is DB1 and the schema of TUDF1 is MYSCHEMA2:

```
CREATE VIRTUAL FUNCTION MYSCHEMA1.VTUDF1 AT "HANA1"."DB1"."MYSCHEMA2"."TUDF1";
```

The virtual function VTUDF1 is called using named parameters:

```
SELECT * FROM VTUDF1(P1 => 1, P2 => 'abcdefg', P3 => 12.1234, P4 => 56.78);
```

The virtual function VTUDF1 is called based on the order of the parameters in the function definition:

```
SELECT * FROM VTUDF1(1, 'abcdefg', 12.1234, 56.78);
```

15.1.8.2 Refresh a Virtual Table User-Defined Function

Update a virtual table user-defined function (TUDF) to reflect metadata changes in the corresponding remote table user-defined function.

Prerequisites

Requires the CREATE VIRTUAL FUNCTION object privilege.

Context

When changes are made to the metadata in a remote table user-defined function, the changes are not automatically reflected in the corresponding virtual table user-defined function. You must manually update the virtual table user-defined function to reflect the changes.

> **Note**
>
> There is no virtual function consistency check to check for mismatches between the definition of a virtual user-defined function in the local system and the definition of the remote user-defined function it points to.

Procedure

To refresh the virtual function, execute the following:

```
ALTER VIRTUAL FUNCTION [<schema_name>.]<virtual_function_name> REFRESH
DEFINITION;
```

> **Sample Code**
>
> ```
> ALTER VIRTUAL FUNCTION VTUDF1 REFRESH DEFINITION;
> ```

15.1.8.3 Delete a Virtual Table User-Defined Function

Remove an existing virtual table user-defined function (TUDF) from the local SAP HANA system.

Prerequisites

One of the following applies:

- You created the virtual user-defined function.
- You have the DROP privilege on the virtual user-defined function.

Procedure

To delete the virtual function, execute the following:

```
DROP FUNCTION [<schema_name>.]<virtual_function_name>;
```

Sample Code

```
DROP FUNCTION VTUDF1;
```

Related Information

DROP FUNCTION Statement (Procedural)

15.1.8.4 System Views for Monitoring Virtual Table User-Defined Functions

You can find information about virtual table user-defined functions (TUDFs) in various system views.

View	Description
VIRTUAL_FUNCTION_PARAMETERS System View	Provides information about the parameters of virtual functions
VIRTUAL_FUNCTION_PARAMETER_COLUMNS System View	Provides information about the columns in the table parameters of virtual functions
VIRTUAL_FUNCTIONS System View	Provides information about virtual functions. The FUNCTION_USAGE_TYPE column indicates the type of virtual function.

15.1.9 Monitor Remote Connections and Statements

Monitor active connections and running statements on remote connections by using the SQL console.

Context

Use the SQL console to monitor the following:

Remote connections active in the database	Provides details about the connections that were opened in the current session, including when the connection was opened, how many remote statements were executed, and the name of the remote source.
Remote statements executed in the database	Allows you to see the full SQL text of the SQL statements executed on remote sources. It also shows you when the query was started, how long the query took, and the number of records that were returned.

Procedure

1. Open an SQL console.
2. For information on remote statements, execute a SELECT statement using the M_REMOTE_STATEMENTS system view.

   ```
   SELECT * FROM SYS.M_REMOTE_STATEMENTS;
   ```

3. For information on connections, execute a SELECT statement using the M_REMOTE_CONNECTIONS system view.

   ```
   SELECT * FROM SYS.M_REMOTE_CONNECTIONS;
   ```

Results

For details on the system views, see M_REMOTE_STATEMENTS System View and M_REMOTE_CONNECTIONS System View.

Related Information

M_REMOTE_STATEMENTS System View
M_REMOTE_CONNECTIONS System View
Monitor Remote Statements and Connections (SAP HANA Cockpit)

15.1.10 Data Type Support

Data type support varies by remote source.

The SAP HANA Smart Data Access 2.0 master note lists by remote source how data types are mapped to SAP HANA. Data types not listed are not supported.

Related Information

SAP Note 2352696

15.1.10.1 Supported Data Types for Remote Join Pushdown and Remote Join Relocation

The data types accessed in a query are one of the factors that determine whether remote join pushdown and remote join relocation can be performed.

Even if hints such as REMOTE_JOIN_RELOCATION and REMOTE_JOIN are appended to the SQL statement, the join is not pushed down to the remote source if the applicable criteria are not met. Instead, the join is executed locally.

For join pushdown to be able to work, each field accessed in the remote table (in a SELECT clause, WHERE clause, JOIN clause, and so on) must be defined with one of the data types listed below:

Supported Data Types

INT	DECIMAL_DIGIT_ARRAY	NVARCHAR
SMALLINT	DECIMAL	STRING
TINYINT	SMALLDECIMAL	NSTRING
BIGINT	REAL	SECONDDATE
FIXED8	DOUBLE	LONGDATE
FIXED12	CHAR	SECONDTIME
FIXED16	VARCHAR	DAYDATE

Related Information

Join Relocation for Multiple Remote Sources [page 1619]

15.1.10.2 Support for SAP HANA Spatial

Smart data access supports spatial capabilities if the remote source is SAP HANA.

SAP HANA Spatial includes a multilayered spatial engine and supports spatial columns, spatial access methods, and spatial reference systems. SAP HANA Spatial is an SAP HANA optional component.

In the remote source, the ST_POINT data type maps to the ST_GEOMETRY data type in the SAP HANA virtual table.

Support for SAP HANA Spatial has some functional restrictions. For a list of these restrictions, see SAP Note 2609914 - *Smart Data Access - Functional Restrictions for Supporting Spatial Data Types*.

For information about SAP HANA Spatial, see *SAP HANA Spatial Reference*.

Related Information

Configure Transfer Formats for Spatial Data Types [page 1651]
SAP Note 2609914
SAP HANA Spatial Reference

15.1.10.2.1 Configure Transfer Formats for Spatial Data Types

Spatial data is encoded using one of the following ODBC standard formats: WKB (well-known binary) or WKT (well-known text). The default transfer format is WKB. If binary transfer is disabled, extended transfer formats should be used for data federation between the SAP HANA systems.

Context

Binary transfer can be disabled explicitly. It can also be disabled internally because certain data types are not supported or because the binary versions of the two SAP HANA systems are not the same.

If binary transfer has been disabled, extended transfer formats are needed. The extended transfer formats EWKB (extended well-known binary) and EWKT (extended well-known text) include the spatial reference system identifier (SRID). You can set an extended transfer format using the `spatial_xfer_format` parameter in the `indexserver.ini` file.

Procedure

Execute the following statement to set the `spatial_xfer_format` parameter in the `indexserver.ini` file to either 'EWKB' or 'EWKT':

Sample Code

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini', 'SYSTEM') SET
('smart_data_access', 'spatial_xfer_format') = 'EWKB' WITH RECONFIGURE;
```

Option	Description
EWKB **(extended well-known binary)**	If you do not have empty POINT data, use 'EWKB' since it provides better performance.
EWKT **(extended well-known text)**	If you have empty POINT data, use 'EWKT'. If you do not set 'EWKT', empty POINT data will be mapped to MULTIPOINT objects that contain a single empty POINT.

Related Information

ALTER SYSTEM ALTER CONFIGURATION Statement (System Management)
Smart Data Access System Parameters [page 1655]

15.1.11 Functions Pushed Down to Remote Sources

When executing queries on virtual tables that use SAP HANA functions, whenever possible smart data access pushes execution of the function to the remote source to improve query performance.

The ability to push an SAP HANA function down to a remote source depends on the capabilities of the remote source. In some instances, where the SAP HANA function does not map one-to-one to a remote source function, but an equivalent remote function is available, the equivalent remote function is used. If a remote source doesn't support an SAP HANA function, the SAP HANA function is executed on the local host.

The SAP HANA Smart Data Access 2.0 master note lists by remote source the SAP HANA functions that can be pushed down, and where applicable, the equivalent remote source function. Where the SAP HANA function does not map one-to-one, the list indicates what equivalent function is pushed down.

Related Information

SAP Note 2352696

15.1.12 Synonyms

Use synonyms to create an alternative name for a virtual table or a remote table when using linked database with SAP HANA view modeling.

When creating the synonym for linked database, specify the three part name. For virtual tables, specify `<identifier>` or `<schema_name>.<identifier>`.

```
CREATE SYNONYM <synonym_name> FOR [ [
<database_name>.]<schema_name>.]<identifier>;
```

where:

Variable	Description
`<database_name>`	Specifies the name of the remote source for linked database. This value is not applicable to virtual tables.
`<schema_name>`	For linked database, specifies the schema name of the table on the remote source. For virtual tables, specifies the local schema and virtual table name.

Variable	Description
`<identifier>`	For linked database, specifies the name of the table on the remote source. For virtual tables, specifies the name of the local virtual table.

Example

Create synonym TABLE1_SYNONYM for table ADMIN.TABLE1 on remote source SOURCE1.

```
CREATE SYNONYM TABLE1_SYNONYM FOR SOURCE1.ADMIN.TABLE1;
```

15.1.13 Automatic Failover Support

If a connection to a remote source becomes unavailable, then the remote source automatically reconnects to the failover node.

Automatic failover functionality depends on the ODBC connection configuration and whether the remote source itself supports failover.

Currently, the following ODBC connections support failover:

- SAP HANA
- SAP Adaptive Server Enterprise (ASE)
- Oracle

To enable support for failover, configure the remote source using the remote databases ODBC properties. If a connection to the remote database becomes unavailable due to failover, then SAP HANA automatically reconnects to the failover node.

For details on enabling failover, see the specific database under *Creating Remote Sources* .

Related Information

Creating Remote Sources [page 1550]

15.1.14 Safe Mode for ODBC Connections

Provides the capability to load ODBC drivers and execute ODBC calls from within the scriptserver process. This reduces potential issues with the indexserver caused by third-party ODBC drivers.

Currently, smart data access loads the ODBC drivers required for communication with remote database as shared objects directly in the indexserver process. This can be problematic as most of these objects are

third-party libraries and may not provide the stability and quality expected from SAP HANA. Any bug in the third-party ODBC driver library may cause the indexserver to crash, impacting customer productivity.

To address this, smart data access allows the scriptserver to be used as a remote driver manager service. With this configuration the ODBC driver manager and ODBC drivers are loaded and executed in the scriptserver.

This functionality is configured in the *smart_data_access* > *odbc_adapters_in_scriptserver* property in the `indexserver.ini` file and can be enabled and disabled (default) for individual remote sources.

Enter a comma-separated list of the adapter types to use the scriptserver. Valid entries include:

Supported Database	Adapter Name
SAP HANA	hanaodbc
SAP IQ SAP HANA Cloud, data lake Relational Engine	iqodbc
SAP Adaptive Server Enterprise (ASE)	aseodbc
SAP HANA Accelerator for ASE	ets
SAP MaxDB	maxdb
ABAP system	abapodbc
Teradata	tdodbc
SQL Server	mssql
IBM DB2	db2
IBM Netezza	netezza
Oracle	oracle
Google BigQuery	bigquery
Hive	hiveodbc

In addition, the following entries can be used to configure multiple adapters at once:

Supported Database	Logical Adapter Group Name
SAP IQ SAP ASE SAP HANA SAP HANA Cloud, data lake Relational Engine ABAP system	hana_family
All supported databases excluding those included in hana_family	3rd_party
All supported databases	all

The scriptserver, disabled by default, must be enabled to use safe mode. See SAP Note 1650957 – *SAP HANA Database: Starting the Script Server*.

15.1.15 Session-Specific Information for Connections

Set session-specific client information on SAP HANA remote source connections.

When creating remote sources, you add syntax to specify the session information as follows:

```
sessionVariable:<session_variable_name>=?
```

When the connection is made via smart data access to the remote source, the ? is replaced with the value of the variable in the local current session context and is added to the connection string. This behavior allows values of local session variables to be used to set session variables on the remote SAP HANA system.

Variable replacement is done for every connection that is made to the remote source. Each connection to the local SAP HANA system has its own session. When a session executes a query using a virtual table, it establishes a connection to the remote SAP HANA system. The value of the variable used to generate the connection string depends on the value of the variable in the local SAP HANA session.

For example, you have set session information on the remote SAP HANA system using the variables `MY_VAR1` and `MY_VAR2`. In the local session, the value of the variable `MY_VAR1` is abc and the value of the variable `MY_VAR2` is 100. When the connection to the remote SAP HANA system is established, the following values are added to the connection string: `sessionVariable:MY_VAR1=abc; sessionVariable:MY_VAR2=100`.

> **Note**
>
> The session variable is a static variable. It is set only once when the first connection to the remote source is established by a query that is executed on a virtual table based on this remote source. This means that if the value of the variable in the local SAP HANA session is changed, the session variable set on the remote source connection is not updated to reflect these changes.

This feature is supported for SAP HANA remote sources only. For configuration information, see *Create an SAP HANA Remote Source*.

Related Information

Create an SAP HANA On-Premise Remote Source [page 1563]
SET [SESSION] Statement (Session Management)

15.1.16 Smart Data Access System Parameters

Configuration parameters for smart data access are available in the `smart_data_access` and `linked_database` sections of the `indexserver.ini` file.

Default values should not be changed without direction from SAP support.

smart_data_access Section

Parameter Name	Description	Type	Length	Values	Default Value	Hidden
enable_binary_transfer	Specifies whether binary transfer should be used for remote execution.	BOOLEAN		TRUE FALSE	TRUE	YES
enable_remote_source_capability	Specifies the complexity of queries to be sent to the remote sources.	BOOLEAN		TRUE = any query in the remote source dialect can be sent for remote execution FALSE = only projections are sent for remote execution	TRUE	NO
semi_join_execution_strategies	Specifies the preferred order of semi-join execution strategies.	VARCHAR	16	IT = attempt of in-clause strategy followed by attempt of temporary table strategy TI = attempt of temporary table strategy followed by attempt of in-clause strategy T = temporary table strategy I = in-clause strategy N = turns off the semi-join	IT	NO
semi_join_max_in_elements	Specifies the maximum number of values in the IN clause for semi-join usage.	INTEGER		Positive integer value	1024	NO
semi_join_max_temp_table_cardinality	Specifies the maximum number of values to be inserted in a semi-join temp table.	INTEGER		Positive integer value	16384	NO
semi_join_min_temp_table_cardinality	Specifies the minimum number of values to be inserted in a semi-join temp table.	INTEGER		Positive integer value		YES

Parameter Name	Description	Type	Length	Values	Default Value	Hidden
semi_join_reduction_factor	Specifies the estimated percentage reduction required for an attribute to be considered for semi-join reduction.	TINYINT		Positive integer value		YES
semi_join_virtual_table_threshold	Specifies the minimum number of estimated rows for fact subplan, to be considered for semi-join reduction.	TINYINT		Positive integer value		YES
spatial_xfer_format	Specifies the transfer format to be used for spatial data types. An extended transfer format should be used if binary transfer is disabled between the SAP HANA systems.	VARCHAR	16	WKB = well-known binary WKT = well-known text EWKB = extended well-known binary EWKT = extended well-known text	WKB	YES
virtual_table_format	Forces the optimizer to choose column or row-based operators.	VARCHAR	16	ROW = row based COLUMN = column based AUTO = let the optimizer choose	AUTO	NO

linked_database Section

Parameter Name	Description	Type	Length	Values	Default Value	Hidden
linked_database_cleanup_interval	Specifies the interval in seconds to perform linked object housekeeping tasks.	integer		Positive integer value in seconds 0 = task is disabled	0	NO

Related Information

Configure Transfer Formats for Spatial Data Types [page 1651]

15.1.17 Smart Data Access Performance Tools

There are several tools available that can impact performance for smart data access.

15.1.17.1 Managing Statistics

Statistics assist the query optimizer in making better decisions and work for both virtual tables and linked database.

15.1.17.1.1 Create Statistics on a Virtual Table or Linked Database

Create data statistic virtual objects that the query optimizer uses to make better decisions for query plans.

Prerequisites

One of the following:

- You created the virtual table you are creating statistics on.
- You have the ALTER privilege on the object you are creating statistics on.
- For linked database with linked database optimized mode disabled, you require the LINKED DATABASE object level privilege on the remote source, regardless of who created the remote source. No additional privilege is required if optimized mode is enabled.

Context

To verify that the statement executed correctly, see *Monitor Remote Connections and Statements*.

Procedure

To create statistics on a virtual table or linked database, execute:

```
CREATE STATISTICS <data_statistics_name> ON <data_sources>
 <data_statistics_type>
 [ <data_statistics_properties> ]
 [ <initial_refresh> ];
```

<data_statistics_name>

Specifies a unique name for the data statistics object.

```
<data_statistics_name> ::= [ <schema_name>.]<identifier>
```

<data_statistics_name> is only allowed when the result of the creation is a single data statistics object. The number of data statistics objects created by CREATE STATISTICS is determined by the combination of <data_statistics_type> and the number of columns specified in <data_sources>.

```
<schema_name> ::= <identifier>
```

<schema_name> must be the same as specified for data source.

<data_sources>

Specifies the data source you want to create data statistics objects for.

```
<data_sources> ::=
<table_name> [ ( <column_name>[, <column_name>[,...] ] ) [
<match_type> ]
```

For RECORD COUNT data statistics objects, you cannot specify columns as part of <data_sources>.

<table_name>

Specifies the table name you want to create statistics on.

```
<table_name> ::= [
<database_name>.]<schema_name>.]<identifier>
```

For linked database, <database_name> is the name of the remote source. For all other cases, <database_name> is the name of the database where the table is located.

<column_name>

Specifies the column for which the data statistics are defined.

```
<column_name> ::= <identifier>
```

If no <column_name> is specified, then all statistics for the table that match the specified properties are altered, including table-wide statistics (RECORD COUNT).

<data_statistics_type>

Specifies the type of data statistics object to create.

```
<data_statistics_type> := TYPE <type_name>
<type_name> ::=
HISTOGRAM
 | SIMPLE
 | TOPK
 | SKETCH
 | SAMPLE [ <sample_size_modifier> ]
 | RECORD COUNT
```

A data source can have only one data statistics object of a certain type. For example, column A of table T can have one data statistics object of type HISTOGRAM and one of

type SIMPLE. If the TYPE clause is not specified, then the default is HISTOGRAM. Some data statistic types may not be appropriate for a given data source.

HISTOGRAM	Creates a data statistics object that helps the query optimizer estimate the data distribution in a single-column data source. If you specify multiple columns in `<data_sources>`, then multiple data statistics objects (HISTOGRAM) are created--one per column specified.
SIMPLE	Creates a data statistics object that helps the query optimizer calculate basic statistics, such as min, max, null count, count, and distinct count for a single-column data source. If you specify multiple columns in `<data_sources>`, then multiple data statistics objects are created--one per column specified. When beneficial, the SQL optimizer maintains system SIMPLE data statistics objects automatically on column and row store tables only.
TOPK	Creates a data statistics object that helps the query optimizer identify the highest-frequency values in a table data source. If you specify multiple columns in `<data_sources>`, then multiple data statistics objects are created--one per column specified. When beneficial, the SQL optimizer maintains system TOPK data statistics objects automatically (column store only).
SKETCH	Creates a data statistics object that helps the query optimizer estimate the number of distinct values in the data source. A data statistics object is created for the specified `<table_name>(<column-name>,...)`, which approximates the number of distinct tuples in the projection of the table on the set of specified columns.
SAMPLE	Creates a sample of data from `<data_source>` that the SQL optimizer can use during optimization. When beneficial, the SQL optimizer generates system SAMPLE data statistics objects automatically on column and row store tables. However, this behavior can incur a cost to performance. You can avoid this cost by creating SAMPLE data statistics objects explicitly (in advance). Creating them explicitly is especially useful in situations where sampling live table data is expensive (for example, very large tables). <sample_size_modifier> :: = SAMPLE SIZE <unassigned_integer> defines the sample size. SAMPLE SIZE <n> is optional; if it is not given, the default value is 1000.
RECORD COUNT	Creates a data statistics object that helps the query optimizer calculate the number of records (rows) in a table data source. The RECORD COUNT type is a table-wide statistic. You do not specify columns in `<data_sources>` when creating a RECORD COUNT data statistics object. When beneficial, the SQL optimizer maintains system RECORD COUNT data statistics objects automatically on column and row store tables.

`<data_statistics_properties>`

Specifies the properties of the data statistics object.

```
<data_statistics_properties> ::=
 <data_statistics_property>[,<data_statistics_property>[,...] ]

<data_statistics_property> ::=
 REFRESH TYPE <refresh_type>
 | ENABLE <on_off>
 | BUCKETS <unsigned_integer>
 | QERROR <numeric_literal>
 | QTHETA <unsigned_integer>
 | { MEMORY <memory_bytes> | MEMORY PERCENT
<memory_percentage> }
 | PERSISTENT <on_off>
 | VALID FOR <valid_for_list>
 | CONSTRAINT '<constraint_param>'
```

Restrictions to which properties apply to which statistic types are noted in the property descriptions.

REFRESH TYPE <refresh_type>

Specifies the strategy for the data statistics object.

```
<refresh_type> ::= { MANUAL | DEFAULT }
```

MANUAL specifies that the database statistics object is not refreshed until a rebuild is explicitly requested by a REFRESH STATISTICS statement.

DEFAULT specifies that the database server decides the best refresh strategy based on the data source. Since only MANUAL is supported for virtual tables, DEFAULT uses MANUAL.

REFRESH TYPE only affects data statistics objects that are enabled.

ENABLE <on_off>

Controls whether the optimizer uses the data statistics object.

```
<on_off> ::= ON | OFF
```

ENABLE ON enables the optimizer to see the data statistics object. The data statistics object must be populated with data for the optimizer to use it. ENABLE ON specified with NO INITIAL REFRESH returns an error.

ENABLE ON is the default behavior.

ENABLE OFF disables the use of the data statistics object by the optimizer and prevents the ability to refresh the data statistics object. Data statistics objects that are not enabled can still be dropped. To make a data statistics object with ENABLE OFF accessible to the optimizer, execute an ALTER STATISTICS...ENABLE ON statement.

BUCKETS <unsigned_integer>

The BUCKETS property is only for use with TYPE HISTOGRAM or TOPK. For HISTOGRAM, BUCKETS

	specifies the maximum number of data buckets in the HISTOGRAM. For TOPK, BUCKETS specifies the K value. The default is automatically determined by the data statistics building algorithm in use.
CONSTRAINT **<constraint_param>**	Specifies constraints to use for the specified <data_statistics_type>. The CONSTRAINT property is not supported for the data statistics type SKETCH.

<initial_refresh>

Specifies whether to populate the data statistics object with data after creation.

```
<initial_refresh> ::= [ NO ] INITIAL REFRESH
```

INITIAL REFRESH	Creates the definition of the data statistics object and populates it with data. The default behavior is INITIAL REFRESH.
NO INITIAL REFRESH	Creates the definition of the data statistics object, but does not populate it with data. Use NO INITIAL REFRESH when you want to change the underlying data before refreshing the data statistics object. You cannot specify NO INITIAL REFRESH if ENABLE OFF is not specified.

Examples

Create SIMPLE statistics on virtual table `HANA1_T1`.

```
CREATE STATISTICS "TEST1" ON HANA1_T1 (A1) TYPE SIMPLE;
```

Using linked database, create TOPK statistics on table `T1` using remote source `HANA1` and schema `MYSCHEMA` with 10 buckets.

```
CREATE STATISTICS ON HANA1.MYSCHEMA.T1 (A1) TYPE TOPK BUCKETS 10;
```

Related Information

Monitor Remote Connections and Statements [page 1648]
Alter Statistics on a Virtual Table or Linked Database [page 1663]
Using Linked Database [page 1624]

15.1.17.1.2 Alter Statistics on a Virtual Table or Linked Database

Alter the properties of a data statistic object for virtual tables or linked database.

Prerequisites

One of the following:

- You own the virtual table you are altering statistics on.
- You have the ALTER privilege on the object you are altering statistics on.
- For linked database, you require the LINKED DATABASE object level privilege on the remote source, regardless of who created the remote source.

Procedure

To alter statistics on a virtual table or linked database, execute:

```
ALTER STATISTICS { <data_statistics_name> [,...] | ON <data_sources>
[ [ HAVING ] <match_properties> ] }
 [ SET <set_data_statistics_properties> ]
 [ <initial_refresh> ]
```

<data_statistics_name>

Specifies the name of the data statistics object.

```
<data_statistics_name> ::= [<schema_name>.]<identifier>
<schema_name> ::= <identifier>
```

<data_sources>

Specifies the data source(s) of the data statistics objects.

```
<data_sources> ::=
<table_name> [ ( <column_name>[, <column_name>[,...] ] ) [
<match_type> ]
```

For RECORD COUNT data statistics objects, you cannot specify columns as part of <data_sources>.

<table_name> Specifies the table on which the data statistics are defined.

```
<table_name> ::= [ [
<database_name>.]<schema_name>.]<identifier>
```

For linked database, <database_name> is the name of the remote source. For all other cases, <database_name> is the name of the database where the table is located.

<column_name>

Specifies the column for which the data statistics are defined.

```
<column_name> ::= <identifier>
```

If no <column_name> is specified, then all statistics for the table that match the specified properties are altered, including table-wide statistics (RECORD COUNT).

<match_properties>

Specifies properties to use for matching when selecting data statistics.

```
<match_properties> ::= <match_property>[...]
<match_property> ::=
 TYPE <data_statistics_type>  | REFRESH TYPE
<refresh_type_filter>
```

If TYPE is not specified, then all data statistics objects of any type on the specified data sources are altered (ALL). For descriptions of the supported data statistics types see Create Statistics on a Virtual Table or Linked Database [page 1658].

<data_statistics_type>

Specifies the type of data statistics objects to match when selecting the data statistics.

```
<data_statistics_type> := TYPE
<type_name>
<type_name> ::=
 HISTOGRAM
  | SIMPLE
  | TOPK
  | SKETCH
  | SAMPLE
  | RECORD COUNT
  | ALL
```

<refresh_type_filter>

Specifies the refresh strategy to match in the data statistics objects when selecting the data statistics to alter. ALL is the default.

```
<refresh_type_filter> ::= MANUAL | ALL
```

<set_data_statistics_properties>

Specifies the properties of the data statistics objects to modify.

```
<data_statistics_properties> ::=
 <data_statistics_property> [<data_statistics_property>]...

<data_statistics_property> ::=
  | REFRESH TYPE <refresh_type>
  | ENABLE <on_off>
  | BUCKETS <unsigned_integer>
  | PERSISTENT <on_off>
```

```
| CONSTRAINT <constraint_param>
```

REFRESH TYPE **<refresh_type>**

Specifies the strategy for the data statistics object.

```
<refresh_type> ::= { MANUAL | DEFAULT }
```

MANUAL specifies that the database statistics object is not refreshed until a rebuild is explicitly requested by a REFRESH STATISTICS statement.

DEFAULT specifies that the database server decides the best refresh strategy based on the data source. Since only MANUAL is supported for virtual tables, DEFAULT uses MANUAL.

REFRESH TYPE only affects data statistics objects that are enabled.

ENABLE **<on_off>**

Controls whether the optimizer uses the data statistics object.

```
<on_off> ::= { ON | OFF }
```

ENABLE ON enables the optimizer to see the data statistics object. The data statistics object must be populated with data for the optimizer to use it.

ENABLE OFF disables the use of the data statistics object by the optimizer and prevents the ability to refresh the data statistics object. Data statistics objects that are not enabled can still be dropped. To make a data statistics object with ENABLE OFF accessible to the optimizer, execute an ALTER STATISTICS...ENABLE ON statement.

BUCKETS **<unsigned_integer>**

The BUCKETS property is only for use with TYPE HISTOGRAM or TOPK. For HISTOGRAM, BUCKETS specifies the maximum number of data buckets in the HISTOGRAM. For TOPK, BUCKETS specifies the K value.

The default is automatically determined by the data statistics building algorithm in use.

CONSTRAINT **<constraint_param>**

Specifies constraints to use for the specified <data_statistics_type>.

The CONSTRAINT property is not supported for the data statistics type SKETCH.

<initial_refresh>

Specifies whether to repopulate the data statistics object with data after altering it.

```
<initial_refresh> ::= [ NO ] INITIAL REFRESH
```

If the object was built, then disabled, and is now being re-enabled, then initial refresh is not required.

INITIAL REFRESH	Alters the definition of the data statistics object and repopulates it with data. The default behavior is INITIAL REFRESH.
NO INITIAL REFRESH	Alters the definition of the data statistics object, but does not repopulate it with data. Use NO INITIAL REFRESH when you want to change the underlying data before refreshing the data statistics object. You cannot specify NO INITIAL REFRESH if ENABLE OFF is not specified.

Examples

Set the number of buckets to 150 for the virtual table Remote1_A1.

```
ALTER STATISTICS ON MYSYSTEM.REMOTE2_A1 TYPE TOPK SET BUCKETS 10 NO INITIAL
REFRESH;
```

The following example sets the number of buckets to 10 on the remote source remote2 using linked database.

```
ALTER STATISTICS ON "remote2"."SYSTEM"."A1" TYPE TOPK SET BUCKETS 10;
```

Related Information

Monitor Remote Connections and Statements [page 1648]
Create Statistics on a Virtual Table or Linked Database [page 1658]

15.1.17.1.3 Refresh Statistics on a Virtual Table or Linked Database

Refreshes data statistic virtual objects that the query optimizer uses to make better decisions for query plans.

Prerequisites

One of the following:

- You own the virtual table you are refreshing statistics on.

- You have the ALTER privilege on the object you are refreshing statistics on.
- For linked database, you require the LINKED DATABASE object level privilege on the remote source, regardless of who created the remote source.

Context

Specify EXACT to refresh a data statistics virtual object that precisely matches `<data_sources>` (including column order). Specify CASCADE to refresh data statistics objects that reference at least one column in `<data_sources>`. If `<match_type>` is not specified, then any data statistics objects that reference all or some of the columns specified in `<data_sources>` are refreshed.

Procedure

To refresh statistics, execute:

```
REFRESH STATISTICS { <data_statistics_name>[,<data_statistics_name>[,...] ]
   | ON <data_sources> [ [HAVING] <match_properties> ] };
```

For descriptions of each supported property, see the Alter Statistics on a Virtual Table or Linked Database [page 1663] topic

Example

Refresh all statistics on virtual table `HANA1_T1`.

```
REFRESH STATISTICS ON MYSCHEMA.HANA1_T1;
```

Using linked database, refresh only SIMPLE statistics on table `T1` using remote source `HANA1` and schema `MYSCHEMA1`.

```
REFRESH STATISTICS ON HANA1.MYSCHEMA1.T1 TYPE SIMPLE;
```

Related Information

REFRESH STATISTICS Statement (Data Definition)
Monitor Remote Connections and Statements [page 1648]

15.1.17.1.4 Drop Statistics on a Virtual Table or Linked Database

Drop data statistic virtual objects that the query optimizer uses to make better decisions for query plans.

Prerequisites

One of the following:

- You own the virtual table you are dropping statistics from.
- You have the ALTER privilege on the object you are dropping statistics from.
- For linked database, you require the LINKED DATABASE object level privilege on the remote source, regardless of who created the remote source.

Procedure

To drop statistics on a virtual table, execute:

```
DROP STATISTICS ON [<schema_name>.] <virtual_table_name>
    [ [HAVING] <match_properties> ];
```

For descriptions of each supported property, see the Alter Statistics on a Virtual Table or Linked Database [page 1663] topic

Example

The following statement drops all statistics on virtual table `HANA1_T1`.

```
DROP STATISTICS ON MYSCHEMA1.HANA1_T1;
```

Using linked database, the following statement drops SIMPLE statistics on table `T1` using remote source `HANA1` and schema `MYSCHEMA2`.

```
DROP STATISTICS ON HANA1.MYSCHEMA2.T1 TYPE SIMPLE
```

Related Information

DROP STATISTICS Statement (Data Definition)
Monitor Remote Connections and Statements [page 1648]

15.1.17.1.5 Retrieve Statistics from a Remote Source

For SAP HANA, SAP IQ, and Teradata remote sources, the statistics for virtual tables and linked database can be retrieved by querying a remote table.

The name of the remote table that is queried for statistics retrieval is as follows:

Remote Source	Remote Table Name
SAP HANA	SYSTEM.SDA_STATISTICS
SAP IQ	SYS_STATISTICS
Teradata	SYS_STATISTICS

The SYS_STATISTICS remote table is located in the default schema of the connection used to create the virtual tables.

When SIMPLE statistics are computed for a virtual table, the remote statistics table is queried first. If this table is not available (or has a different format), then the standard behavior used to obtain statistics from remote sources is triggered; that is, queries are sent to fetch data for each column from the remote table so that the statistics can be computed locally. These statistics are stored in the SYS.DATA_STATISTICS system table.

The remote source capability CAP_STATISTICS allows statistics retrieval at the remote source type level. It is only enabled for SAP HANA, SAP IQ, and Teradata remote sources.

Remote Statistics Table

The schema of the remote statistics table used to store the computed statistics in the remote database is as follows:

Index	Name	Type	Precision	Description
1	SCHEMA_NAME	VARCHAR	128	Schema name
2	TABLE NAME	VARCHAR	128	Table name
3	COLUMN_NAME	VARCHAR	128	Column name
4	MIN	VARCHAR	128	String representation of the min value
5	MAX	VARCHAR		String representation of the max value
6	COUNT_STAR	INTEGER		Count (*)
7	DCOUNT	INTEGER		Distinct count
8	COUNT	INTEGER		Count (used to count NULL values)

Creating and Populating a Remote Statistics Table

You need to create and fill the remote statistics table on the remote database. Examples of user stored procedures you could create on the remote database to create and populate data statistics for a given table are shown below.

> ⓘ Note
>
> You need to manage the remote statistics table yourself. If there are any changes in the statistics of the remote objects, the table should be updated manually.

SAP HANA

> {·} Sample Code

```
CREATE PROCEDURE PR_INSERT_STATS_ON_TABLE(
 IN schemaname varchar(256),
 IN tablename varchar(256)
 )
 LANGUAGE SQLSCRIPT AS
     sys_tab_exists smallint := 0;
     CURSOR c_curs FOR SELECT COLUMN_NAME FROM "SYS"."COLUMNS" WHERE
SCHEMA_NAME=:schemaname AND TABLE_NAME=:tablename;
         BEGIN
            SELECT COUNT(*) INTO sys_tab_exists from "PUBLIC"."M_TABLES"
WHERE SCHEMA_NAME = 'SYSTEM' AND TABLE_NAME = 'SDA_STATISTICS';
            IF (:sys_tab_exists = 0) THEN
                CREATE COLUMN TABLE "SYSTEM"."SDA_STATISTICS" ("SCHEMA_NAME"
VARCHAR(256), "TABLE_NAME" VARCHAR(256), "COLUMN_NAME" VARCHAR(256), "MIN"
VARCHAR(256), "MAX" VARCHAR(256), "COUNT_STAR" INTEGER, "DCOUNT" INTEGER,
"COUNT" INTEGER, PRIMARY KEY ("SCHEMA_NAME", "TABLE_NAME", "COLUMN_NAME"));
            END IF;

            FOR cur_row AS c_curs DO
               EXEC 'UPSERT SYSTEM.SDA_STATISTICS
SELECT ''''||:schemaname||''', '''||:tablename||''', '''||
cur_row.COLUMN_NAME||''', TO_CHAR(MIN(''||cur_row.COLUMN_NAME||'')),
TO_CHAR(MAX(''||cur_row.COLUMN_NAME||'')), COUNT(*), COUNT(DISTINCT
''||cur_row.COLUMN_NAME||''), COUNT(''||cur_row.COLUMN_NAME||'') FROM
''||:schemaname||''."''||:tablename||''";
            END FOR;
        END;
```

SAP IQ

> {·} Sample Code

```
CREATE PROCEDURE PR_SDA_INSERT_STATS_ON_TABLE(IN catalogname varchar(256), IN
schemaname varchar(256),IN tablename varchar(256))
BEGIN
      DECLARE sql varchar(1024);
      -- Create the table SYS_STATISTICS if not exists
      SET SQL = 'IF NOT EXISTS (SELECT 1 from SYSCATALOG WHERE "tname" =
''SYS_STATISTICS'' and creator=''' + schemaname + ''')
                  CREATE TABLE "' + catalogname + '"."' + schemaname +
'"."SYS_STATISTICS" (
                   "SCHEMA_NAME" VARCHAR(256),
                              "TABLE_NAME" VARCHAR(256),
                              "COLUMN_NAME" VARCHAR(256),
                              "MIN" VARCHAR(256),
```

```
                                "MAX" VARCHAR(256),
                                "COUNT_STAR" INTEGER,
                                "DCOUNT" INTEGER,
                                "COUNT" INTEGER)';
      EXECUTE IMMEDIATE WITH RESULT SET ON SQL;
      -- Delete all existing statistics on the given table
      SET SQL = 'DELETE FROM "' + catalogname + '"."' + schemaname +
'"."SYS_STATISTICS"
                                                WHERE "SCHEMA_NAME" = ''' +
schemaname + '''
                                                AND "TABLE_NAME" = ''' +
tablename + '''';
      EXECUTE IMMEDIATE WITH RESULT SET ON SQL;
       -- Loop over columns from the given table
       BEGIN
                        DECLARE err_notfound EXCEPTION FOR SQLSTATE '02000' ;
                        DECLARE c_curs CURSOR  FOR SELECT cname FROM
"SYS"."SYSCOLUMNS" WHERE "creator" = schemaname AND "tname" = tablename ORDER
BY "colno" ASC;
                        DECLARE @COLUMN_NAME varchar(256);
                        OPEN c_curs WITH HOLD;
                          ResultSetLoop:
                          LOOP
                            -- Populate the table with statistics computed
                            FETCH  NEXT c_curs INTO @COLUMN_NAME;
                            IF SQLSTATE = err_notfound THEN
                                LEAVE ResultSetLoop ;
                            END IF ;
                            -- Compute statistics and insert
                            SET SQL = 'INSERT INTO "' + catalogname + '"."' +
schemaname + '"."SYS_STATISTICS"
                                SELECT  ''' + schemaname + ''', ''' +
tablename + ''',  ''' + @COLUMN_NAME  + ''',
                                                STRING(MIN("' + @COLUMN_NAME +
'")) ,
                                                STRING(MAX("' + @COLUMN_NAME +
')),
                                                COUNT(*),
                                                COUNT(DISTINCT "' +
@COLUMN_NAME + '"),
                                                COUNT("' + @COLUMN_NAME + '")
                                FROM "' + catalogname + '"."' + schemaname +
'"."' + tablename + '"' ;
                            EXECUTE IMMEDIATE WITH RESULT SET ON SQL;
                          END LOOP ResultSetLoop;
                        CLOSE c_curs   ;
       END
END
```

Teradata

Sample Code

```
CREATE PROCEDURE PR_INSERT_STATS_ON_TABLE(IN a_schemaName varchar(256),IN
a_tableName varchar(256))
BEGIN
 DECLARE schemaName, tableName, columnName varchar(256);
 DECLARE sys_stats varchar (256);
 DECLARE qualified_tablename varchar(512);

 SET schemaName = a_schemaName;
 SET tableName = a_tableName;
 SET qualified_tablename ='"' || schemaName || '"."' || tableName ||'"';
 SET sys_stats = '"' || schemaName || '"."SYS_STATISTICS"';
 -- Create the table SYS_STATISTICS if not exists
```

```
 IF NOT EXISTS (SELECT 1 FROM DBC.Tables WHERE TableName = 'SYS_STATISTICS'
and DatabaseName = schemaName)
 THEN
   CALL DBC.SysExecSQL('CREATE TABLE ' || sys_stats || ' ("SCHEMA_NAME"
VARCHAR(256), "TABLE_NAME" VARCHAR(256), "COLUMN_NAME" VARCHAR(256), "MIN"
VARCHAR(256), "MAX" VARCHAR(256), "COUNT_STAR" INTEGER, "DCOUNT" INTEGER,
"COUNT" INTEGER)');
 ELSE
   CALL DBC.SysExecSQL('DELETE FROM ' || sys_stats || ' WHERE "SCHEMA_NAME" =
''' || schemaName || ''' AND "TABLE_NAME" = ''' || tableName || '''') ;
 END IF;
 BEGIN
   -- Loop over columns from the given table
   DECLARE c_curs CURSOR FOR
   SELECT ColumnName FROM DBC.COLUMNS WHERE DatabaseName = a_schemaName AND
TableName = a_tableName;
   OPEN c_curs;
   ResultSetLoop:
   LOOP
     -- Populate the table with statistics computed
     FETCH c_curs INTO columnName;
     IF (SQLSTATE = '02000') THEN
       LEAVE ResultSetLoop ;
     END IF ;
     CALL DBC.SysExecSQL('INSERT INTO ' || sys_stats || ' SELECT ''' ||
schemaName || ''','''' || tableName || ''','''' || columnName || ''', CAST
(MIN("' || columnName || '") as VARCHAR(256)) , CAST (MAX("' || columnName
|| '") AS VARCHAR(256)), COUNT(*), COUNT(DISTINCT "' || columnName || '"),
COUNT("' || columnName || '") FROM ' || qualified_tablename);
   END LOOP ResultSetLoop;
   CLOSE c_curs;
 END;
END;
```

Related Information

DATA_STATISTICS System View

15.1.17.2 Pool of Remote Connections

Use a pool of remote connections to scale out multithreaded execution. A pool of remote connections can only be used for read-only remote sources (DML_MODE=readonly).

Provided no external updates (by third parties) occur on the remote source, full consistency is assured. However, if external updates occur in parallel on a declared read-only remote source, then consistency can be ensured only within one SAP HANA thread.

Configuration Parameter for Pool of Remote Connections

Use the parameter `default_connections_pool_max_size` in the `smart_data_access` section of the `indexserver.ini` file to configure a pool of remote connections:

```
indexserver.ini/smart_data_access/default_connections_pool_max_size
```

- Default value: 3
- Highest value allowed: 50
- Value to disable the connection pool: 1

The maximum number of connections allowed in one pool is controlled by the value specified in `default_connections_pool_max_size`. Each SAP HANA connection has its own connection pool for each remote source it uses. The number of connections depends on the degree of multithreading of the executed statements, but cannot exceed the number specified in `default_connections_pool_max_size`. Also, each SAP HANA node has its own connection pool, so `default_connections_pool_max_size` applies per node and is not a global maximum. The query optimizer may decide to increase the degree of parallelism by using multiple SAP HANA nodes for the query execution.

15.1.17.2.1 Configure the Pool of Remote Connections Parameter

Configure the number of remote connections to enable for multi-threaded execution.

Prerequisites

Requires the INIFILE ADMIN system privilege.

Configure the Parameter Using SQL

Procedure

In an SQL console, execute:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','SYSTEM')
    SET ('smart_data_access', 'default_connections_pool_max_size')='<value>';
```

Example

The following statement sets the number of connections to `10`:

```
ALTER SYSTEM ALTER CONFIGURATION ('indexserver.ini','SYSTEM')
    SET ('smart_data_access', 'default_connections_pool_max_size')='10';
```

15.1.17.3 Results Caching for Virtual Tables and Linked Database

Only view results caching is supported for virtual tables and linked database.

You configure result caching using the `<cache_type>` parameter in the CREATE VIEW statement, or the `<alter_cache_settings_clause>` parameter in the ALTER VIEW statement. Only static result caching is supported for virtual tables and linked database. Dynamic result caching is not supported.

Caching of user-defined functions that reference virtual tables is not supported.

When you query a virtual table, if there is insufficient memory to cache the results, then an internal error message appears, the query halts, and no results are cached. To prevent this scenario, modify the view definition if possible, to filter data before caching results, partition caching between multiple views, or disable result caching. You may need to clear the SQL plan cache after modifying the view definition. Using HINT IGNORE_PLAN_CACHE when executing a query allows the system to validate if there is sufficient memory to cache the results before execution and helps you decide if caching should be used. However, since remote data size is only an estimate, this configuration may not prevent the error of insufficient memory.

When creating a view on virtual tables with caching configured, you must apply filters to the view definition not outside the view definition (part of the SQL query referencing the view) whenever possible. Using common filters within the view definition increases the usability and performances of the cache.

When results caching is enabled, only filters applied to the view definition are pushed down to the remote source. Filters outside of the view definition are not pushed. They are applied once the unfiltered results are cached. This behavior potentially impacts how much data is retrieved from the remote source. When results caching is disabled, and the filter is outside the view, the filter may be pushed down to the remote source, returning less data, but the results are not cached. The next time the view is used, the new query is sent to the remote source and new data is returned.

Cached views can have a retention period, which when exceeded triggers a refresh of the view on next use. The frequency of refresh can impact performance. Regularly monitor cache use. The retention period of a view should reflect the nature of the data returned. Apply a longer retention period to views returning data that is more static in nature. It is not recommended that you cache views that are rarely used, regardless of the nature of the data. The constant refresh of the data could impact performance without any potential benefits from caching.

Data in a cached view is available to all users with the SELECT privilege on the view. Remote privileges on virtual tables are validated only when a remote statement is executed, which populates the results cache. The view results cache is available to all users. Any query can populate a view results cache, but the data cached is based on the remote credentials of the current user. As a result, users querying a cached view may see less data than expected if their remote privileges are greater than those of the user who populated the cache. Conversely, users with lesser remote privileges may see data in the shared cache that they normally could not access.

Use cached views only with remote sources using technical user credentials. For all other types of remote authentications, only SAP HANA side authorizations are enforced. When executing a query, to prevent populating the results cache with data specific to credentials, include the HINT RESULT_CACHE_NO_REFRESH clause.

Take careful consideration when enabling caching on views with virtual tables that point to data with static or dynamic analytical privileges or data that cannot be seen as a relational data set (that binds to relational operator rules). Remote constraints on data may not be enforced regardless of whether view results cache is enabled or not.

Result caching is disabled by default. When using HINT RESULT_CACHE in a query on views, if result cache is configured on the views, then result caching is enabled for the query only.

Result caching is also controlled through the `<result_cache>` set of system configuration parameters within `indexserver.ini`.

For full syntax to create, alter, and drop view cache, or to enable result caching, see *SAP HANA SQL and System Views Reference*.

Example

This statement creates a view called `view_tableA` on `tableA` with a static result cache retention period of 10 minutes:

Sample Code

```
CREATE VIEW view_tableA AS SELECT * FROM tableA WITH STATIC CACHE RETENTION
10;
```

This statement changes the retention period from `10` minutes to `5`:

Sample Code

```
ALTER VIEW view_tableA ALTER STATIC CACHE RETENTION 5;
```

Related Information

CREATE VIEW Statement (Data Definition)
ALTER VIEW Statement (Data Definition)

15.1.18 Troubleshooting Smart Data Access

Find solutions to common smart data access problems.

If you don't find your issue in the troubleshooting topics, for a list of known issues see SAP Note 2352696 - *SAP HANA Smart Data Access 2.0 Master Release Note*.

15.1.18.1 Invalid Connection String Message When Querying a Google BigQuery Database

When executing a query using Google BigQuery as a remote source, you receive a message regarding an invalid connection string.

Ensure that the DriverManagerEncoding property in the `simba.googlebigqueryodbc.ini` file on the SAP HANA host is configured to use `UTF-16`. See Google BigQuery ODBC Driver [page 1534].

15.2 SAP HANA Hadoop Integration

Regardless of structure, you can combine the in-memory processing power of SAP HANA with Hadoop's ability to store and process huge amounts of data.

SAP HANA is designed for high-speed data and analytic scenarios, while Hadoop is designed for very large, unstructured data scenarios. Hadoop can scale to thousands of nodes and is designed for use in large distributed clusters and to handle big data. Combining SAP HANA with Hadoop leverages Hadoop's lower storage cost and type flexibility with the high-speed in-memory processing power and highly structured data conformity of SAP HANA.

SAP HANA Hadoop integration is designed for users who may want to start using SAP HANA with their Hadoop ecosystem.

For information about SAP HANA Hadoop Integration and SAP HANA Spark controller installation, see SAP HANA Hadoop Integration.

16 Important Disclaimer for Features in SAP HANA

For information about the capabilities available for your license and installation scenario, refer to the Feature Scope Description for SAP HANA.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see https://www.sap.com/about/legal/trademark.html for additional trademark information and notices.