PUBLIC
SAP HANA Platform 2.0 SPS 05
Document Version: 1.1 – 2021-07-30

SAP HANA System Replication

THE BEST RUN SAP

Content

1 SAP HANA System Replication

The SAP HANA System Replication guide is the entry point for all information related to SAP HANA system replication.

Why should I use SAP HANA system replication?

SAP HANA system replication is a mechanism ensuring the high availability of your SAP HANA system. System replication is SAP's recommended configuration for addressing SAP HANA outage reduction due to planned maintenance, faults, and disasters. It supports a recovery point objective (RPO) of 0 seconds and a recovery time objective (RTO) measured in minutes.

What can I learn about SAP HANA system replication in this guide?

Before configuring your system replication landscape, it's important to learn about basic concepts.

You will have to take decisions related to the used operation and replication modes. The selected operation mode determines what types of data packages are sent to the secondary system. The operation mode determines also which technique (data retention or log retention) is used to achieve a resync whenever system replication is out of sync. The network connection between the primary and the secondary systems is also important, because it impacts the overall performance of the systems involved in a system replication landscape. Finally, in system replication landscapes you can run also other systems (for example, DEV, QA systems, or even productive systems) on the secondary system's hardware while the primary system is in production. For information on these aspects, see *SAP HANA System Replication: Basic Concepts*.

After checking all the necessary prerequisites for configuring system replication, you can use the SAP HANA cockpit, the SAP HANA studio, or the `hdbnsutil` command line tool to configure system replication. At this point, it's also useful to know how to initialize the secondary system, how to change the chosen operation or replication mode, or how to add or remove hosts. You can enable the full sync option in a synchronous system replication to reach a true RPO=0. For information on these aspects, see *SAP HANA System Replication: Configuration*. This chapter also provides an overview of the configuration parameters, a command line reference, as well as information about SAP HANA tenant database systems in a system replication configuration, and the system replication setup for XS advanced.

Learn next how to perform a takeover and a failback for planned or unplanned downtimes of the primary system. You can perform a standard takeover, a takeover with handshake, or an invisible takeover. For information, see *SAP HANA System Replication: Takeover and Failback*.

To quickly access again data, which was deleted in the original system, you can prepare your system replication landscape for secondary time travel. Secondary time travel allows you to start the secondary system or the log replay at a previous point in time. For information, see *SAP HANA System Replication: Secondary Time Travel*.

To support read access on the secondary system, you can configure an Active/Active (read enabled) system replication. Active/Active (read enabled) reduces the load on the primary system, but does not double the

capacity; it simply extends read capabilities. In an Active/Active (read enabled) system replication configuration, 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. For more information about Active/Active (read enabled), see *SAP HANA System Replication with Active/Active (Read Enabled)*.

Besides the standard setup, in which a primary system ships all the data to the secondary system, you can also configure a multitier or a multitarget system replication. In a multititer system replication, a tier 2 system replication setup can be used as the source for adding further tiers in a chain. In a multitarget system replication, the primary system can replicate data changes to more than one secondary system. To learn how to configure the different setups and how to handle in different disaster recovery scenarios, see *SAP HANA System Replication Setups*.

There are multiple ways to monitor system replication and to verify if the primary and secondary systems are in sync and are running correctly:

- Alerts on the primary and secondary systems warn you of potential problems.
- To ensure rapid takeover in the event of planned or unplanned downtime, you can check the status of the replication between the primary and the secondary systems.
- You can monitor system replication using the SAP HANA cockpit, the SAP HANA studio, the `hdbnsutil` command line tool, or SQL queries.

For more information, see *SAP HANA System Replication: Operation and Maintenance*. This chapter also includes information about the network connection, how to copy or move tenants in a system replication configuration, or how to use SAP HANA system replication to update your SAP HANA systems.

Troubleshoot System Replication describes how to analyze, avoid, and solve problems related to system replication.

Communication between systems in a system replication scenario is always authenticated. In addition, it is possible to secure internal network communication between primary and secondary systems using TLS/SSL. For more information, see *Security Aspects for SAP HANA System Replication*.

Finally, use the *SQL and System View Reference* chapter in this guide to have a look at the system views relevant for system replication (for example, M_SERVICE_REPLICATION, M_SYSTEM_REPLICATION).

Related Information

SAP HANA System Replication: Basic Concepts [page 8]
SAP HANA System Replication: Configuration [page 35]
SAP HANA System Replication: Takeover and Failback [page 89]
Secondary Time Travel [page 113]
SAP HANA System Replication with Active/Active (Read Enabled) [page 120]
SAP HANA System Replication Setups [page 134]
SAP HANA System Replication: Operation and Maintenance [page 154]
Troubleshoot System Replication [page 216]
Security Aspects for SAP HANA System Replication [page 239]
SQL and System View Reference [page 257]

2 SAP HANA System Replication: Basic Concepts

Before configuring your system replication landscape, learn about basic concepts and possible setups.

What do I need to know before configuring system replication?

Learn which operation and replication modes are supported. The selected operation mode determines what types of data packages are sent to the secondary system. The operation mode determines also which technique (data retention or log retention) is used to achieve a resync whenever system replication is out of sync.

The network connection between the primary and the secondary systems is also important, because it impacts the overall performance of the systems involved in a system replication landscape. While the network throughput requirements of the communication channel used in SAP HANA system replication are influenced by the used operation modes, the selected replication modes impact the network latency requirements.

Finally, in system replication landscapes you can run also other systems (for example, DEV, QA systems, or even productive systems) on the secondary system's hardware while the primary system is in production.

Where can I find more information?

The following SAP Notes are relevant for a full understanding of the basic concepts described in this chapter:

SAP Notes

SAP Note	Title
1999880	FAQ: SAP HANA System Replication
1969700	SQL Statement Collection for SAP HANA
1681092	Multiple SAP HANA DBMSs (SIDs) on one SAP HANA system
2211663	The license changes in an SAP HANA database after the deregistration of the secondary site from a system replication setting
2447994	SAP HANA Dynamic Tiering Support for SAP HANA System Replication

Related Information

Introduction to System Replication [page 9]
Replication Modes for SAP HANA System Replication [page 12]
Operation Modes for SAP HANA System Replication [page 14]
Data Transferred to the Secondary System [page 16]
Resync Optimization [page 19]
Network Recommendations [page 27]
Secondary System Usage [page 32]

2.1 Introduction to System Replication

SAP HANA system replication is a mechanism for ensuring the high availability of your SAP HANA system.

What is system replication?

System replication is SAP's recommended configuration for addressing SAP HANA outage reduction due to planned maintenance, faults, and disasters. It supports a recovery point objective (RPO) of 0 seconds and a recovery time objective (RTO) measured in minutes.

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. Furthermore, it requires a reliable link between the primary and secondary systems.

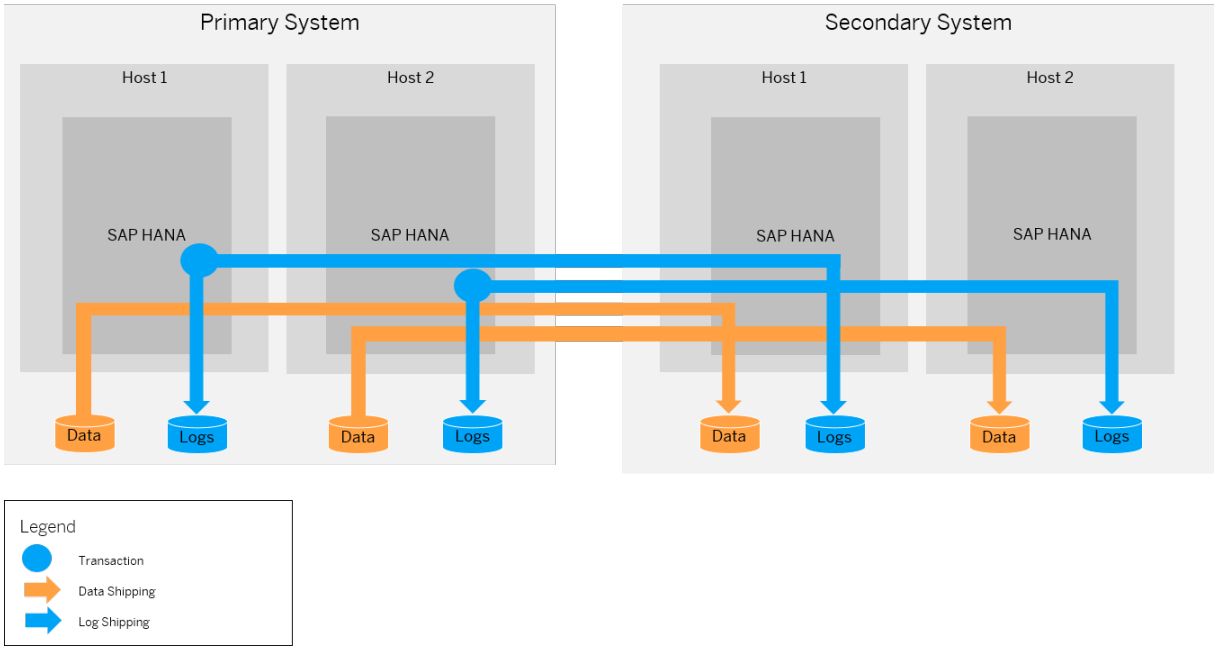
Each service of the primary system communicates pairwise with a counterpart in the secondary system. The main difference to the primary system is that the secondary system does not accept requests or queries. The secondary system can accept queries only in an Active/Active (read enabled) configuration. For more information, see *SAP HANA System Replication with Active/Active (Read Enabled)*.

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. Alternatively or additionally, a secondary system can be installed in a remote data center for disaster recovery. The instances in the secondary system operate in live replication mode. In this mode all secondary system services constantly communicate with their primary counterparts, replicate and persist data and logs, and typically load data to memory. The log and data can be compressed before shipping. For more information, see *Data and Log Compression*.

How does system replication work?

Once SAP HANA system replication is enabled, each server process on the secondary system establishes a connection with its primary system counterpart and requests a snapshot of the data. From then on, all logged changes in the primary system are replicated continuously. Whenever logs are persisted (meaning they are written to the log volumes of each service) in the primary system, they are also sent to the secondary system.

The following graphic illustrates the general system replication processes:



> **i 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*.

A transaction in the primary system is not committed before the redo logs are replicated. This is determined by the selected replication mode when setting up system replication. For a detailed description of each replication mode, see *Replication Modes for SAP HANA System Replication*.

If the connection to the secondary system is lost or if the secondary system crashes, the primary system (after a brief, configurable timeout) resumes operations. The way the received logs on the secondary system are handled depends on the selected operation mode. For a detailed description of each operation mode, see *Operation Modes for SAP HANA System Replication*.

While system replication is running, the secondary system configured identically to the primray system is on standy until a takeover takes place.

In the event of a failure that justifies a full system takeover, you switch the secondary system from the live replication mode to a full operation mode. The secondary system, which already preloaded the same column data as the primary system and possibly is already read enabled, becomes the primary system by replaying the last transaction logs and then starts to accept queries.When the original system can be restored to service, it can be configured as the new secondary system or reverted to the original configuration.

Which other setups are possible?

Besides the above presented standard setup, in which a primary system ships all the data to the secondary system, you can also configure a multitier or a multitarget system replication.

In a multititer 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. The primary system is always on tier 1. The replication source for the tier 2 secondary system is the primary system, while the replication source for the tier 3 secondary system is the tier 2 secondary. For more information, see *SAP HANA Multitier System Replication*.

In a multitarget system replication, the primary system can replicate data changes to more than one secondary system. For more information, see *SAP HANA Multitarget System Replication*.

What about license validity?

The primary system automatically replicates relevant license information to the secondary system. No additional license needs to be installed, since the primary and secondary system have the same SID. For more information about licensing in SAP HANA system replication, see *SAP Note 2211663*.

When using an Active/Active (read enabled) system replication configuration, the secondary system is subject to licensing. For more information, see *SAP HANA Feature Scope Description*.

What other system requirements apply?

In general, the replicating systems must be identical, but full details of prerequisites which apply and things you need to know before you start are given in the topic *General Prerequisites for Configuring SAP HANA System Replication*.

Related Information

SAP HANA System Replication with Active/Active (Read Enabled) [page 120]
Replication Modes for SAP HANA System Replication [page 12]
Operation Modes for SAP HANA System Replication [page 14]
Data and Log Compression [page 30]
SAP HANA Multitier System Replication [page 134]
SAP HANA Multitarget System Replication [page 148]
General Prerequisites for Configuring SAP HANA System Replication [page 36]
SAP Note 2211663

2.2 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`. **i 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.

i 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 62]
Changing the Replication Mode [page 65]

2.3 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 16]
Active/Active (Read Enabled)
SAP HANA System Replication Command Line Reference [page 75]
LogReplay: Managing the Size of the Log File [page 234]
Network Recommendations [page 27]
Changing the Operation Mode [page 64]

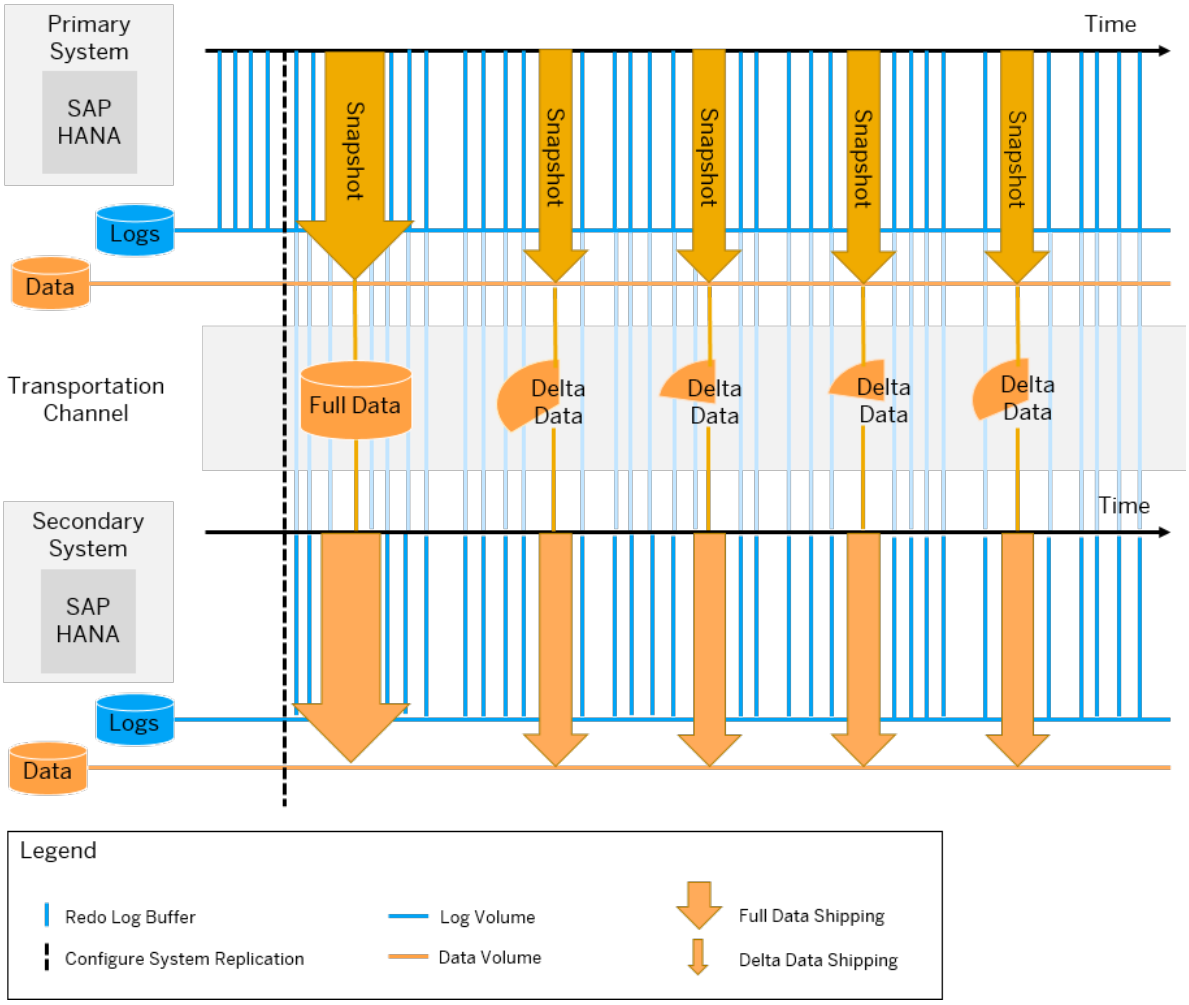
2.4 Data Transferred to the Secondary System

The selected operation mode determines what types of data packages are sent to the secondary system.

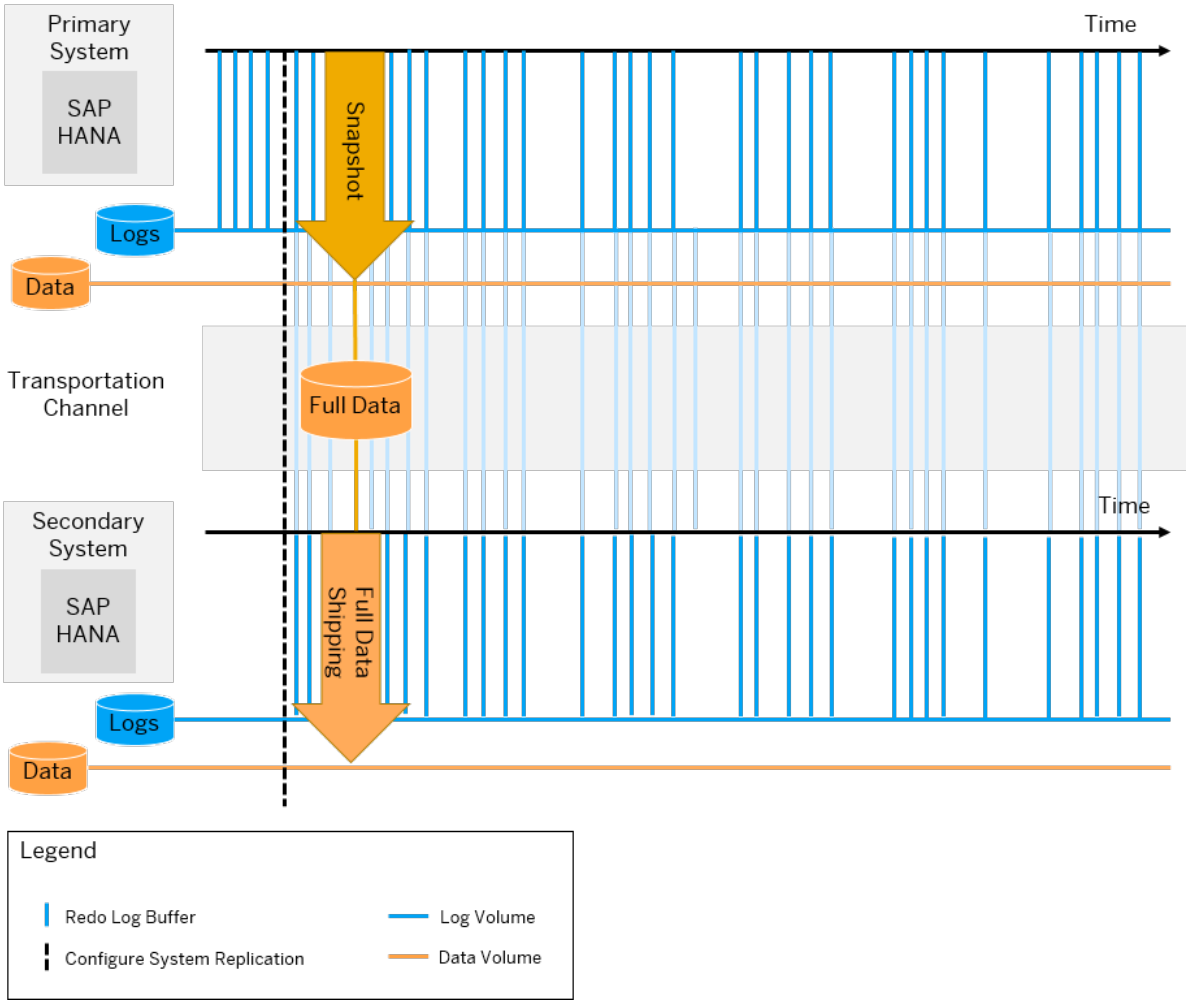
When system replication is configured, the following types of data packages can be sent to the secondary system:

- 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:



> **i 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 65]
Troubleshoot System Replication [page 216]
System Replication Network Connection [page 194]

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. If none of the three savepoints or snapshots are compatible, then a full data shipping will automatically be carried out.

> **i** Note
>
> If system replication is out of sync and you need to register again the initial secondary system, use the command `hdbnsutil -sr_register`. It is not needed to unregister the secondary system before registering it again. Unregistering the initial secondary system before registering it would hinder an optimized resync and would trigger a full data shipping.

Depending on the chosen operation mode, two different techniques are in place to achieve a resync: data retention and log retention. For more information, see *Data Retention* or *Log Retention*

Related Information

Data Retention [page 20]
Log Retention [page 21]

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 [page 89]
SAP HANA System Replication Configuration Parameters [page 65]

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 21]
Log Retention for Failback [page 22]
Estimating the Maximum Retention Time [page 23]
Log Retention and Multitarget System Replication [page 25]
SAP HANA System Replication Configuration Parameters [page 65]

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.

> **i 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.

> **i 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 [page 23]

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 primray 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`

  > i 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.

2.5.2.3 Estimating the Maximum Retention Time

How long will your SAP HANA system replication landscape survive in a disconnect scenario before running into a log full situation, when `logreplay` or `logreplay_readaccess` is configured?

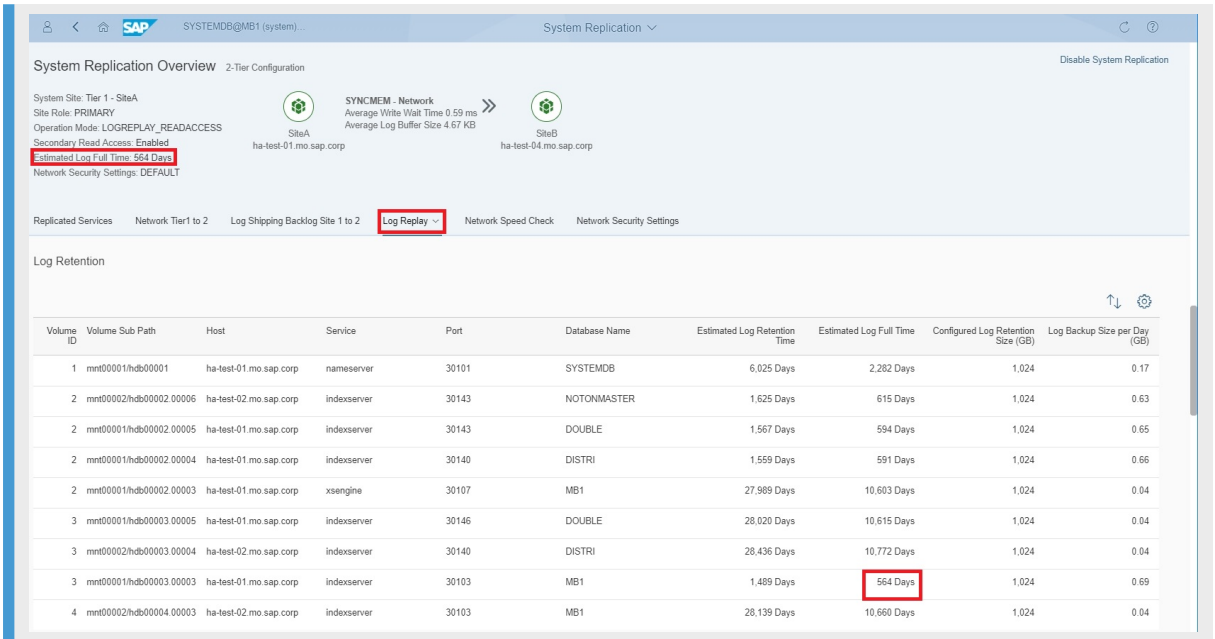
You can use the SAP HANA cockpit to get an estimation of the maximum retention time.

After opening the system replication tile from the system database overview page, the *System Replication Overview* section provides information either about the *Estimated Log Full Time* or the *Estimated Log Retention Time*.

The *Estimated Log Full Time* value shows you how much time is left before the primary system runs into a log full situation.

The *Estimated Log Retention Time* value shows you how much time is left before the primary system starts to overwrite the `RetainedFree` marked log segments making a full data shipping necessary for resync.

> ✤ Example
>
> The screenshot below shows you where to find the relevant values on the *System Replication Overview*:



The smallest calculated value from the *Estimated Log Retention Time* and *Estimated Log Full Time* columns will be displayed. In the screenshot above, you see an estimated log full time of 564 days. This means that the indexserver of the MB1 tenant will run into a log full in 564 days.

For more information, open the *LOG REPLAY* tab and look at the *Estimated Log Full Time* and *Estimated Log Retention Time* columns. In this services list, one row is shown for each service that is relevant for replication, because it has its own persistence consisting of data and log volume. The calculation is based on the value of the `logshipping_max_retention_size` parameter, the write load on the primary system (taken from the log backup size), and the available disk space. The columns provide the log retention details only for the log volumes.

For a full description of each column, see the table below:

Column Name	Description
Volume ID	The volume ID of the service
Volume Sub Path	The last two directories of the file system path where the log volume of this service is located
Host	The host where the service is running
Service	The relevant service for system replication (for example, indexserver, nameserver, xsengine). Only services that have their own persistence are relevant for replication.
Port	The port number
Database Name	The database name
Estimated Log Retention Time:	Maximum number of days or hours on which the logs can be retained to resync the primary and the secondary after the replication was interrupted without requiring a full data shipping

Column Name	Description
Estimated Log Full Time	Maximum number of days or hours until the log volume runs full when created redo logs can no longer be reused
Configured Log Retention Size (GB)	Determined by the value of the `logshipping_max_retention_size` parameter that is valid for this service. For example, the setting of this parameter in the indexserver.ini on the host on which the service is running overwrites the setting of this parameter in global.ini.
Log Backup Size per Day (GB)	The maximum log backup size per day indicating which replication relevant write load occurred for this service.
Disk ID	Internal Device ID (optional information)
Free Disk Space (GB)	Optional information

2.5.2.4 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.

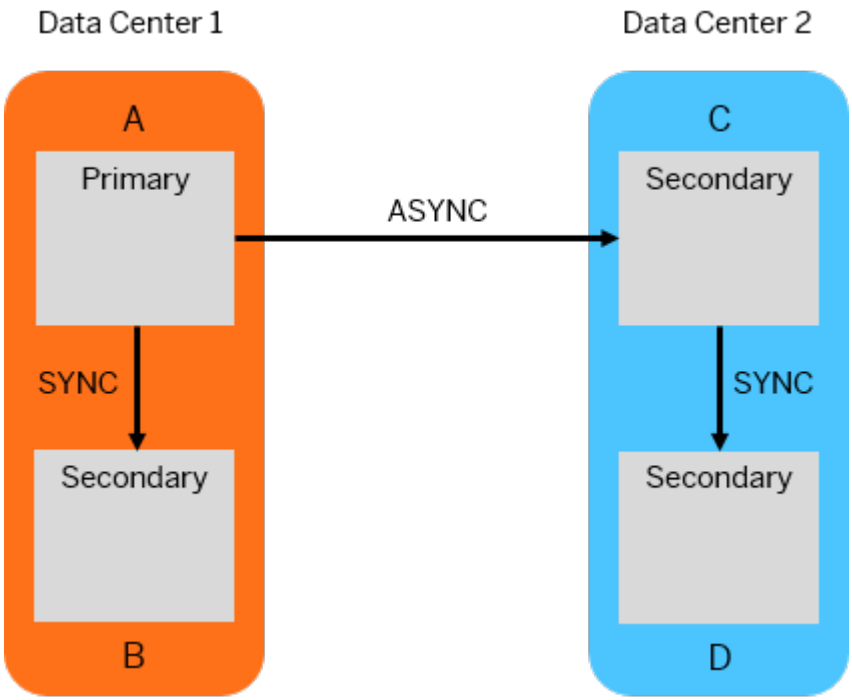
This can be done by setting the following parameter in global.ini: `[system_replication]/propagate_log_retention = on`. If you want to propagate log retention in a system replication landscape between all systems, this parameter should be set on all systems in the landscape.

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 148]
Disaster Recovery Scenarios for Multitarget System Replication [page 151]

2.6 Network Recommendations

The network connectionn between the primary and the secondary system impacts the overall performance of the SAP HANA systems involved in an SAP HANA system replication landscape.

The network throughput requirements of the communication channel used in SAP HANA system replication are influenced by the used operation modes. For more information, see *Network Throughput*.

The network latency requirements are influenced by the used replication modes. For example, for a synchronous replication mode the performance of the primary system is influenced by the time it takes until the acknowledgement from the secondary system arrives. For more information, see *Distance Between Data Centers*.

For more information about monitoring and configuring the network connection between the primary and the secondary systems for SAP HANA system replication, see *System Replication Network Connection*.

Related Information

Distance Between Data Centers [page 28]
Network Throughput [page 28]
Data and Log Compression [page 30]
Data and Log Volume Encryption [page 31]
System Replication Network Connection [page 194]

2.6.1 Distance Between Data Centers

The network latency is influenced by the selected replication mode.

If the distance between your sites is less than 100 km, use the synchronous replication modes: SYNC or SYNCMEM.

If the data centers are more than 100 km apart, the asynchronous replication mode ASYNC is recommended. For more information, see the *Check Network Configuration (Long Distance)* section in *Replication Performance Problems*.

Related Information

Replication Performance Problems [page 223]
Replication Modes for SAP HANA System Replication [page 12]
Monitoring the Network Latency [page 200]

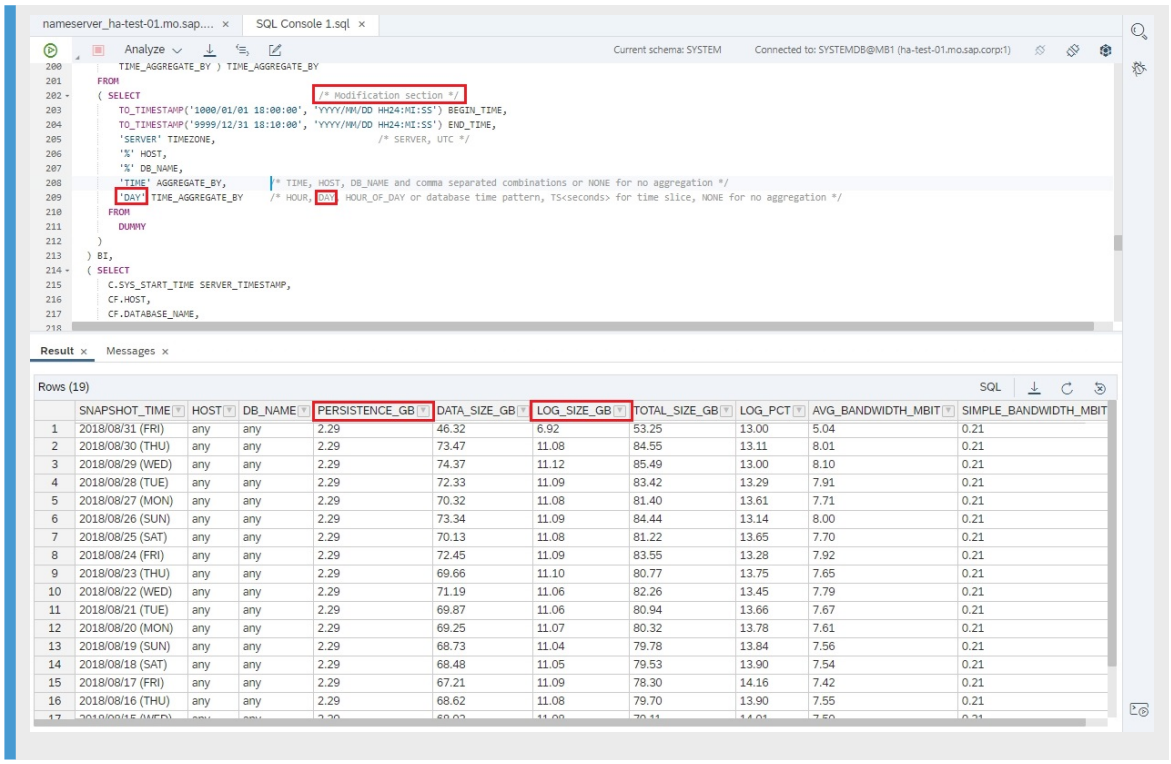
2.6.2 Network Throughput

The requirements regarding the network throughput are influenced by the selected operation mode.

The selected operation mode influences the size of the shipped data over the network. To estimate the required throughput, you need to know the size of the data and log that are generated during your daily workload. You can gain this information using one of the SQL statements contained in the zip file attached to the *SAP Note 1969700* as follows:

1. Open the *HANA_Replication_SystemReplication_Bandwidth.txt* from the SQL Statements.zip file.
2. Copy the statement to the SQL Console.
3. Change the *Modification section* and execute.

> ### Example
>
> You can change this section so that the results are displayed per day.



The displayed results provide the following information:

Tab Name	Description
SNAPSHOT_TIME	Time slot for which the results are valid
HOST	Host name
PERSISTENCE_GB	(Current) persistence data size (GB)
DATA_SIZE_GB	Total size of data written to disk (GB)
LOG_SIZE_GB	Total size of logs generated (GB)
TOTAL_SIZE_GB	Total amount of data and logs generated (GB)
LOG_PCT	Percentage of log compared to total size (%)
AVG_BANDWIDTH_MBIT	Average required network bandwidth to replication site (Mbit). It is only available for certain TIME_AGGREGATE_BY values.
SIMPLE_BANDWITH_MBIT	Simple network bandwidth calculation (Mbit) based on the formula that it should be possible to ship the persistence once per day.

As mentioned above, the requirements regarding the network throughput depend on the selected operation mode. *PERSISTENCE_GB* and *LOG_SIZE_GB* are the most important values in this context. The following overview distinguishes between the operation modes making use of these values:

Operation mode	Throughput requirements
delta_datashipping	It must be possible to transport the size of the persistently stored data within one day from the primary system to the secondary system. The size of the persistently stored data can be obtained from the above-mentioned *PERSISTENCE_GB* column. **Example** Given: 4,3 TB of persistently stored data Throughput: 4,3 TB/day <-> ~ 50 MByte/s => ~ 0,5 GBit/s connection required
logreplay & logreplay_readaccess	It must be possible to transport the size of the log backups of one day from the primary system to the secondary system within one day. With `logreplay` and `logreplay_readaccess` a pure log shipping is done, since no delta data shippings are necessary. The network throughput requirements depend mainly on the transactional workload on your primary system. The *LOG_SIZE_GB* column indicates the log size that was created per day. During a normal system replication operation, your network channel must be capable of handling this amount of data per day.

For more information about operation modes, see *Operation Modes* for SAP HANA System Replication.

Related Information

SAP Note 1969700
Operation Modes for SAP HANA System Replication [page 14]

2.6.3 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

2.6.4 Data and Log Volume Encryption

Depending on the version of your SAP HANA database, you must enable data volume encryption differently.

Before SAP HANA 2.0 SPS02, data volume encryption was enabled by setting the ini parameter `global.ini/persistence/data_encryption=true`.

Depending on the configuration of the inifile checker, this change was either automatically replicated to the secondary (for the `setting globel.ini/[inifile_checker]=replicate`) or you had to set it manually there as well. For more information, see *Monitoring and Replicationg INI File Parameter Changes*.

As of SAP HANA 2.0 SPS02, you must change the root keys used for data volume encryption and log volume encryption only in the primary system. The new keys will be propagated to all secondary systems. Enable or disable the data and log volume encryption in the primary system only. The setting will be propagated to all secondary systems as well. For details on enabling data and log volume encryption, see *Enable Data and Log Volume Encryption in an Existing SAP HANA Database*.

Related Information

Monitoring and Replicating INI File Parameter Changes [page 165]
Enable Encryption

2.7 Secondary System Usage

In system replication landscapes, you can run other systems (for example, DEV, QA systems, or even productive systems) on the secondary system's hardware while the primary system is in production.

Recommendations

When running other systems on the secondary system's hardware, keep in mind the following recommendations:

- We recommend using a separate storage infrastructure for each system. Since the secondary system requires the same I/O capacity as the primary, the additional systems could have a negative impact on the secondary's I/O.
- The SIDs and instance numbers used for the additional systems running on the secondary hardware must be different from the system replication SID.
- The <instance number>+1 of the productive system must not be used and must be free on both sites, because this port range is used for the system replication communication.
- To save memory resources, switch off the preload of tables on the secondary system using `global.ini/[system_replication]-> preload_column_tables=false`.
  The takeover process will take longer as no data is preloaded to memory on the secondary system.
- The available memory on the secondary system's hardware must be shared among the systems running there. However, the secondary system should receive the amount of memory it needs. Only the remaining memory can be shared among the DEV or QA systems.
  To allocate memory for every system running on the secondary's hardware, set the global allocation limit for each system using `global.ini/[memorymanager]-> global_allocation_limit`.
- If there are not enough resources to handle the load of all systems, the additional systems running on the secondary need to be shut down in case of a takeover.
- You can change the global.ini on the secondary accordingly and then activate the change with `hdbnsutil -reconfig` because no SQL is possible in this state.

Resources required on the secondary system

When planning to run other systems on the secondary system, you need to consider the available hardware resources, the table preload option (`preload_column_tables`), and the memory (`global_allocation_limit`) needed by the secondary system.

The tables below describe these requirements for each combination of table preload and operation mode:

Operation Mode: delta_datashipping

Preload	Memory needed for the secondary system (global_allocation_limit)
On	Set the `global_allocation_limit` to the same value as the memory available on the primary system.
Off	minimum 64 GB or row store size + 20 GB (if this sum is higher) Determine the row store size with the following statement: `select host, round (sum(page_size*USED_BLOCK_COUNT)/1024/1024/1024,2) as "RowStore Size GB" from m_data_volume_page_statistics where page_sizeclass = '16k-RowStore' group by host;` If this limit is not set, the SAP HANA database on the secondary system uses as much memory as it can get and possibly takes it away from the DEV/QA systems, which could run into out-of-memory. **i Note** If the row store size of the primary system grows during operation, it might become necessary to increase the `global_allocation_limit` on the secondary system.

Operation Mode: logreplay

Preload	Memory needed for the secondary system (global_allocation_limit)
On	Set the `global_allocation_limit` to the same value as the memory available on the primary system.
Off	size of loaded column tables (in-memory) + row store size + 50 GB Determine the size of the loaded column tables (in-memory) with this SQL statement: `select host, round(sum(memory_size_in_total)/1024/1024/1024) size_GB from m_cs_tables group by host;` **i Note** If the row store size of the primary system grows during operation, it might become necessary to increase the `global_allocation_limit` on the secondary system.

Operation Mode: logreplay_readaccess

Preload	Memory needed for the secondary system (global_allocation_limit)
On	Set the `global_allocation_limit` to the same value as the memory available on the primary system. **i Note** Read access on the Active/Active (read enabled) secondary system requires additional CPU capacity.

Preload	Memory needed for the secondary system (global_allocation_limit)
Off	Set the `global_allocation_limit` to the same value as the memory available on the primary system. **i Note** This load option reduces the required memory size for the Active/Active (read enabled) secondary thanks to tables which are used read-only in the primary system. However, read access on the Active/Active (read enabled) secondary system requires additional CPU capacity.

All resources that are not needed by the secondary system can be used to run further systems on the secondary. These resources must be granted to them by explicitly setting the global allocation limit.

For more information on multiple SAP HANA systems running productively on one host, see *SAP Note 1681092 Multiple SAP HANA DBMSs (SIDs) on one SAP HANA system*. The note also provides information about the cross replication possibility of two systems with two hosts.

Related Information

SAP Note 1681092

3 SAP HANA System Replication: Configuration

After checking all the necessary prerequisites for configuring system replication, use the SAP HANA cockpit, the SAP HANA studio, or the `hdbnsutil` command line tool to configure system replication.

How can I configure and disable system replication?

After checking all the necessary prerequisites, you can use the SAP HANA cockpit, the SAP HANA studio, or the `hdbnsutil` command line tool to configure system replication. This section describes how to configure and disable system replication. Generally, you have to perform the following steps:

- Perform an initial data backup or a storage snapshot.
- Enable the primary system for system replication.
- Establish a connection between the secondary and the primary systems.
- Initiate a full data replication by configuring system replication on the secondary and starting it. Thereafter, incremental data replication (only in `delta_datashipping` operation mode) and continuous redo log replication (in all operation modes) start automatically.
- Disable system replication on the secondary system.
- Disable system replication on the primary system.

This chapter describes also how to initialize the secondary system, how to change the chosen operation or replication mode, or how to add and remove hosts. Furthermore, it shows you how to enable the full sync option in a synchronous system replication to reach a true RPO=0. This chapter also provides an overview of the configuration parameters, a command line reference, as well as information about SAP HANA tenant database systems in a system replication configuration, and the system replication setup for XS advanced.

Where can I find more information?

The following SAP Notes are relevant for a full understanding of the basic concepts described in this chapter:

SAP Notes

SAP Note	Title
2211663	The license changes in an SAP HANA database after the deregistration of the secondary site from a system replication setting
2369981	Required configuration steps for authentication with HANA System Replication

SAP Note	Title
2300936	Host Auto-Failover & System Replication Setup with SAP HANA extended application services, advanced model
2447994	SAP HANA Dynamic Tiering Support for SAP HANA System Replication
611361	Hostnames of SAP servers
1945676	Correct Usage of hdbnsutil - sr_unregister
2300936	Host Auto-Failover & System Replication Setup with SAP HANA extended application services, advanced model

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 36]
Configuring SAP HANA System Replication [page 40]
Initializing the Secondary [page 60]
Full Sync Option for SAP HANA System Replication [page 62]
Add and Remove Hosts in SAP HANA System Replication [page 63]
Changing the Operation Mode [page 64]
Changing the Replication Mode [page 65]
SAP HANA System Replication Configuration Parameters [page 65]
SAP HANA System Replication Command Line Reference [page 75]
Disabling SAP HANA System Replication [page 77]
SAP HANA System Replication Setup for XS Advanced Runtime [page 85]
SAP HANA System Replication with Tenant Databases [page 87]

3.1 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 the whole system, 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.
- 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.

  > **i 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.

> **i 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.

> **i 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.

> **i 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
SPS11 & SPS12	Linux Intel SLES 11	little-endian
	Linux Power SLES 11	big-endian
SAP HANA 2.0 SPS00	Linux Intel SLES 12	little-endian
	Linux Power SLES 12	

System replication is supported between Intel little-endian (SAP HANA SPS12 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.

> **i 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 local userstore 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
Rename an SAP HANA System Host
Secure Internal Communication Between Sites in System Replication Scenarios
SAP HANA Database Backup and Recovery
Copying and Moving Tenant Databases
Configure a User Under the SRTAKEOVER Key
Connect to a Database With SSO or SAP HANA Credentials
Operating System User <sid>adm
Checking the Status with landscapeHostConfiguration.py
SAP HANA System Replication
SAP Note 2211663
SAP Note 2369981
SAP Note 2300936
SAP Note 2447994
SAP Note 401162
SAP Note 2382421

3.2 Configuring SAP HANA System Replication

You can configure system replication using SAP HANA cockpit, SAP HANA studio, or hdbnsutil.

> **i Note**
>
> To configure system replication, the primary system must have been backed up at least once. For more information, see *Backup and Recovery*.

You can configure system replication using the following tools:

- SAP HANA cockpit
  For more information, see *Configure System Replication with the SAP HANA Cockpit*.
- hdbnsutil
  For more information, see *Configure System Replication with hdbnsutil*.
- SAP HANA studio
  For more information, see *Configure System Replication with the SAP HANA Studio*.

> **i Note**
>
> To configure SAP HANA system replication for tenant databases, all configuration steps must be done on the system database. However, the data backups must be created for the system database as well as for all tenant databases. After a new tenant database was created in an SAP HANA tenant database system running with SAP HANA system replication, a backup of this new tenant database is necessary to automatically start its replication. Otherwise the replication for this new tenant database will not start.

Related Information

SAP HANA Database Backup and Recovery
Configure System Replication with the SAP HANA Cockpit [page 41]
Configure SAP HANA System Replication with hdbnsutil [page 54]
Configure SAP HANA System Replication with the SAP HANA Studio [page 58]

3.2.1 Configure System Replication with the SAP HANA Cockpit

Use the SAP HANA cockpit to configure system replication between two SAP HANA systems.

Prerequisites

- You have considered all the general prerequisites needed to set up system replication. For more information, see *General Prerequisites for Setting Up SAP HANA System Replication*.
- You have added both systems in the SAP HANA cockpit.
- You need the operating system user to set up a system replication landscape with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* and *Connect to a Resource using Database Credentials*.

Context

Use the *System Replication* tile on the system overview page to configure system replication. Once the configuration is done, the tile displays information on the operation mode, the replication mode, the configuration type, and the status of system replication. For more information, see *System Replication Tile*.

To configure system replication in the SAP HANA cockpit, first enable system replication on the primary system and then register the secondary system.

You can configure system replication in two ways in the SAP HANA cockpit:

- From the primary system. For more information, see *Configure System Replication from the Primary System*.
  You can use this method for tier 2 and tier 3 setups. Even though this method is easier, it doesn't work to configure multitarget or more than tier 3 setups directly from the primary system.
- From the primary and secondary systems. For more information, see *Configure System Replication from the Primary and the Secondary Systems*.
  You can use this method to configure any system replication setups you want.

To configure system replication in the SAP HANA cockpit, first enable system replication on the primary system and then register the secondary system.

To change the configuration of an existing system replication, you can register again a previously stopped secondary system when a full data shipping is needed or when you want to change the operation mode. For more information, see *Reinitialize the Secondary System*.

For an example of a system replication configuration, see *Example: Configure SAP HANA System Replication with the SAP HANA Cockpit*.

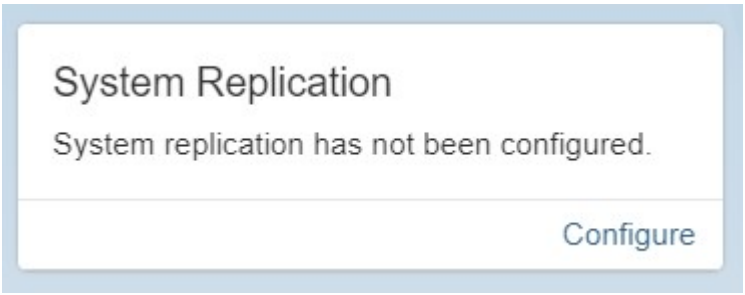
Related Information

System Replication Tile [page 42]
Configure SAP HANA System Replication from the Primary System [page 45]
Configure SAP HANA System Replication from the Primary and the Secondary Systems [page 47]
Reinitialize the Secondary System [page 49]
Example: Configure SAP HANA System Replication with the SAP HANA Cockpit [page 50]

3.2.1.1 System Replication Tile

The *System Replication* tile on the system overview page displays information about the operation mode, the replication mode, the configuration type, and the status of system replication.

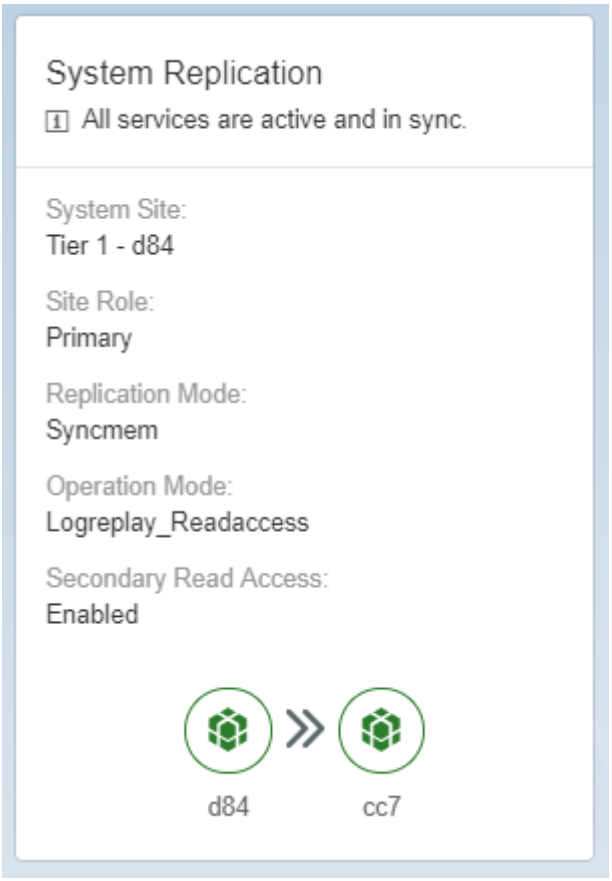
For example, you can see on the *System Replication* tile when system replication is not configured:



After configuring system replication, the tile looks differently for the primary and the secondary system.

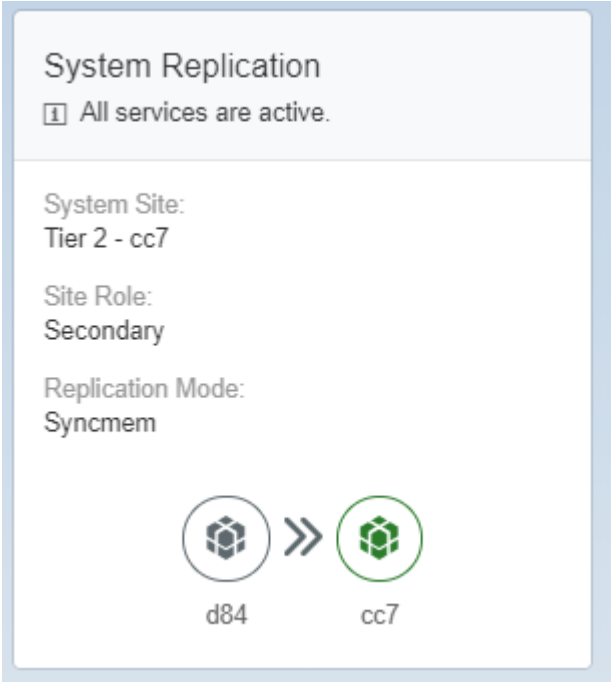
On the primary system's tile you can see the role of the system, the replication mode, the operation mode, and whether read access is enabled for Active/Active (read enabled) configurations.

This is an example of a system replication tile for the primary system:

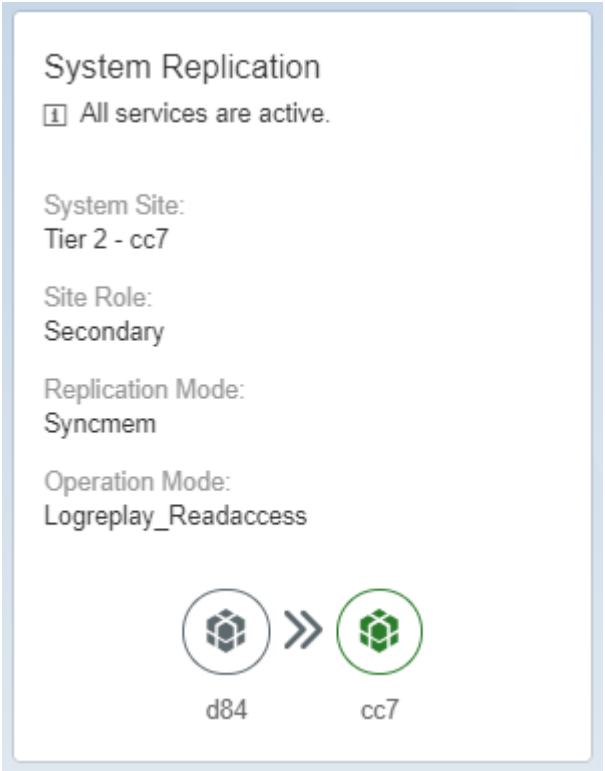


On the secondary system's tile you can see the role of the system and the replication mode.

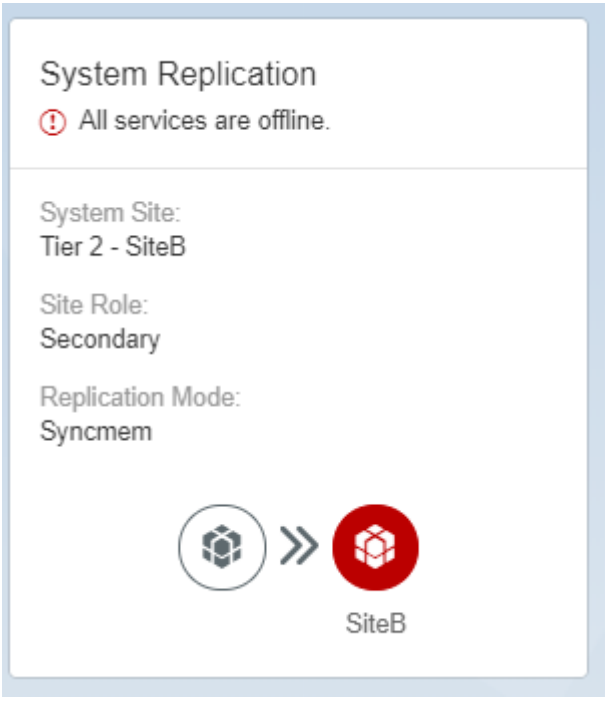
This an example of a system replication tile for the secondary system:



In an Active/Active (read enabled) configuration, the secondary system's tile displays also the `logreplay_readaccess` operation mode:



The system replication tile of your secondary system indicates when the services are offline (for example, when you stop the secondary system to disable system replication):



3.2.1.2 Configure SAP HANA System Replication from the Primary System

To configure SAP HANA system replication, first enable system replication on the primary system and then register the secondary system. Use the SAP HANA cockpit to execute these separate steps in one configuration step from the primary system.

Prerequisites

You've navigated to the *Database Overview* page of the database you want to manage. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.

- You've considered all the general prerequisites needed to set up system replication. For more information, see *General Prerequisites for Setting Up SAP HANA System Replication*.

Context

This topic describes how to configure system replication from the primary system in SAP HANA cockpit in one configuration step. You can use this method for 2-tier and 3-tier setups.

> **i Note**
>
> 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.

Procedure

1. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB) of the future primary system, choose the *System Replication* card.

   If you never configured system replication before, this card displays the message *System replication is not yet enabled for this system*.

   The *System Replication* page opens. If you performed a data backup before enabling system replication, this page displays the last data backup on the top left and the *Configure System Replication* button on the top right.
2. Choose *Configure System Replication*.

   The *System Replication Configuration* dialog opens, allowing you to run the configuration in background.
3. Enter the logical name used to represent the primary system in the *Tier 1 System Details* screen area.
4. Enter the logical name used to represent the secondary system in the *Tier 2 System Details* screen area.

   Keep in mind that the secondary system must have the same SAP system ID (<SID>) and instance number as the primary system so that they're identified as secondaries.
5. Select the secondary system host and mark the checkbox below this area to stop the system.
6. Select a replication mode. For more information on the available replication modes, see *Replication Modes for SAP HANA System Replication* in the *SAP HANA Administration Guide*.
7. Select an operation mode. For more information on the available operation modes, see *Operation Modes for SAP HANA System Replication* in the *SAP HANA Administration Guide*.
8. Decide whether to initiate a full data shipping or not.
9. Check *Start Secondary after Registration*.
10. Optional: To add a third tier to your system replication landscape configuration, click *Add Tier 3 System* on the bottom left.
11. Choose *Configure*.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 36]
Replication Modes for SAP HANA System Replication
Operation Modes for SAP HANA System Replication
SAP HANA System Replication
SAP Note 2447994

3.2.1.3 Configure SAP HANA System Replication from the Primary and the Secondary Systems

To set up SAP HANA system replication, first enable system replication on the primary system and then register the secondary system. Use the SAP HANA cockpit to execute these configuration steps on the primary system and separately on the secondary system.

Prerequisites

You've navigated to the *Database Overview* page of the database you want to manage. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.

- You've considered all the general prerequisites needed to set up system replication. For more information, see *General Prerequisites for Setting Up SAP HANA System Replication*.

Context

This topic describes how to enable system replication on the primary system and then register the secondary system using the SAP HANA cockpit. You can use this method to configure any system replication setups you want.

> **i** Note
>
> 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.

Procedure

1. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB) of the future primary system, choose the *System Replication* card.

   If you never configured system replication before, this card displays the message *System replication is not yet enabled for this system*.

   The *System Replication* page opens. If you performed a data backup before enabling system replication, this page displays overview information on the primary system on the top left and the *Enable This System as Primary* link on the top right.
2. Enter the logical name used to represent the primary system and choose *Configure* on the bottom right.
3. On the *Database Overview* page of the future secondary system, click on the *Services* card.
4. Choose *Stop System* on the bottom right, because the system has to be offline in order to be registered as a secondary system.

Back on the *Database Overview* page of the future secondary system, the *Services* card displays the status *Stopped*.

5. On the *Database Overview* page of the secondary system, choose the *System Replication* card.

   The *System Replication* page opens, displaying overview information about the secondary system on the top left and the *Register Secondary System* button on the top right.

6. Choose *Register Secondary System*.

   The *System Replication Configuration* page opens.

7. On the *System Replication Configuration* page, enter the logical name used to represent the secondary system.
8. On the *System Replication Configuration* page, select a replication mode. For more information on the available replication modes, see *Replication Modes for SAP HANA System Replication* in the *SAP HANA Administration Guide*.
9. Select an operation mode. For more information on the available operation modes, see *Operation Modes for SAP HANA System Replication* in the *SAP HANA Administration Guide*.
10. Enter the host of the source system.

    > **i** Note
    >
    > If you're operating a distributed system on multiple hosts, enter the name of the host on which the master name server is running.

11. Check *Start Secondary after Registration*.
12. Review the configured information and choose *Configure* on the bottom right.

    The *System Replication Configuration* dialog opens. After the configuration is complete, the *System Replication Overview* page displays information on the configured systems.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 36]
Replication Modes for SAP HANA System Replication
Operation Modes for SAP HANA System Replication
SAP HANA System Replication
SAP Note 2447994

3.2.1.4 Reinitialize the Secondary System

You can register again a previously stopped secondary system using the SAP HANA cockpit.

Prerequisites

You've navigated to the *Database Overview* page of the database you want to manage. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.

Context

You can register again a previously stopped secondary system. Reinitialize when a full data shipping is needed or when you want to change the operation mode.

> **i** Note
>
> The *System Replication* card isn't available in the tenant database.

Procedure

1. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB) of the stopped secondary system, choose the *System Replication* card.
2. On the *System Replication Overview*, choose *Reinitialize Secondary System* on the top right.
3. On the *System Replication Configuration* page, you can now change the configuration.
   Change the operation mode or resync the persistencies using the *Initiate full data shipping* option.

   The secondary system is up and running again.

Related Information

SAP HANA System Replication

3.2.1.5 Example: Configure SAP HANA System Replication with the SAP HANA Cockpit

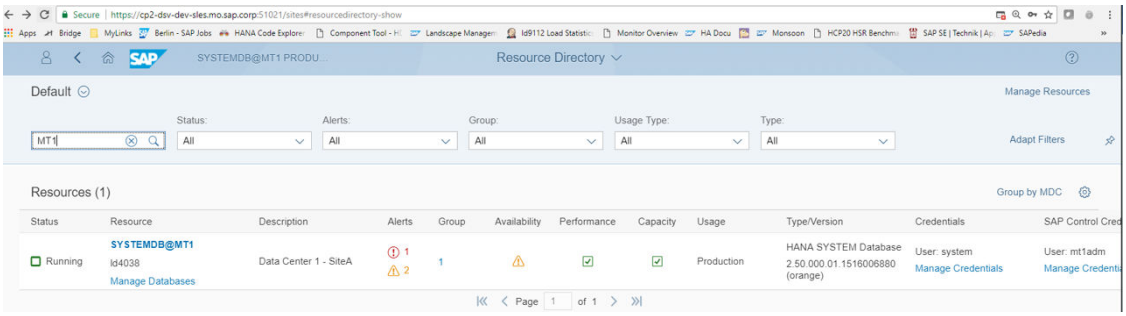
Learn how to configure system replication with the SAP HANA cockpit from the primary system.

Context

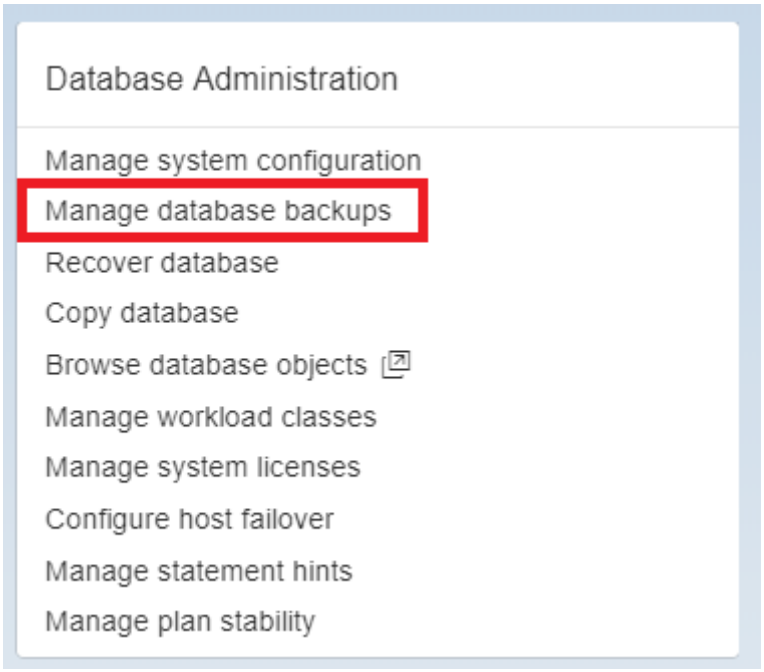
> **i Note**
>
> To configure system replication, the primary system must have been backed up at least once. This step is included in the procedure described below. Skip this step, if the system was backed up before. For more information, see *Backup and Recovery*.

Procedure

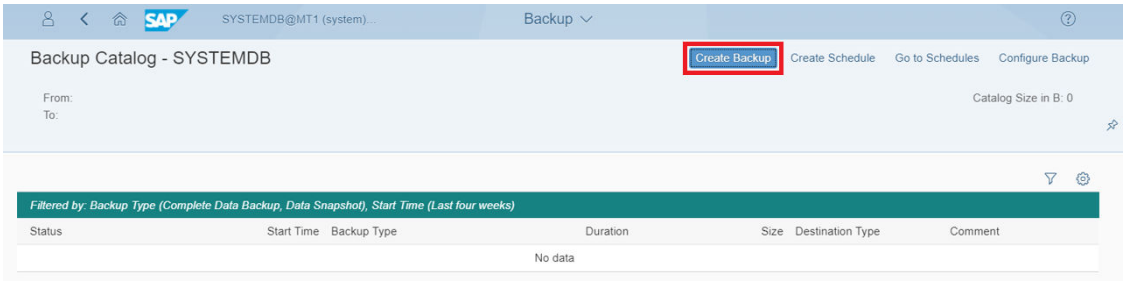
1. Create a full data backup for the system database as well as for all tenant databases.
   a. Choose the system database from the *Resource Directory*. This is the future primary system.



   b. Choose *Manage database backups* on the system overview page of the primary system.

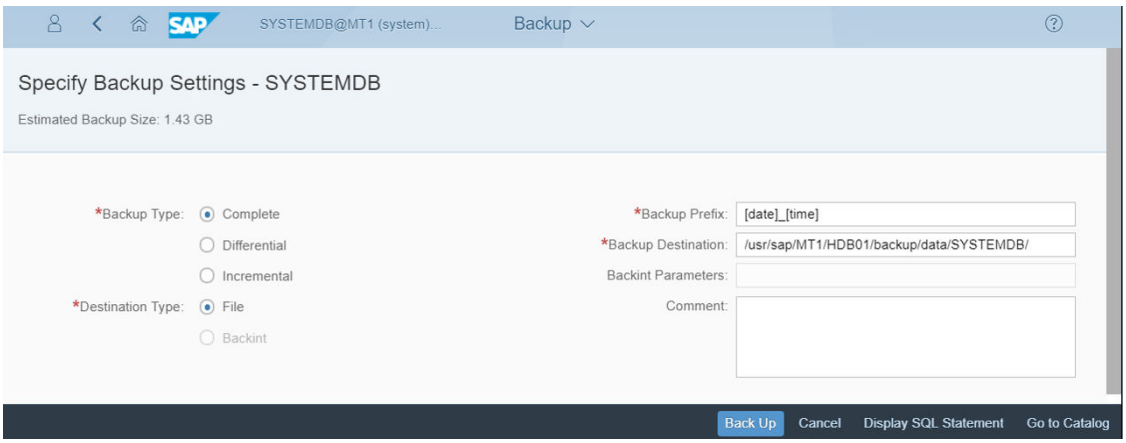


c. Choose *Create Backup* on the top right.

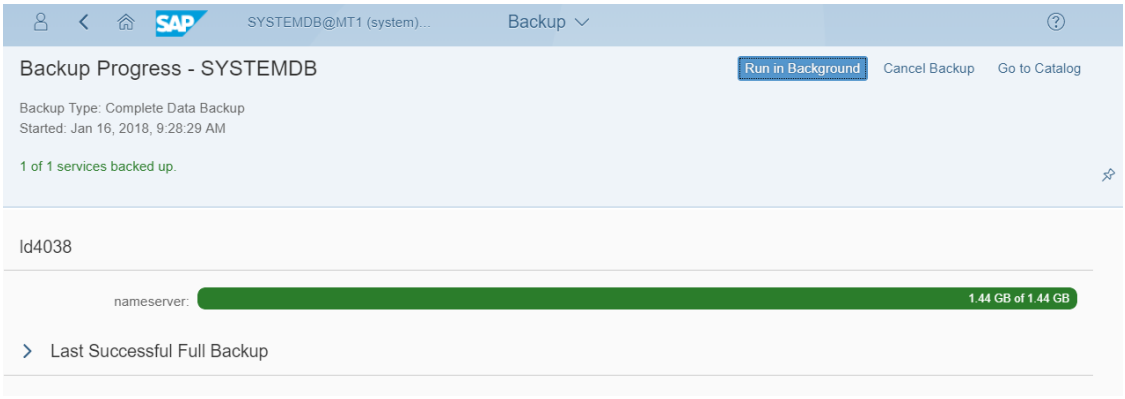


d. Create a complete backup, choose a destination type, enter a prefix and the backup destination, then finally choose *Back Up*.

For example, the backup settings can look like this:



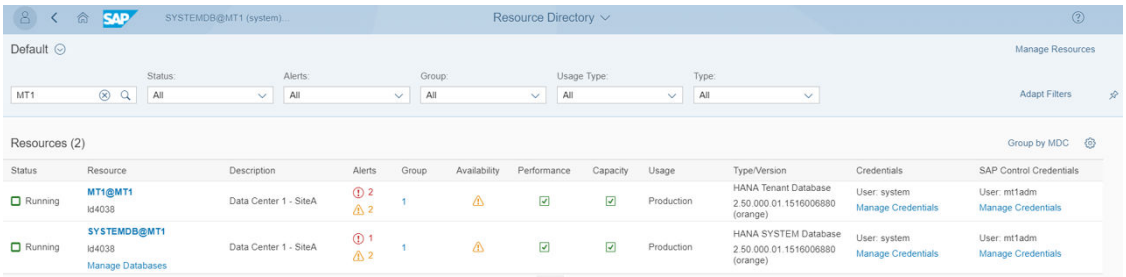
You can see the running backup progress for each service followed by the backup details once the backup is completed:



To see the data backups contained in the backup catalog, go back to the previous page using the arrow on top left.

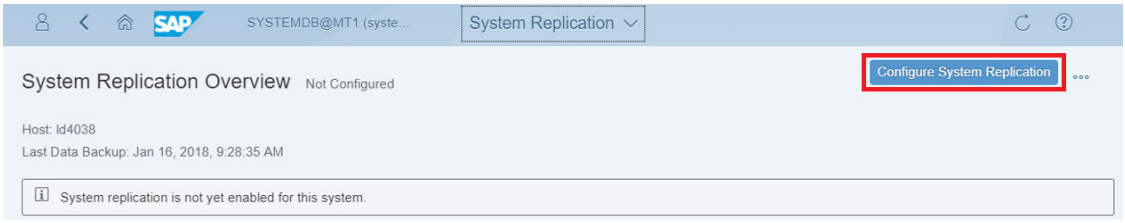
e. To back up also the tenant database, select it from the *Resource Directory* and proceed as described in the previous steps.

This is an example of the system database and the tenant database of the future primary in the resource directory:



> **i Note**
>
> Keep in mind that you 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:
>
> ```
> /usr/sap/<SID>/SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT
> /usr/sap/<SID>/SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY
> ```

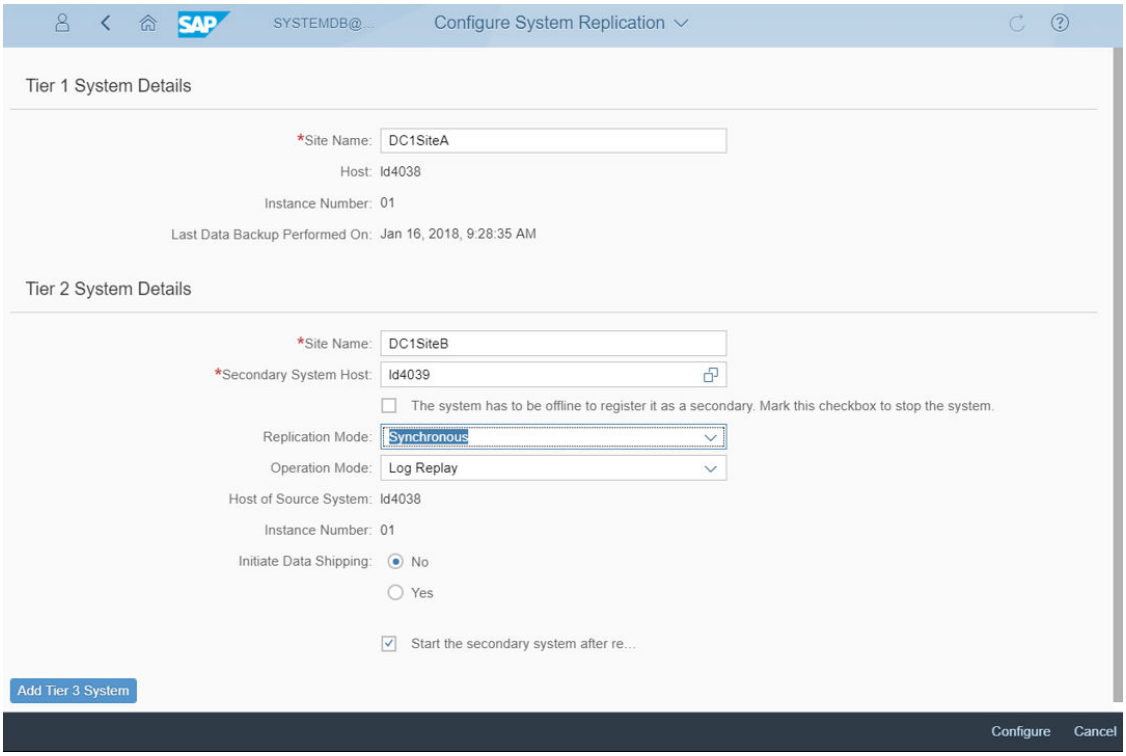
2. To configure system replication, proceed as follows:
   a. Choose the system database that is going to be your primary system from the *Resource Directory*.
   b. On the *System Overview* page of the system database, choose the *System Replication* tile to configure the primary system.
   c. Choose *Configure System Replication* on the top right to configure system replication without having to switch to the future secondary system.



d. Enter the required information on the *System Replication Configuration* page.

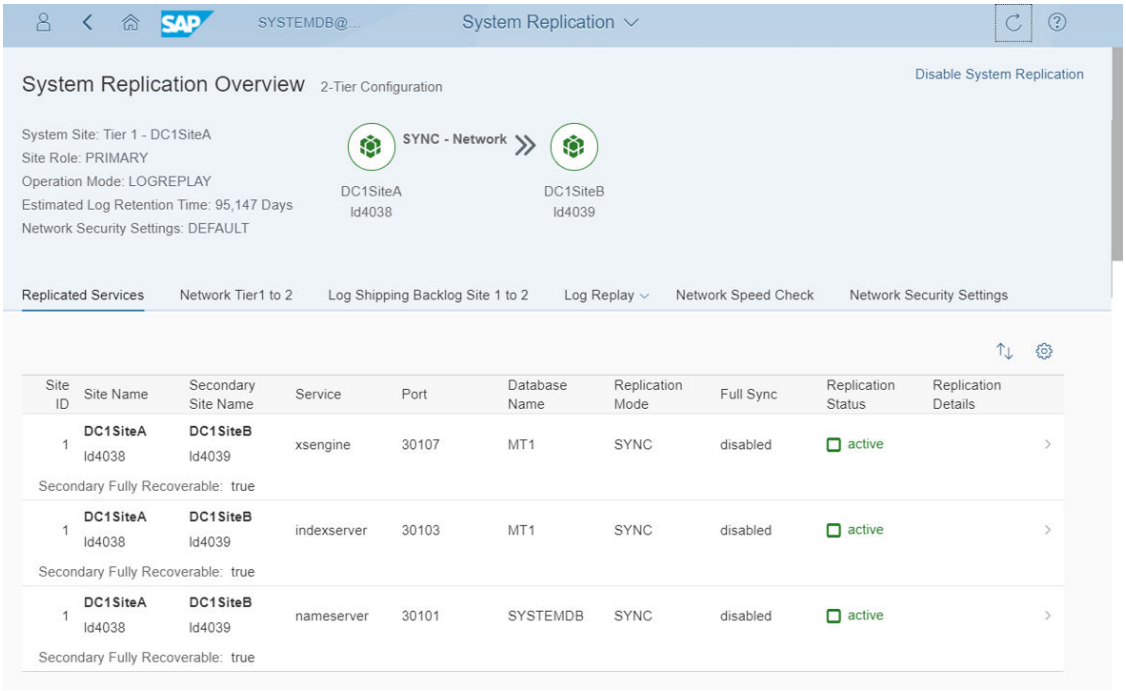
> **i Note**
>
> Keep in mind that the secondary system must have the same SAP system ID (<SID>) and instance number as the primary system so that they are identified as secondaries.

This is an example of possible configuration settings:



> **i Note**
>
> To configure a third tier, choose *Add Tier 3 System* and follow the instructions in *Example: Configure SAP HANA Multitier System Replication with the SAP HANA Cockpit*.

The overview shows now that the SAP HANA system replication configuration was successful:



Related Information

Example: Configure SAP HANA Multitier System Replication with SAP HANA Cockpit [page 139]
Backup and Recovery

3.2.2 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*.

> **i** 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
--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 36]
Rename an SAP HANA System Host
Host Name Resolution for System Replication
Creating Backups

3.2.2.1 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.

> **i** 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
SAP Note 611361

3.2.3 Configure SAP HANA System Replication with the SAP HANA Studio

To configure SAP HANA system replication between two identical SAP HANA systems, you must first enable system replication on the primary system and then register the secondary system.

Prerequisites

- You have considered all the general prerequisites needed to set up system replication. For more information, see *General Prerequisites for Setting Up SAP HANA System Replication*.
- You have added both systems in the SAP HANA studio.

Procedure

1. Enable system replication on the primary system, which has to be online, as follows:
   a. In the *Systems* view, right-click the primary system and choose *Configuration and Monitoring* > *Configure System Replication*.
   The *Configure System Replication* dialog opens. The *Enable System Replication* option is selected by default.

   > **i Note**
   >
   > You can also access the *Configure System Replication* dialog from the *Landscape* > *System Replication* tab.

b. Choose *Next*.
c. Enter the logical name used to represent the primary system and choose *Next*.
d. Review the configured information and choose *Finish*.
e. Stop the secondary with right-click on the secondary system and choosing *Configuration and Monitoring* > *Stop System*.

2. Register the secondary system as follows:
   a. Stop the secondary system if it is still running. Right-click the secondary system and choose *Configuration and Monitoring* > *Stop System*
   b. In the *Systems* view, right-click the 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 secondary system.

   > **i 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 log replication mode. For more information on the available replication modes, see *Replication Modes for SAP HANA System Replication*.
   f. Specify the operation mode. For more information, see *Operation Modes for SAP HANA System Replication*. The `logreplay_readaccess` operation mode is not available in SAP HANA studio.
   g. Review the configured information and choose *Finish*.
3. Optional: Configure the parameters in the `system_replication` section of the `global.ini` file.

   These parameters determine for example the size and frequency of data and log shipping requests. All parameters have a default configuration.
4. If necessary, start the secondary system.

   > **i Note**
   >
   > The secondary system is started automatically unless you deselected the corresponding option during configuration (step 2).

   The secondary system requests an initial full data replica from the primary system.

Results

You have enabled system replication and registered the secondary system with the primary system. The secondary system operates in recovery mode. All secondary system services constantly communicate with their primary counterparts, replicate and persist data and logs, and load data to memory. However, the secondary system does not accept SQL connections.

In the *Systems* view, the primary and secondary systems appear as operational ( ). If the secondary system is not open for read access, it appears as operational ( ) but with an error ( ) indicating that no connection to the database is available. For more information, see *Generic Conditions for Active/Active (Read Enabled)*.

Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 36]
Replication Modes for SAP HANA System Replication [page 12]
Operation Modes for SAP HANA System Replication [page 14]
Add an SAP HANA System
Stop a System
SAP HANA System Replication Configuration Parameters [page 65]
Rename an SAP HANA System Host
Enable Encryption
Data and Log Volume Encryption
Encryption Key Management
Generic Conditions for Active/Active (Read Enabled) [page 123]
SAP Note 611361

3.3 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 61]
Resync Optimization [page 19]

3.3.1 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 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.

3.4 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.

> **i 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
```

> **i** 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 75]

3.5 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

> **i** Note
>
> Hosts must be added equally to both primary and secondary sites.

> **i** Note
>
> Do not turn off system replication when adding a host.

> **→** Recommendation
>
> It is recommended that a host is added 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.

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

Add Hosts Using the Command-Line Interface
Remove Hosts Using the Command-Line Interface

3.6 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
```

> **i** 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

3.7 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 [page 257]

3.8 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.

Parameter	`datashipping_min_time_interval`
Type	Integer
Unit	seconds
Default	600 (10 min)

Parameter	datashipping_min_time_interval
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.

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. **i 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	datashipping_parallel_processing
Type	Boolean
Default	True
System	Secondary
Description	If activated, each data-shipping channel is assigned a separate job, therefore not only the network I/O but the processing steps are also done in parallel. This can be useful when the communication encryption and/or data compression are the limiting factor of the data-shipping performance. This would not be effective if the parameter `datashipping_parallel_channels` was set to 1 or 0.

Related Information

Full Sync Option for SAP HANA System Replication [page 62]
SAP HANA System Replication Alerts [page 161]
Operation Modes for SAP HANA System Replication [page 14]
Log Retention and Multitarget System Replication [page 25]
Change a System Property in SAP HANA Studio
Log Retention [page 21]

3.9 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>]
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. **i 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_changemode
Options	--mode=sync|syncmem|async
System	Secondary
Online/Offline	Online and offline
Description	Changes the replication mode of a secondary system.

Command	-sr_takeover
System	Secondary
Online/Offline	Online and offline
Description	Makes this secondary the new primary system.

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 62]
Example: Restore the Original SAP HANA Multitier System Replication Configuration [page 147]
SAP Note 1945676: Correct usage of hdbnsutil -sr_unregister
SAP Note 2980989: How-To: Performing a Full Data Shipment for a Single Volume / Service

3.10 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*.
- SAP HANA studio
  For more information, see *Disable SAP HANA System Replication with the SAP HANA Studio*.
- hdbnsutil
  For more information, see *Disable SAP HANA System Replication with hdbnsutil*.

Related Information

Disable SAP HANA System Replication in SAP HANA Cockpit [page 78]
Example: Disable SAP HANA System Replication with the SAP HANA Cockpit [page 79]
Disable SAP HANA System Replication with the SAP HANA Studio [page 85]
Disable SAP HANA System Replication with hdbnsutil [page 84]
SAP Note 1945676
SAP Note 2904125

3.10.1 Disable SAP HANA System Replication in SAP HANA Cockpit

You can disable SAP HANA system replication in an SAP HANA system using the SAP HANA cockpit.

Prerequisites

- The secondary system must be offline.
- You need the operating system user to disable system replication with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* in the *SAP HANA Administration Guide* and *Connect to a Database using Database Credentials*.

You've navigated to the *Database Overview* page of the database you want to manage. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.

Procedure

1. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB), use the *Services* card .
2. Unregister the secondary system as follows:
   a. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB) of the stopped secondary system, choose the *System Replication* card to stop the secondary system.

      The *System Replication* card opens displaying the *System Replication Overview*.
   b. Choose *Unregister this Secondary* on the top right.

      Depending whether your system should be online or offline after unregistering it, check the *Start system after unregistration* option in the confirmation dialog and choose *OK*. For more information, see *SAP Note 1945676*.
3. Disable system replication on the primary system as follows:
   a. In the *Monitoring* or *All* view, on the *Database Overview* page of the primary system, choose the *System Replication* card.

      The *System Replication* card opens displaying the *System Replication Overview*.
   b. Choose *Disable System Replication* on the top right and confirm that you want to disable system replication.

      The *Ignore the secondary system* option allows you to disable the primary system even though the secondary is still attached. This could be relevant, if the secondary has been uninstalled in the meantime.

Related Information

Connect to a Database With SSO or SAP HANA Credentials
Operating System User <sid>adm
SAP HANA System Replication
SAP Note 1945676

3.10.1.1 Example: Disable SAP HANA System Replication with the SAP HANA Cockpit

Learn how to disable system replication with the SAP HANA cockpit.

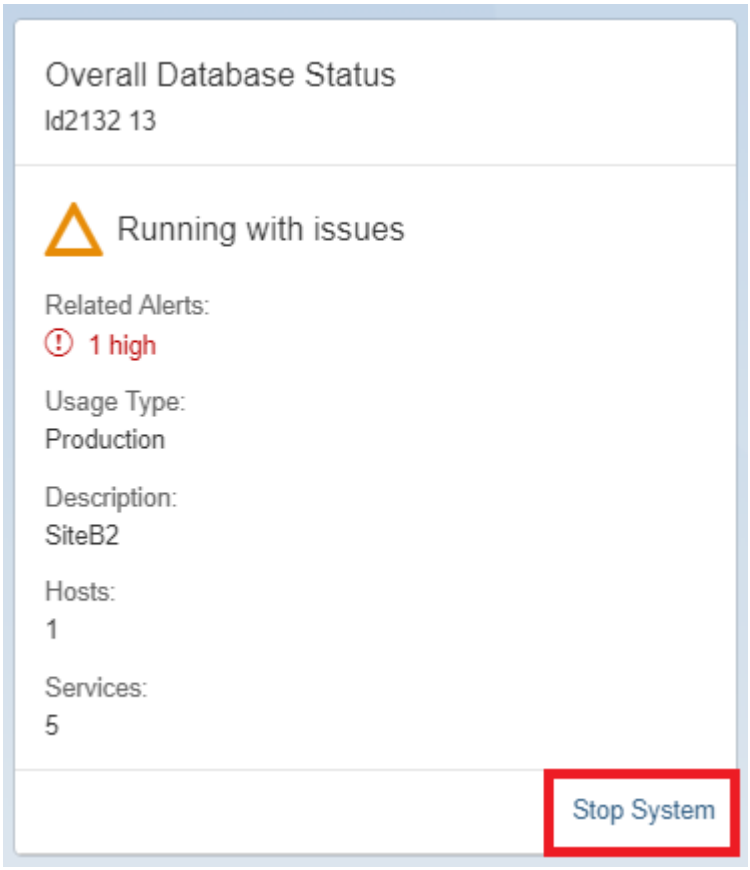
Prerequisites

- The secondary system must be offline.

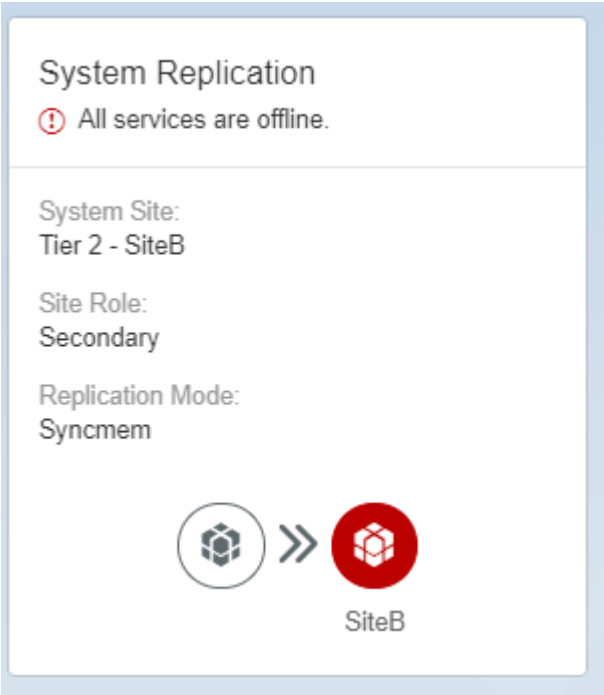
- You need the operating system user to disable system replication with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* and *Connect to a Resource using Database Credentials*.

Procedure

1. Stop the secondary system from the *Overall Database Status* tile.

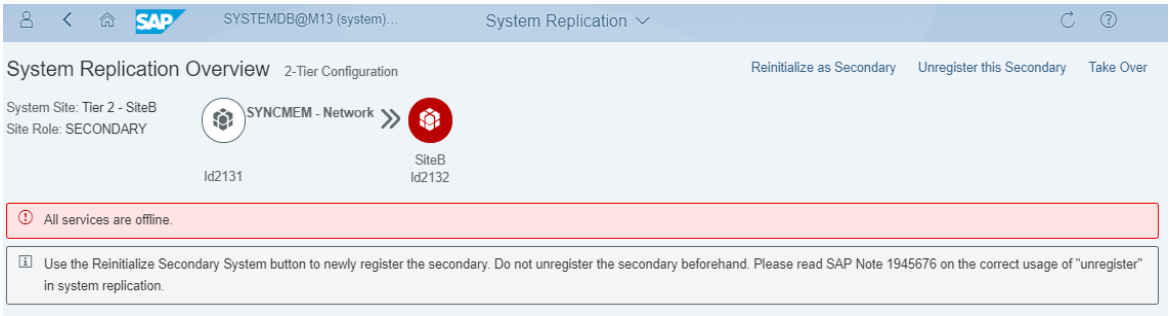


The system replication tile on the overview page of the stopped secondary system indicates that all services are now offline.



2. Choose the system replication tile on the overview page of the stopped secondary system.

   The system replication overview opens giving the possibility to unregister this secondary system:

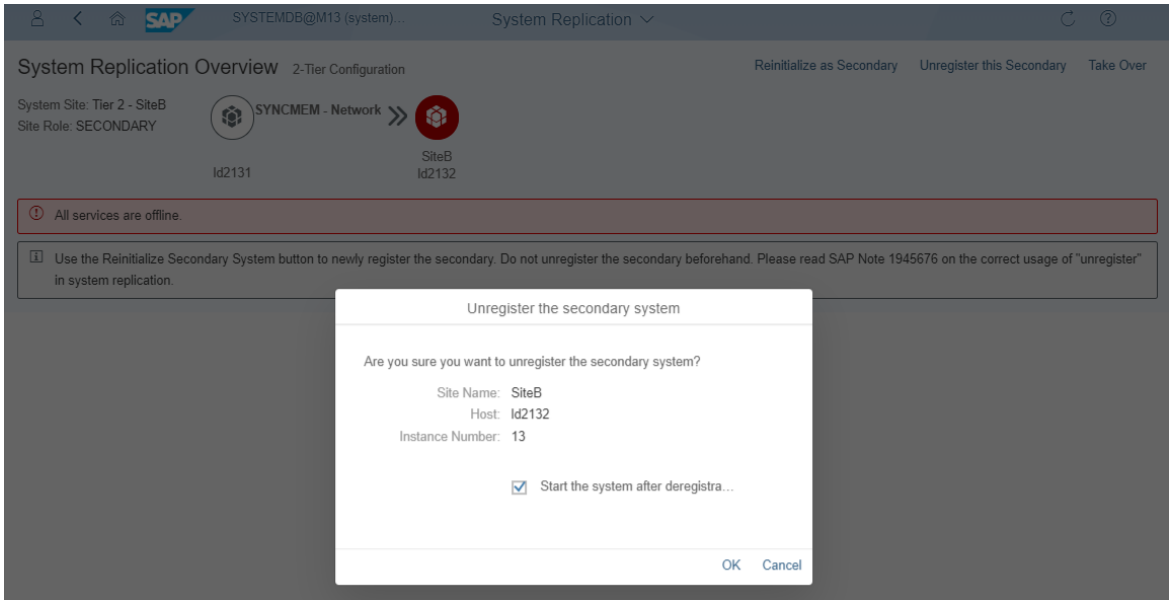


3. Choose *Unregister this Secondary* on the top right.

   A confirmation dialog opens.

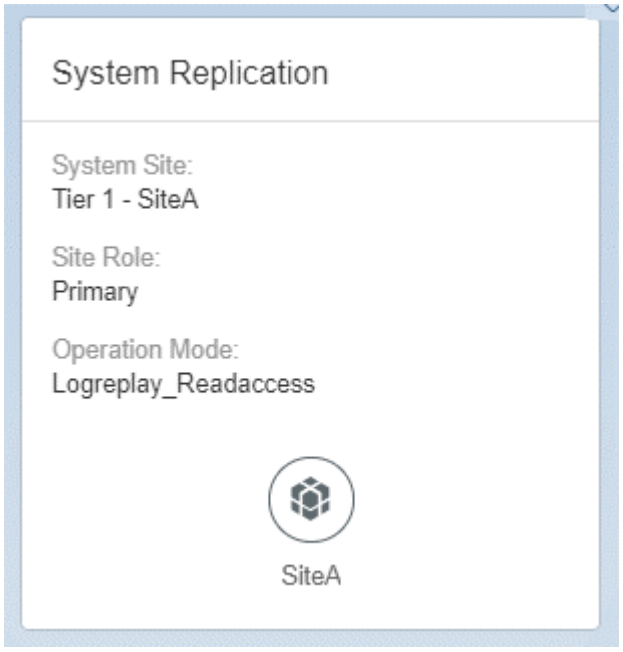
4. Optional: Depending whether your system should be online or offline after unregistering it, check the *Start system after unregistration* option in the confirmation dialog and confirm.

   For more information, see *SAP Note 1945676*.

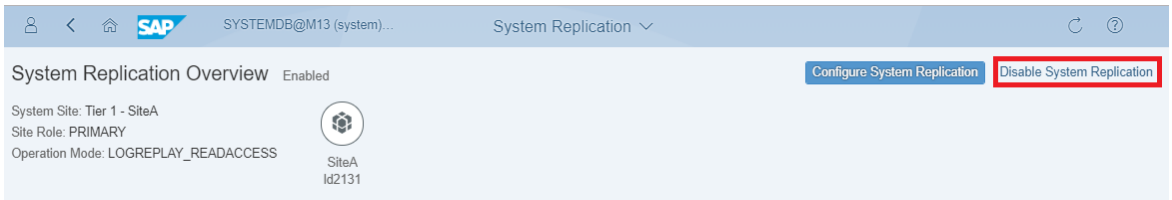


5. Switch now to the primary system. Choose the system replication tile of the primary system.

   This tile indicates that the primary system is not connected to the secondary system anymore:

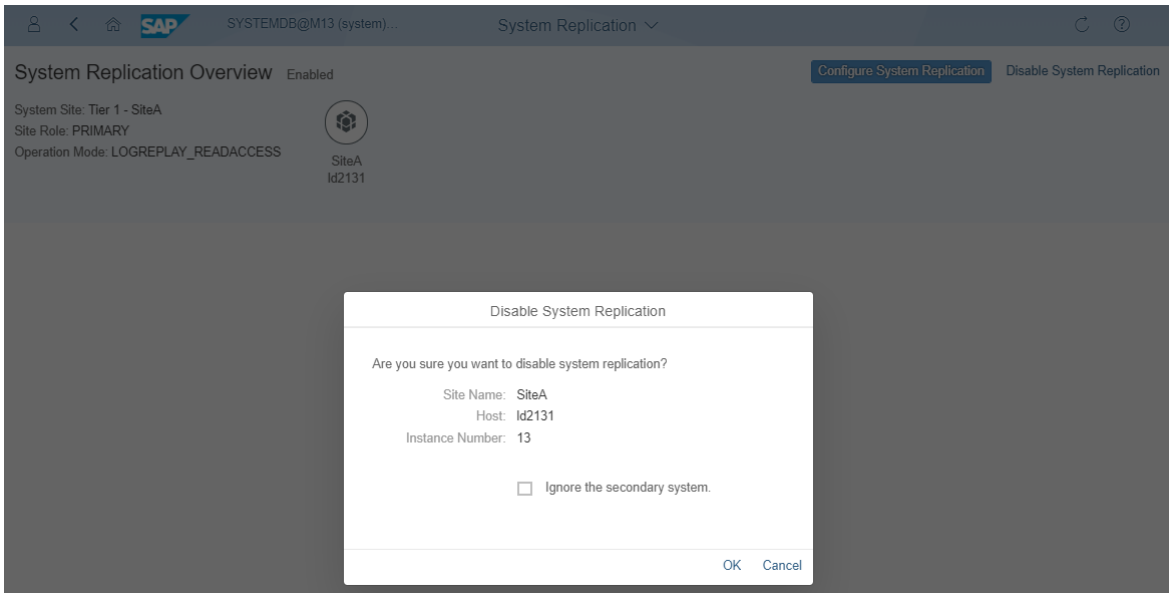


6. On the system replication overview choose *Disable System Replication* on the top right.



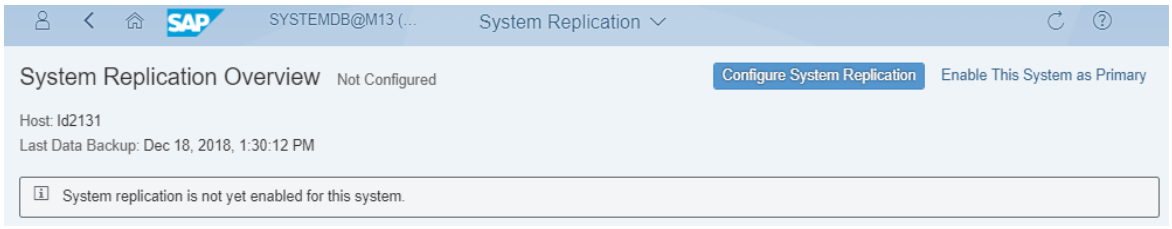
   A confirmation dialog opens.

7. Confirm that you want to disable system replication.



The *Ignore the secondary system* option allows you to disable the primary system even though the secondary is still attached. This could be relevant, if the secondary has been uninstalled in the meantime.

On the overview you can now see that the primary system is disabled and system replication is not configured.



Related Information

SAP Note 1945676

3.10.2 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 --id=<secondarySiteID>
   ```

   > **i Note**
   >
   > If system replication is out of sync and you need to register again the initial secondary system, use the command hdbnsutil –sr_register. It is not necessary to unregister the secondary system before registering it again.
   >
   > For an overview of `hdbnsutil -sr_unregister` use cases, see *SAP Note 1945676*.

3. Disable system replication on the primary system as follows:

   ```
   hdbnsutil -sr_disable
   ```

Related Information

SAP HANA System Replication Command Line Reference [page 75]

SAP Note 1945676

3.10.3 Disable SAP HANA System Replication with the SAP HANA Studio

You can disable SAP HANA system replication in an SAP HANA system using the SAP HANA studio.

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. Unregister the secondary system as follows:
   a. In the *Systems* view, right-click the primary system and choose *Configuration and Monitoring* > *Configure System Replication*.
   The *Configure System Replication* dialog opens.

   > **i Note**
   >
   > You can also access the *Configure System Replication* dialog from the *Landscape* > *System Replication* tab.

   b. Choose *Unregister secondary system* and then *Next*.
   c. Enter the required system information and choose *Next*.
   d. Review the configured information and choose *Finish*.
2. Disable system replication on the primary system as follows:
   a. In the *Systems* view, right-click the primary system and choose *Configuration and Monitoring* > *Configure System Replication*.
   b. Choose *Disable system replication* and choose *Next*.
   c. Review the configured information and choose *Finish*.

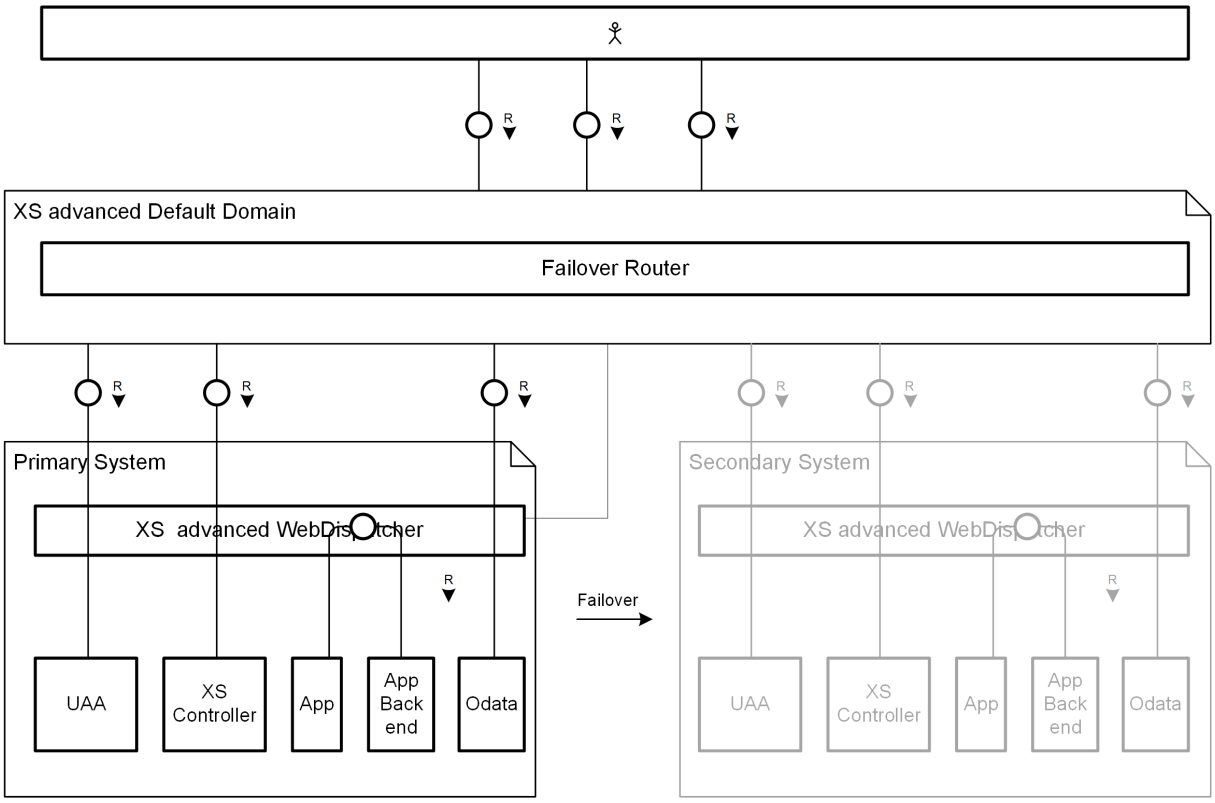
3.11 SAP HANA System Replication Setup for XS Advanced Runtime

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.

After the takeover, all XS advanced services are started which in turn brings up all XS advanced applications on the secondary system. Moreover, XS advanced services and applications use the same domains and certificates that were present in the primary system before the takeover started.

For this to work, the central point for XS advanced requests must be the same on the primary and the secondary systems. This is established by using a failover router similar to the host auto-failover setup. For more information about the host auto-failover setup, see *Host Auto-Failover Setup with XS Advanced Runtime*.



SAP HANA System Replication Setup for XS advanced

In case the failover router terminates SSL, the same rules apply as described in *Host Auto-Failover Setup with XS Advanced Runtime*.

For more information, see *SAP Note 2300936*.

Related Information

Host Auto-Failover Setup with XS Advanced Run Time

SAP Note 2300936

3.12 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.

  > **i** 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
SAP HANA Database Backup and Recovery
Copying and Moving Tenant Databases
Increase the System Isolation Level
Linux Kernel Parameters
SAP Note 401162
SAP Note 2382421

4 SAP HANA System Replication: Takeover and Failback

Learn how to perform a takeover and a failback for planned or unplanned downtimes of the primary system.

How can I perform a takeover and a failover?

This section describes how to perform a takeover and a failover with three different tools. Generally, you have to perform the following steps:

- Switch your active system from the current primary system to the secondary system
- Register the former primary as the secondary with the now active primary system. The roles are switched compared to the original setup.

What types of takeover are available?

You can perform a standard takeover, a takeover with handshake, or an invisible takeover.

Perform a standard takeover when there are unplanned downtimes of the primary system. This can happen when the primary system has a serious problem and is not available. You can also perform a standard takeover for planned downtimes such as near zero downtime upgrades (NZDU).

Perform a takeover with handshake for a safe planned takeover while the primary is still running (for example, in NZDU). All new writing transactions on the primary system will be suspended and the takeover is first executed when all redo log is available on the secondary system. For more information, see *Takeover with Handshake*.

Perform an invisible takeover to achieve an automatic recovery of your sessions after takeover to your new primary. For dedicated client applications this takeover will be invisible. For more information, see *Invisible Takeover*.

Where can I find more information?

The following SAP Notes are relevant for a full understanding of the basic concepts described in this chapter:

SAP Notes

SAP Note	Title
2063657	SAP HANA System Replication Takeover Decision Guideline

SAP Note	Title
2053504	System replication: Hanging client processes after a takeover

Related Information

Takeover [page 90]
Client Connection Recovery After Takeover [page 96]
Invisible Takeover and Restart [page 102]
Takeover with Handshake [page 104]
Failback [page 105]
Checking the SAP HANA System Replication Status [page 166]

4.1 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 a disaster or because of planned downtime) and a decision has been made to take over to the secondary data center, you can perform a takeover on your secondary system. 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*.

We recommend to use third-party external tools to check if hosts, the network, and data center are still available. 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 System Replication Status*.

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'.

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 *Perform a Takeover with the SAP HANA Cockpit*.
- hdbnsutil
  For more information, see *Perform a Takeover with hdbnsutil*.
- SAP HANA studio
  For more information, see *Perform a Takeover with the SAP HANA Studio*.

Related Information

Checking the SAP HANA System Replication Status [page 166]
Perform a Takeover with SAP HANA Cockpit [page 91]
Example: Perform a Takeover with the SAP HANA Cockpit [page 92]
Perform a Takeover with hdbnsutil [page 95]
Perform a Takeover with the SAP HANA Studio [page 96]
Client Connection Recovery After Takeover [page 96]
Invisible Takeover and Restart [page 102]
Takeover with Handshake [page 104]
Example HA/DR Provider Implementation
SAP Note 2063657

4.1.1 Perform a Takeover with SAP HANA Cockpit

You can perform a takeover on your secondary system using the SAP HANA cockpit.

Prerequisites

You've navigated to the *Database Overview* page of the database you want to manage. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.

- It's recommended to stop the primary system before starting a takeover.

  > **i Note**
  >
  > If you're performing a takeover as part of a planned downtime, first make sure that the primary system has been fully stopped before performing a takeover to the secondary system.

- The secondary system must be fully initialized.
- You need the operating system user to perform a takeover with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* in the *SAP HANA Administration Guide* and *Connect to a Database With SSO or SAP HANA Credentials*.
- The takeover command can be executed both when the secondary system is offline or online.

Procedure

1. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB) of the secondary system meant to perform the takeover, choose the *System Replication* card.

   The *System Replication Overview* is displayed.
2. Choose *Take Over*.
3. To start the takeover, click *Start Takeover* in the *Takeover* dialog.

   You can also start a takeover with handshake by choosing to fully synchronize the secondary system. For more information about the takeover with handshake, see *Takeover with Handshake* in the *SAP HANA Administration Guide*.
4. Stop the primary system from the *Services* card on the *Database Overview*.

Related Information

Connect to a Database With SSO or SAP HANA Credentials
Operating System User <sid>adm
Takeover with Handshake
SAP HANA System Replication

4.1.1.1 Example: Perform a Takeover with the SAP HANA Cockpit

Learn how to perform a takeover with the SAP HANA cockpit.

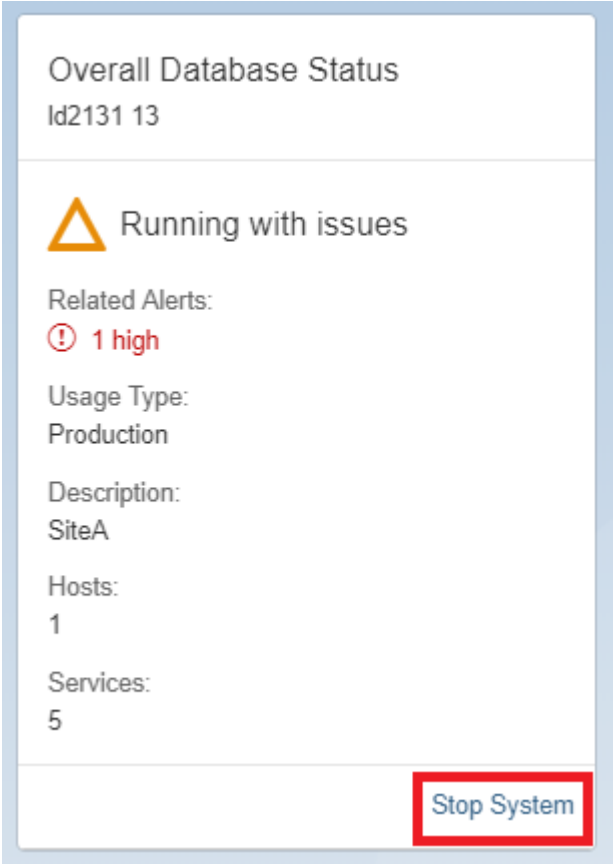
Prerequisites

You need the operating system user to perform a takeover with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* and *Connect to a Resource using Database Credentials*.

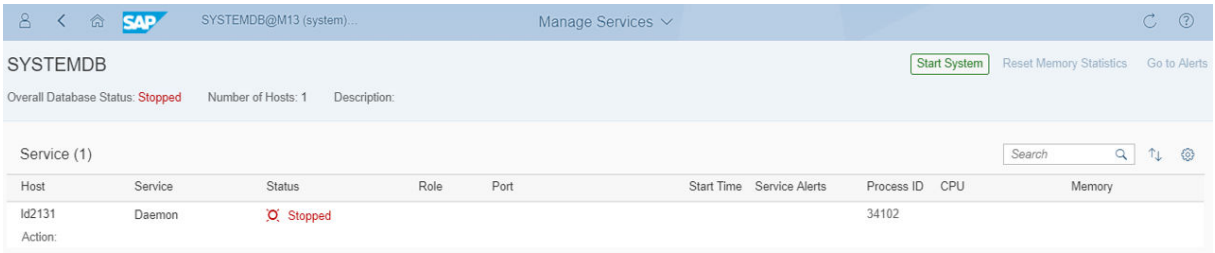
Context

This examples shows how to perform a takeover from the primary system Site A (Host Id2131) to the secondary system Site B2 (Host Id2132).

For a planned takeover (for example, within the process of a near zero downtime upgrade) stop the primary system (Site A) from the *Overall Database Status* tile.

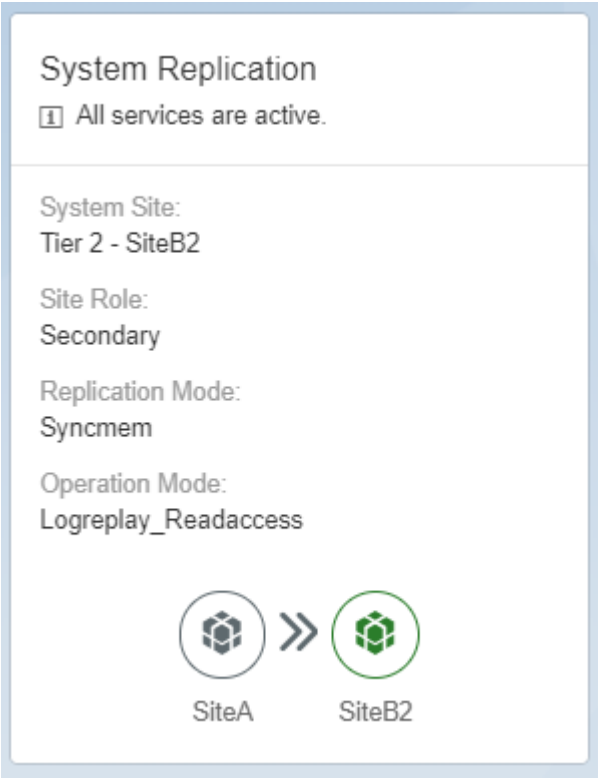


On the *Manage Services* page, you should see that the primary system (Site A) was stopped.



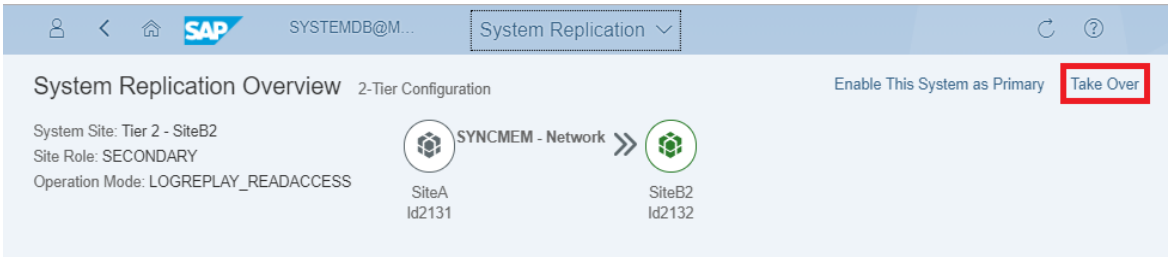
Procedure

1. Choose the system replication tile on the secondary system's (Site B2) overview page.



The *System Replication Overview* opens.

2. Choose *Take Over* on the top right.



You can also choose to start a takeover with handshake. For more information, see *Takeover with Handshake*.

After the takeover is finished, this system is not functioning as a secondary system anymore. It becomes the new primary.

4.1.2 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.

> **i 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 [page 257]
M_SYSTEM_REPLICATION_TAKEOVER_HISTORY System View [page 266]

4.1.3 Perform a Takeover with the SAP HANA Studio

You can perform a takeover on your secondary system using the SAP HANA studio.

Prerequisites

- The secondary system must be fully initialized.
  You can check this in 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.
- You are logged on to the secondary system as the operating system user (user <sid>adm) or can enter these credentials when prompted.

> **i 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

1. In the *Systems* view, right-click the secondary system and choose *Configuration and Monitoring* > *Configure System Replication*.
2. Choose *Perform Takeover* from the actions list.
3. Enter the required system information and choose *Next*.
4. Review the information and choose *Finish*.

Results

The secondary system is now the production system. If the system is already running, it comes out of recovery mode and becomes fully operational immediately: it replays the last transaction logs and starts to accept queries. If the system is offline, it takes over production operation when you start it.

4.1.4 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.

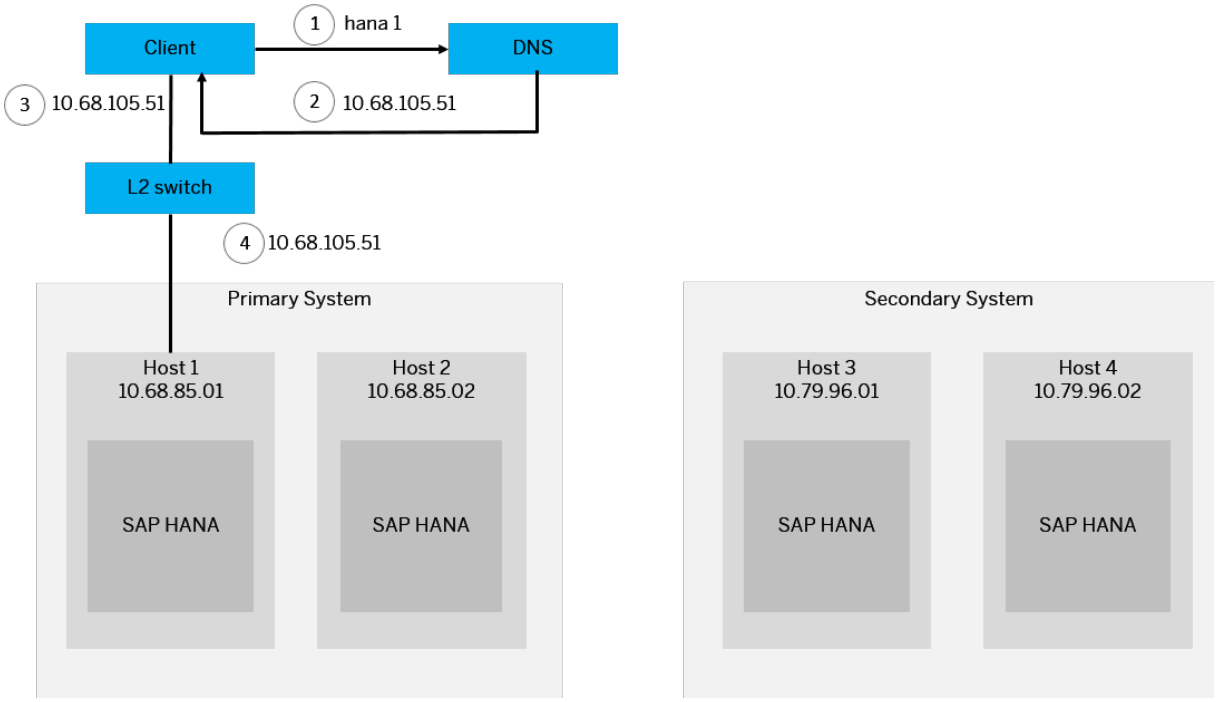
There are different possibilities for enabling client connection recovery:

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.

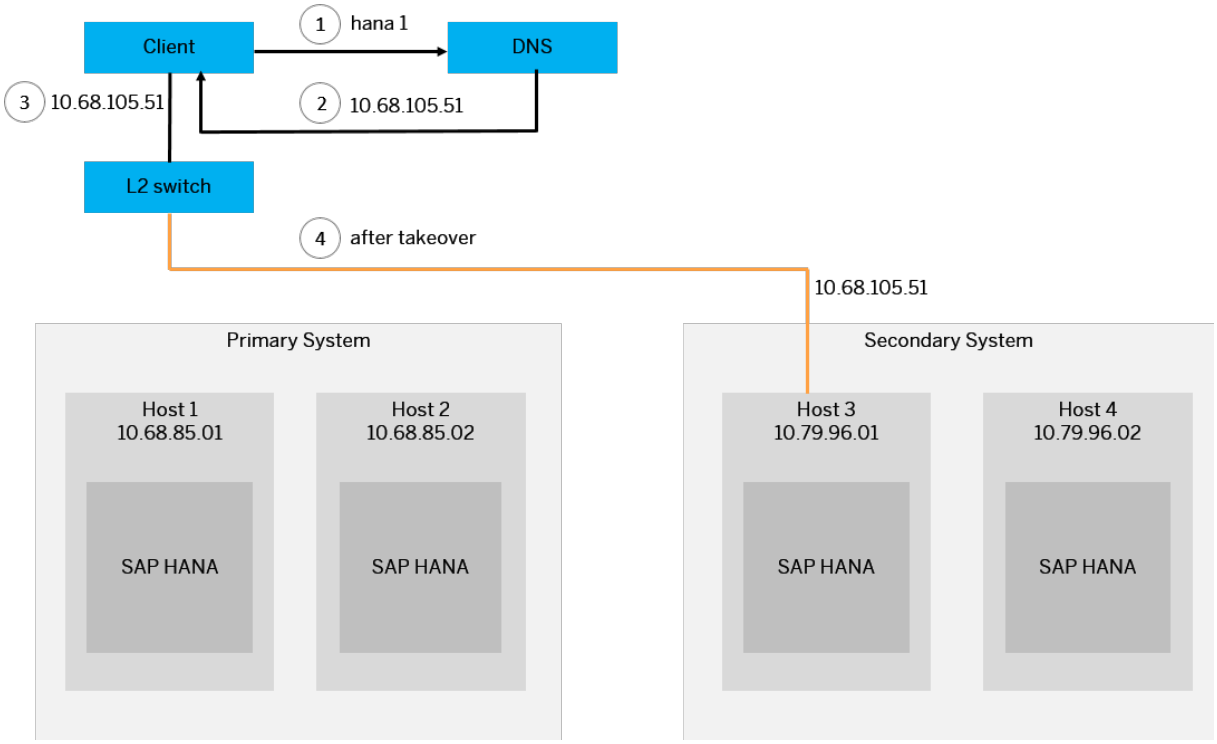
1. Network-based IP Redirection

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:

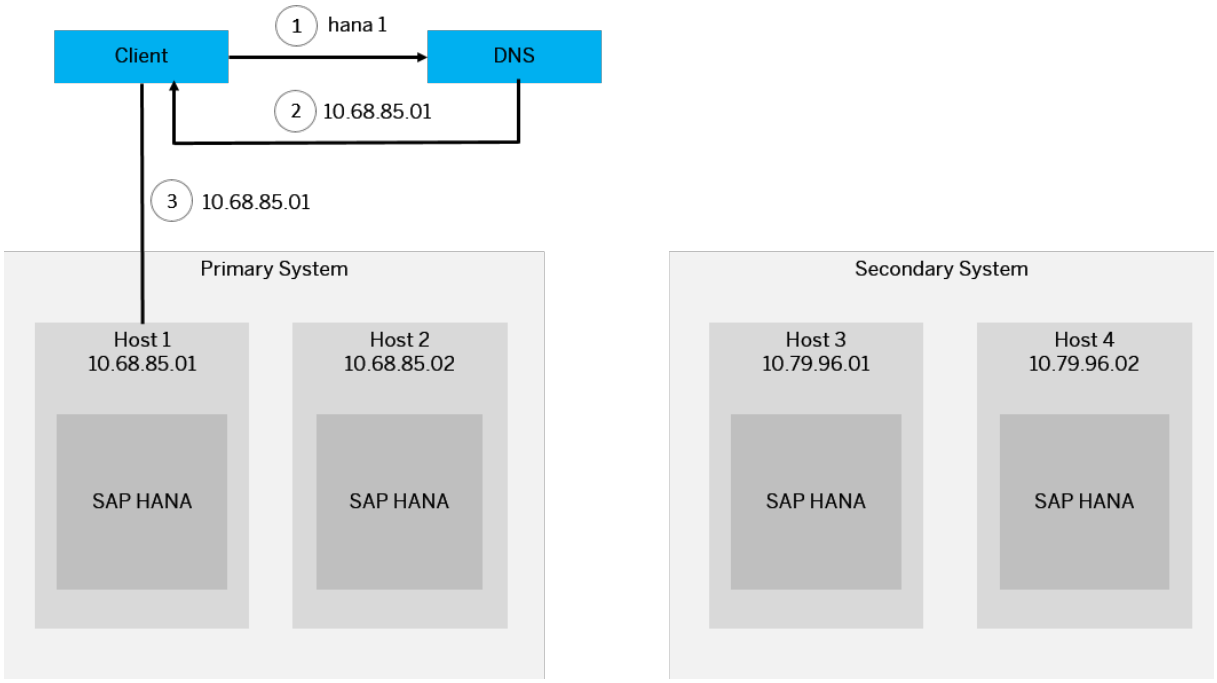


IP redirection can be implemented using a number of actual techniques, for example with the use of Linux commands which affect the network Address Resolution Protocol (ARP) tables (ip addr add/del...), by configuring L2 network switches directly, or by using cluster management software. Following the IP redirection configuration, the ARP caches should be flushed, to provide an almost instantaneous recovery experience to clients.

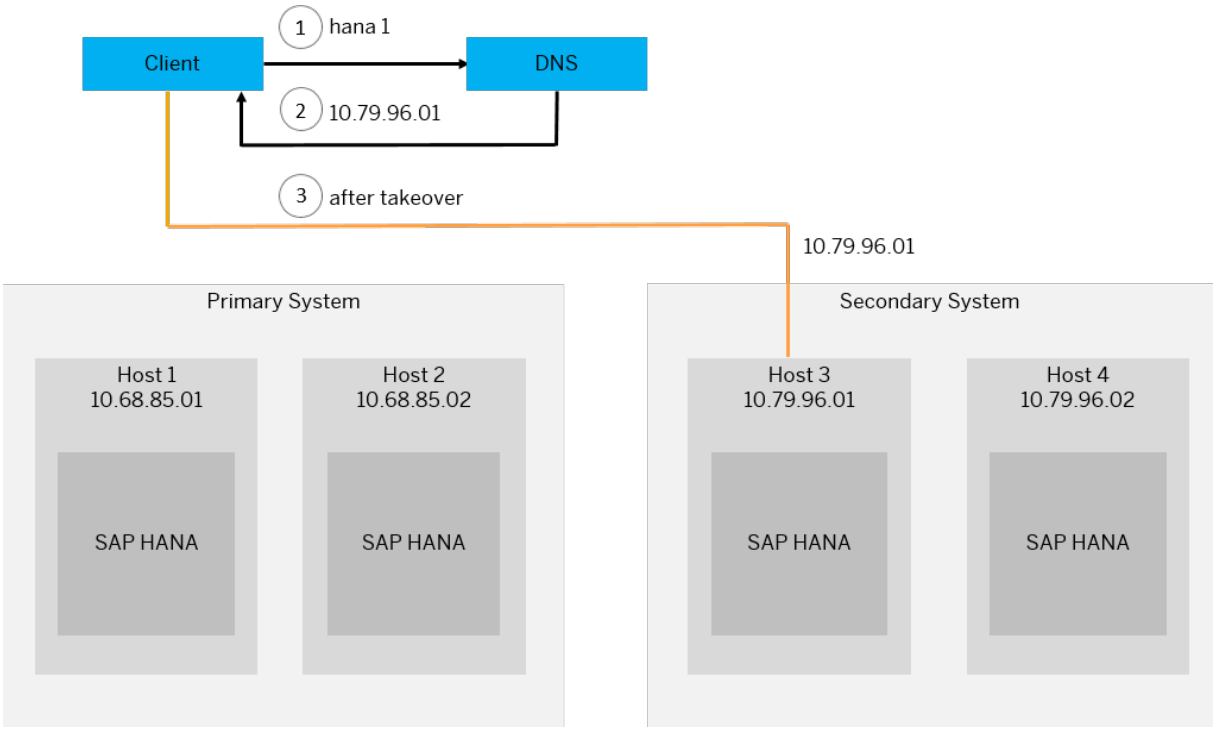
IP redirection requires that both the primary and failover host(s) are on the same L2 network. If the standby system is in a completely separate L3 network, then DNS redirection is the preferred alternative solution.

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
Connecting Using Active/Active (Read Enabled) (SAP HANA Platform)
SAP Note 2053504 - System replication: Hanging client processes after a takeover

4.1.5 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.

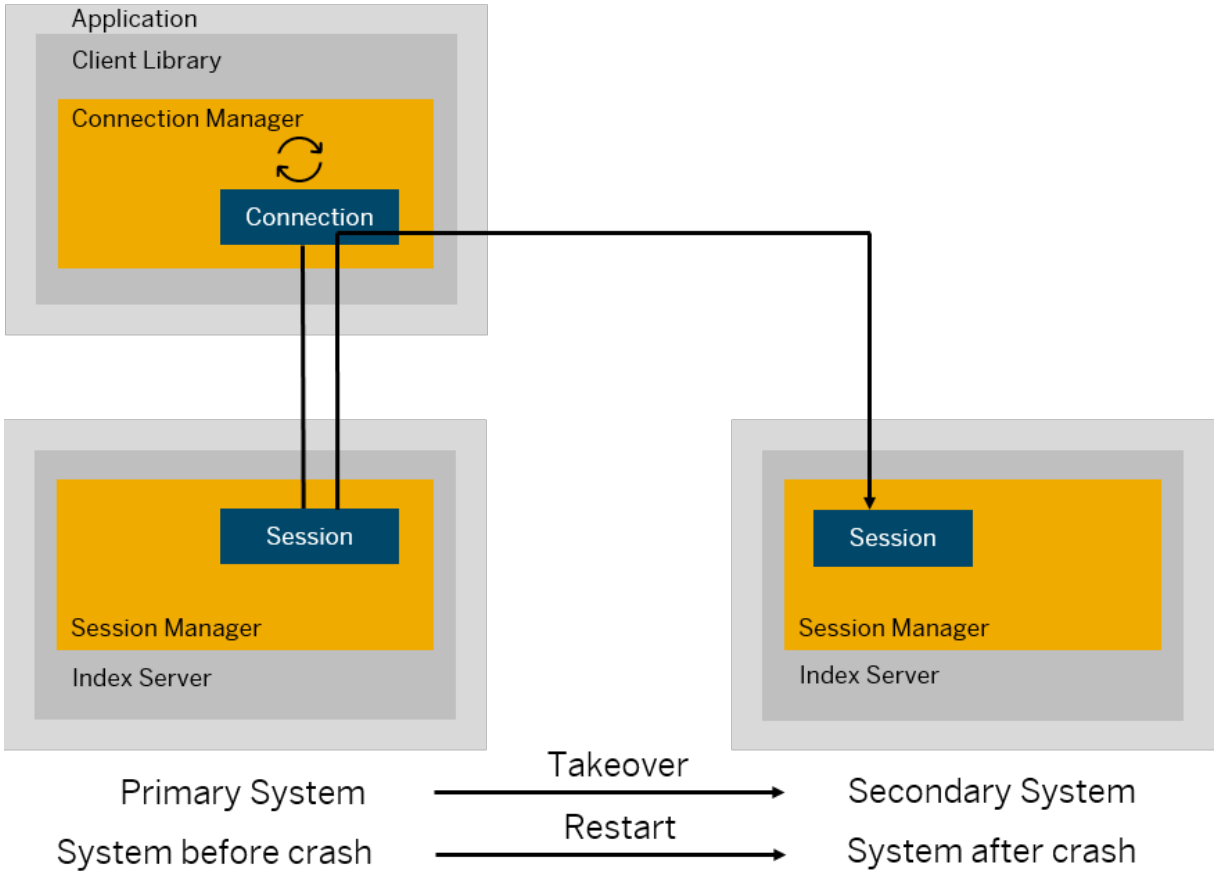
You can perform an invisible takeover to achieve an automatic recovery of your sessions after takeover to your new primary system. For dedicated client applications this takeover will be invisible. Differently from a standard takeover, an invisible takeover ensures that the client reconnects to the primary 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.

> **i** Note
>
> The transparent session recovery is supported by 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.
- 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.

4.1.6 Takeover with Handshake

The takeover with handshake ensures that all the sent redo log is written to disk on the secondary system.

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 the former primary system is isolated to avoid a split-brain situation with multiple active primary systems.

The takeover with handshake is ideal for a safe planned takeover while the primary is still running. 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. When performing a takeover with handshake, it is not needed to check the replication status or to stop the old primary before the takeover.

In a nutshell, a takeover with handshake avoids:

- Data loss, because the log is available on the secondary system before the takeover is triggered.
- Split-brain situations, because the former primary will be suspended.

> **i Note**
>
> The takeover with handshake will only be performed if the two previously mentioned conditions are guaranteed. Otherwise, the takeover will be aborted and the primary resumed.

Trigger a takeover with handshake using `hdbnsutil -sr_takeover --suspendPrimary` on the secondary system.

If a primary service cannot be accessed or a service replication is not active or in sync, the takeover will be aborted and reported as an error. In this case, there is no impact on the system and the replication remains as it was. The suspended primary can be unblocked using the `-sr_register` hdnsutil command.

> **i Note**
>
> The following limitations apply:
>
> - It is supported only on the second tier.
> - It is not supported with Dynamic Tiering services.

Related Information

Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 206]

4.1.7 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 151]

4.2 Failback

The failback process is the name for the task of registering the former primary system as a new secondary when it becomes available again.

After a takeover has been carried out, the roles between primary and secondary can be switched over. In the case of a failback, 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.

This is the same procedure as for setting up a normal secondary described in *Configuring SAP HANA System Replication*. However, in this scenario when the new secondary is registered with the new primary, it checks if a delta shipping is possible to resync the two sites rather than carrying out a full data shipping. If a delta shipping is possible, it only ships the delta, which significantly reduces the initialization time during registration of the new secondary. When the new secondary starts, it checks first if there is a local snapshot available from the time when the system was the primary system. If a snapshot is available, the system then checks if it is compatible with the new primary. When both checks are positive, the new secondary can be initialized with a delta replica from the new primary.

You can perform a failback using the following tools:

- SAP HANA cockpit
  For more information, see *Perform a Failback with the SAP HANA Cockpit*.
- hdbnsutil
  For more information, see *Perform a Failback with hdbnsutil*.
- SAP HANA studio
  For more information, see *Perform a Failback with the SAP HANA Studio*.

Related Information

Perform a Failback in SAP HANA Cockpit [page 106]

Example: Perform a Failback with the SAP HANA Cockpit [page 108]
Perform a Failback with the SAP HANA Studio [page 109]
Perform a Failback with hdbnsutil [page 111]
Virtual IP Address Handling [page 132]

4.2.1 Perform a Failback in SAP HANA Cockpit

To perform a failback in the SAP HANA cockpit, register the former primary system as secondary to the current primary.

Prerequisites

You've navigated to the *Database Overview* page of the database you want to manage. See *Getting to the Database Overview Page* in the *SAP HANA Administration with SAP HANA Cockpit* guide.

- You need the operating system user to perform a failback with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* in the *SAP HANA Administration Guide* and *Connect to a Database With SSO or SAP HANA Credentials*.
- The former primary system isn't running.
- The current primary system is running.

Context

In the SAP HANA cockpit, you can perform a failback either from the current primary system or from the former primary system (the future secondary). The configuration steps are the same as described in *Configure System Replication from the Primary System* or *Configure System Replication from the Primary and the Secondary System*.

This procedure describes how to register the former primary as a new secondary. Use the *System Replication* card on the system *Database Overview* page of the former stopped primary to register this system as a new secondary.

Procedure

1. Register the secondary system as follows:
   a. In the *Monitoring* or *All* view, on the *Database Overview* page of the system database (SYSTEMDB) of the former primary system, choose the *System Replication* card.

      The *System Replication* page opens.
   b. Choose *Register as Secondary* on the top right.

      The *Register Secondary System* page opens.

c. On the *Register Secondary System* page, enter the logical name used to represent the secondary system.
d. Select a replication mode. For more information on the available replication modes, see *Replication Modes for SAP HANA System Replication*.
e. Select an operation mode. For more information on the available operation modes, see *Operation Modes for SAP HANA System Replication*.
f. Enter the host of the source system.

> **i Note**
>
> If you're operating a distributed system on multiple hosts, enter the name of the host on which the master name server is running.

g. Check *Start Secondary after Registration*.

2. Review the configured information and choose *Configure* on the bottom right.

Results

The original primary system is now registered as the secondary system with the current primary system (that is, the original secondary system). The secondary system is getting in sync again with the primary system. As such, it's attempting to avoid a full data shipping.

Verify that the secondary system replication status is `All services are active and in sync`.

Related Information

Perform a Failback in SAP HANA Cockpit [page 106]
Configure SAP HANA System Replication from the Primary and the Secondary Systems [page 47]
Replication Modes for SAP HANA System Replication
Operation Modes for SAP HANA System Replication
Connect to a Database With SSO or SAP HANA Credentials
Operating System User <sid>adm
SAP HANA System Replication

4.2.1.1 Example: Perform a Failback with the SAP HANA Cockpit

Learn how to perform a failback with the SAP HANA cockpit.

Prerequisites

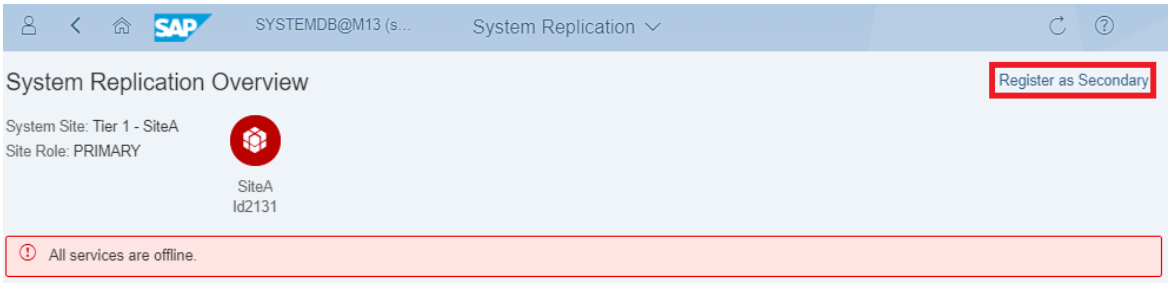
You need the operating system user to perform a failback with the SAP HANA cockpit. For more information, see *Operating System User <sid>adm* and *Connect to a Resource using Database Credentials*.

Context

This example shows how to perform a failback by registering the former primary (Site A) as a secondary system to the new primary (Site B2).

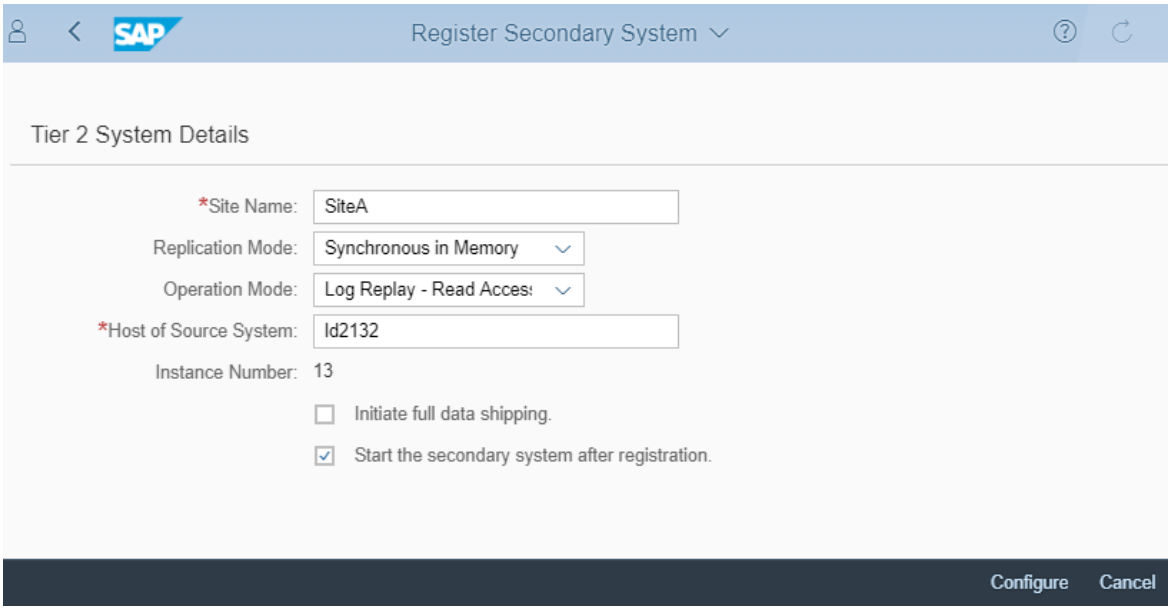
Procedure

1. Choose the system replication tile of the stopped primary system (Site A) and choose *Register as Secondary* on the overview page.

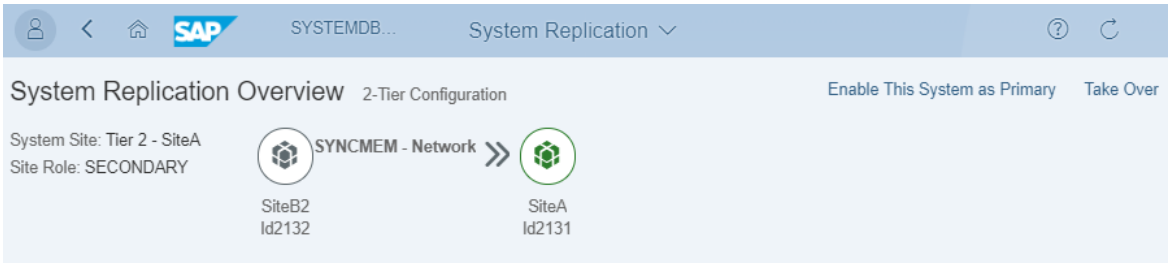


   This will register the primary system (Site A) as a secondary system.
2. On the configuration page, add the name of the former primary system (Site A), the replication and operation modes, as well as the master host name of the former secondary system (Id2132). Finally, choose *Configure*.

The configuration can look like this:



After the configuration is completed, the roles switch: the former primary (Site A) runs as a secondary system to the new primary (Site B2). This new primary (Site B2) is the former secondary system. You can verify the switch on the overview page.



4.2.2 Perform a Failback with the SAP HANA Studio

You can perform a failback using the SAP HANA studio.

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 former primary system is not running.
- The current primary system is running.

Context

To fail back, you must attach your former primary system as new secondary system to the current primary system.

Procedure

1. Register the former primary system as the new secondary system as follows:
   a. In the *Systems* view, right-click the primary system and choose *Configuration and Monitoring* > *Configure System Replication*.
   The *Configure System Replication* dialog opens.

   > **i Note**
   >
   > You can also access the *Configure System Replication* dialog from the *Landscape* > *System Replication* tab.

   b. Choose *Register Secondary System* and then *Next*.
   c. Enter the required system information and the logical name used to represent the system.

   > **i 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.

   d. Specify the log replication mode. For more information on the available replication modes, see *Replication Modes for SAP HANA System Replication*.
   e. Specify the operation mode. For more information on the available operation modes, see *Operation Modes for SAP HANA System Replication*.
   f. Review the configured information and choose *Finish*.
   g. If necessary, start the former primary system.

   > **i Note**
   >
   > The former primary system is started automatically unless you deselected the corresponding option during configuration.

   The former primary system is now registered as the secondary system with the current primary system (that is, the former secondary system). As the data that is already available in the former primary system cannot be reused, a complete initialization is carried out. This means that a full data replication takes place until the former primary system is fully in sync.
2. Verify that the secondary system replication status is `All services are active and in sync`.
   You can see this status in the Administration editor on the *Overview* tab.

Results

The former primary system and the former secondary system switched their roles.

Related Information

Configuring SAP HANA System Replication [page 40]
Replication Modes for SAP HANA System Replication [page 12]
Operation Modes for SAP HANA System Replication [page 14]

4.2.3 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 sidadm

Configure SAP HANA System Replication with hdbnsutil [page 54]

Example: Configure SAP HANA System Replication [page 57]

5 SAP HANA System Replication: Secondary Time Travel

Learn how to quickly access again data, which was deleted in the original system.

Secondary time travel allows you to start the secondary system or the log replay at a previous point in time.

Related Information

Secondary Time Travel [page 113]
Execute Secondary Time Travel [page 115]
Execute Secondary Time Travel While Replication Continues [page 116]
Configuration Parameters [page 117]
Monitoring Secondary Time Travel [page 118]

5.1 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.

> **i 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 115]
Configuration Parameters [page 117]
Execute Secondary Time Travel While Replication Continues [page 116]
Monitoring Secondary Time Travel [page 118]
Replication Modes for SAP HANA System Replication [page 12]
Configure SAP HANA System Replication with hdbnsutil [page 54]

5.2 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*.

  > **i** 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 117]
M_SYSTEM_REPLICATION_TAKEOVER_HISTORY System View [page 266]
Implementing a HA/DR Provider

5.3 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*.

  > **i** 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 117]

5.4 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

Parameter	timetravel_snapshot_creation_interval
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
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.

5.5 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 CO-ORDINATOR_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.

6 SAP HANA System Replication with Active/Active (Read Enabled)

Active/Active (read enabled) enables SAP HANA system replication to support read access on the secondary system.

What can I learn about Active/Active (Read Enabled) in this section?

After checking all the necessary prerequisites, learn how to configure and monitor an Active/Active (read enabled) system replication. This chapter also provides information about memory management, virtual IP address handling, and authentication methods.

Where can I find more information?

The following SAP Notes are relevant for Active/Active (Read Enabled):

SAP Notes

SAP Note	Title
2447994	SAP HANA Dynamic Tiering Support for SAP HANA System Replication
1681092	Multiple SAP HANA DBMSs (SIDs) on one SAP HANA system
2116157	FAQ: SAP HANA Consistency Checks and Corruptions
2391079	Access restrictions in Active/Active (read enabled) system setup

Related Information

Active/Active (Read Enabled) System Replication [page 121]
Generic Conditions for Active/Active (Read Enabled) [page 123]
Connection Types [page 127]
Configuring an Active/Active (Read Enabled) System Replication [page 124]
Memory Management [page 131]
Virtual IP Address Handling [page 132]
Authentication Methods [page 133]
Monitoring Active/Active (Read Enabled) [page 133]

6.1 Active/Active (Read Enabled) System Replication

Active/Active (read enabled) enables SAP HANA system replication to support read access on the secondary system and is supported by all client APIs.

Active/Active (Read Enabled)

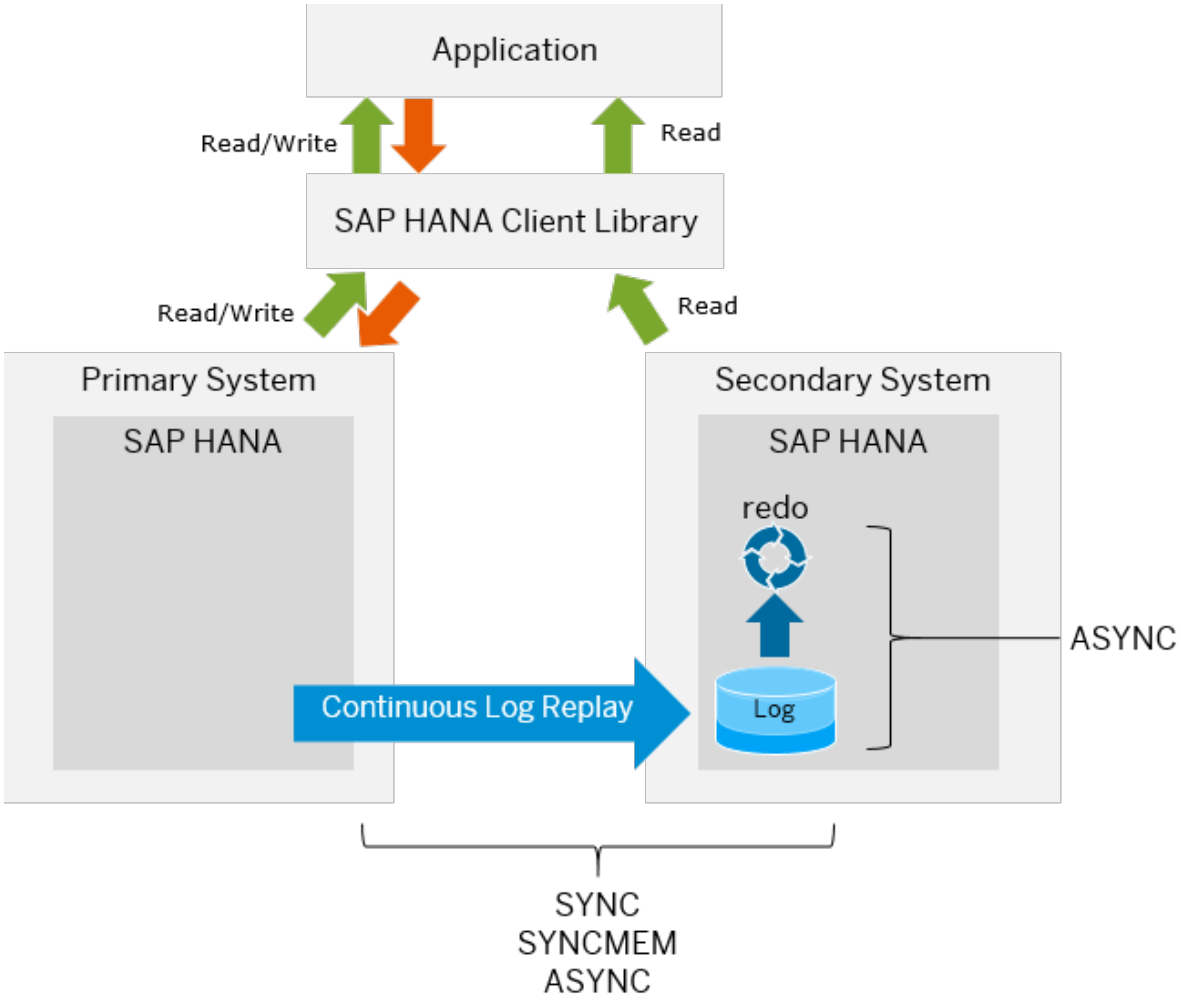
Active/Active (read enabled) reduces the load on the primary system but does not double the capacity; it simply extends read capabilities. In an Active/Active (read enabled) system replication configuration, the SQL ports on the secondary system are open for read access. This makes it possible to use the secondary system for read-intense tasks and to have a better balance of workloads improving the overall performance of the SAP HANA database.

> **i** Note
>
> In case of an outage, all functions concentrate on the secondary system. Because of this, the sizing of the secondary system is important to ensure the right performance in disaster scenarios.

Active/Active (read enabled) is integrated into the SAP HANA system replication solution and gets activated with the operation mode `logreplay_readaccess`. This operation mode provides fast takeovers, reduced need for bandwidth in continuous operation, and support for replication modes such as SYNC (with or without the full sync option), SYNCMEM, and ASYNC. The redo log replay runs asynchronously to the primary operations.

> **i** Note
>
> To use the `logreplay_readaccess` 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.

The following graphic focuses on the Active/Active (read enabled) configuration. The primary system is fully active and supports reading and writing, while the secondary system is enabled for read queries. For a detailed illustration of the general system replication processes, see *Introduction to System Replication*.



In a multitier setup, read access is only supported on the tier 2 secondary and the `logreplay_readaccess` mode is required between the primary and the active secondary systems. In a 3-tier system using Active/Active (read enabled) 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 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*.

License Management

In an Active/Active (read enabled) configuration, the secondary system is operated automatically with the license key of the primary system. Changes of the license key are done on the primary system and replicated to the secondary system. For more information, see *Managing SAP HANA Licenses*.

Related Information

Introduction to System Replication [page 9]
Configuring SAP HANA System Replication
Operation Modes for SAP HANA System Replication [page 14]
Client Support for Active/Active (Read Enabled) - Client Interface Programming Reference
Managing SAP HANA Licenses
Video: SAP HANA Academy - System Replication: Active/Active Read Enabled
SAP Note 2391079
SAP Note 3122356 - How to designate the secondary site for hint based routing

6.2 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

6.3 Configuring an Active/Active (Read Enabled) System Replication

Active/Active (read enabled) is integrated into the SAP HANA system replication solution and gets activated with the operation mode `logreplay_readaccess`.

To configure an Active/Active (read enabled) system replication, follow the procedure described for configuring system replication and select the operation mode `logreplay_readaccess`.

For more information about checking the Active/Active (read enabled) configuration, see *Check the Active/Active (Read Enabled) Configuration*.

Related Information

Configuration Parameters [page 125]
Configuring SAP HANA System Replication
Checking the Active/Active (Read Enabled) Configuration [page 126]
https://www.youtube.com/embed/jR4GBWak1n0

6.3.1 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 65]

6.3.2 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 [page 42]
Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit [page 184]

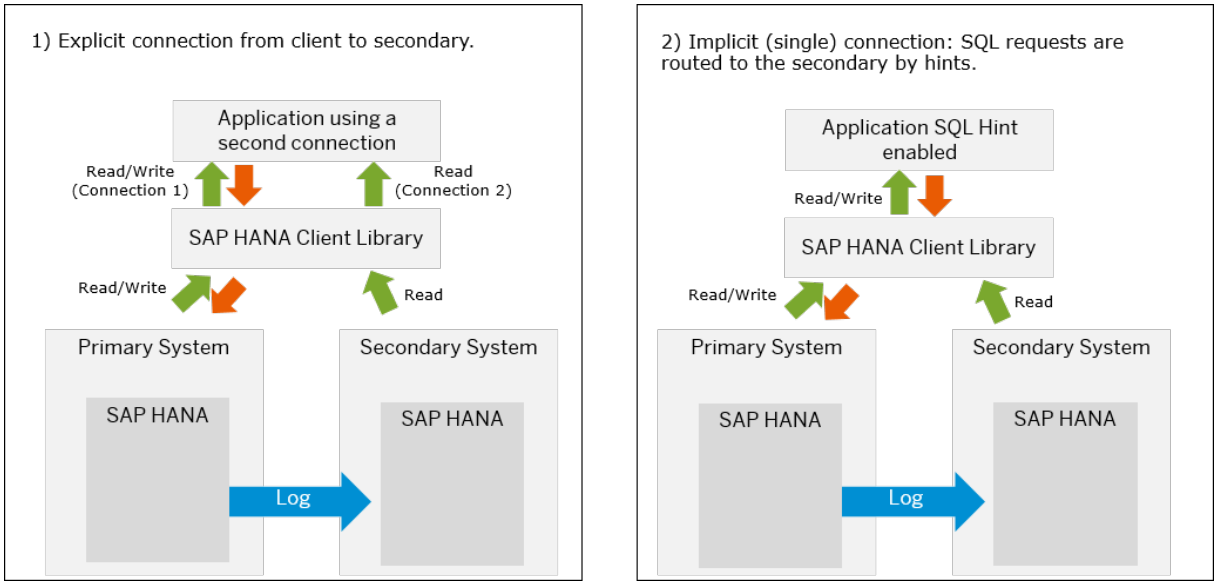
6.4 Connection Types

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:

> **i** 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 129]
Client Support for Active/Active (Read Enabled) - Client Interface Programming Reference

6.4.1 Client Requirements For A Takeover

Configure the client connection so that the client connects to the correct system after failover occurs.

Failover is when a secondary system takes over from a primary system that is offline. Use the following configuration to ensure that the client connects to the correct system when a failover occurs.

- If you are using connection distribution, then you must use the siteType connection property to specify whether the connection is made to the PRIMARY or SECONDARY system. The siteType connection property is required because connection distribution adds server locations from cached topology, which may be out of date because it contains data from before the takeover occurred.
- If you are not using connection distribution, then specify one or more virtual IP addresses (if available) that always point to the desired type of system. For example, if you are connecting to the primary system, then these virtual IP addresses must map to the new primary system (the old secondary system that took over as the primary system) after a failover so that the client can reliably connect to the correct system.
- If you are not using connection distribution and there are no virtual IP addresses, then you must use the siteType connection property to specify whether the connection is made to the PRIMARY or SECONDARY system. To reliably connect both before and after a failover, you must specify one or more locations for both systems (the system that was initially the primary and the system that was initially the secondary). For example, to connect to the current primary system using an ODBC-style connection string, use `siteType=PRIMARY;serverNode=<site1host>:<port>,<site2host>:<port>`, where `<site1host>:<port>` is a location for the site that was initially the primary system, and `<site2host>:<port>` is a location for the site that was initially the secondary system.

Failure to follow these guidelines can result in connecting to the wrong system after a failover (for example, connecting to the secondary system instead of the primary system).

Related Information

Client Connection Recovery After Takeover [page 96]
Virtual IP Address Handling [page 132]

6.4.2 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.

> **i** 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 127]

6.5 Memory Management

Several parameters can be used to set the memory limit for read accesses on the secondary system.

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.

Use the parameters below to set the memory 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

Parameter:	**sr_enable_tracking**
Type:	bool
Default:	on
Section	resource_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.

6.6 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*.

> **i** 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 96]
Failback [page 105]
Connecting Using Active/Active (Read Enabled) (SAP HANA Platform)

6.7 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.

6.8 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 163]
Monitoring SAP HANA System Replication in SAP HANA Cockpit [page 182]

7 SAP HANA System Replication Setups

Learn how to configure a multitier or a multitarget system replication.

Which setups are available for system replication?

Besides the standard setup, in which a primary system ships all the data to the secondary system, you can also configure a multitier or a multitarget system replication.

In a multitier system replication, a tier 2 system replication setup can be used as the source for adding further tiers in a chain. The primary system is always on tier 1. The replication source for the tier 2 secondary system is the primary system, while the replication source for the tier 3 secondary system is the tier 2 secondary. For more information, see *SAP HANA Multitier System Replication*.

In a multitarget system replication, the primary system can replicate data changes to more than one secondary system. For more information, see *SAP HANA Multitarget System Replication*.

Related Information

SAP HANA Multitier System Replication [page 134]
Configuring SAP HANA Multitier System Replication [page 135]
Performing a Takeover and a Failback in SAP HANA Multitier System Replication [page 145]
SAP HANA Multitarget System Replication [page 148]
Example: Configure a SAP HANA Multitarget System Replication [page 150]
Disaster Recovery Scenarios for Multitarget System Replication [page 151]

7.1 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 135]
Performing a Takeover and a Failback in SAP HANA Multitier System Replication [page 145]

7.1.1 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, 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 136]
Example: Configure SAP HANA Multitier System Replication with SAP HANA Cockpit [page 139]
Example: Configure SAP HANA Multitier System Replication with hdbnsutil [page 142]
Example: Configure SAP HANA Multitier System Replication with SAP HANA Studio [page 144]
Operation Modes for SAP HANA System Replication [page 14]

7.1.1.1 Supported Replication Modes Between Systems

In a multitier system replication scenario, the following replication mode combinations are supported.

Replication Mode Combinations

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 1 and tier 2 are located in a local data center for fast takeover.

Tier1 to Tier 2	Tier 2 to Tier 3	Description	Use Case
		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 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 IOalso 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.	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
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	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.

Tier1 to Tier 2	Tier 2 to Tier 3	Description	Use Case
		replication can have an impact on the primary.	
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.

7.1.1.2 Example: Configure SAP HANA Multitier System Replication with SAP HANA Cockpit

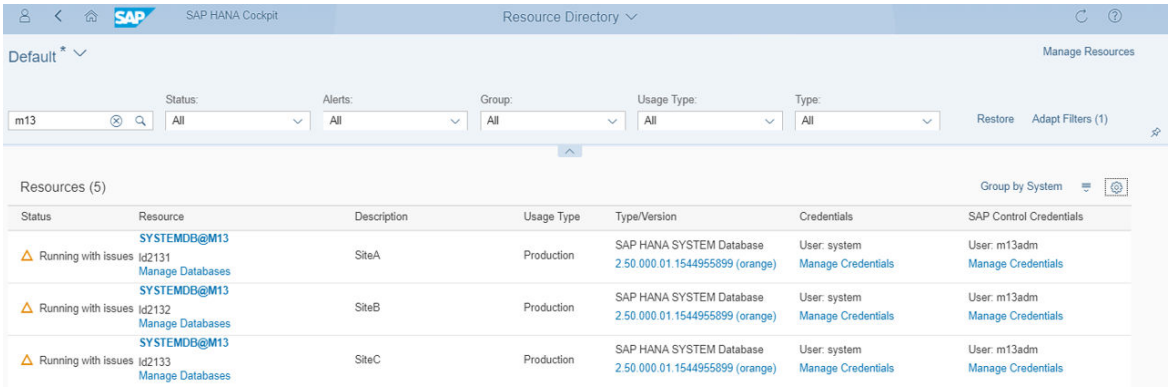
Learn how to configure a multitier system replication landscape with the SAP HANA cockpit from the primary system.

Prerequisites

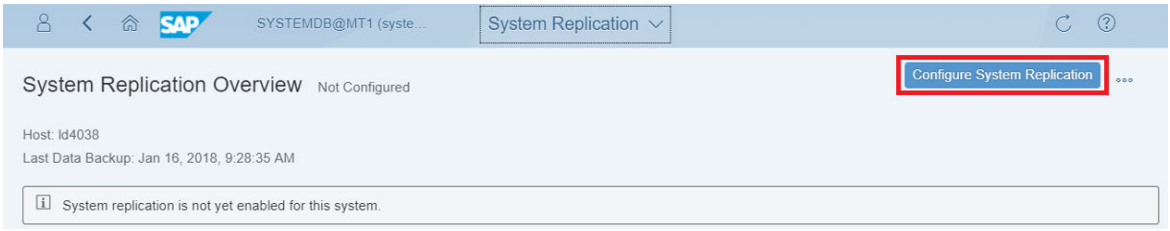
- You have considered all the general prerequisities needed to configure system replication. For more information, see *General Prerequisites for Configuring SAP HANA System Replication*.
- All three systems A,B, and C are registered in the SAP HANA cockpit.

Procedure

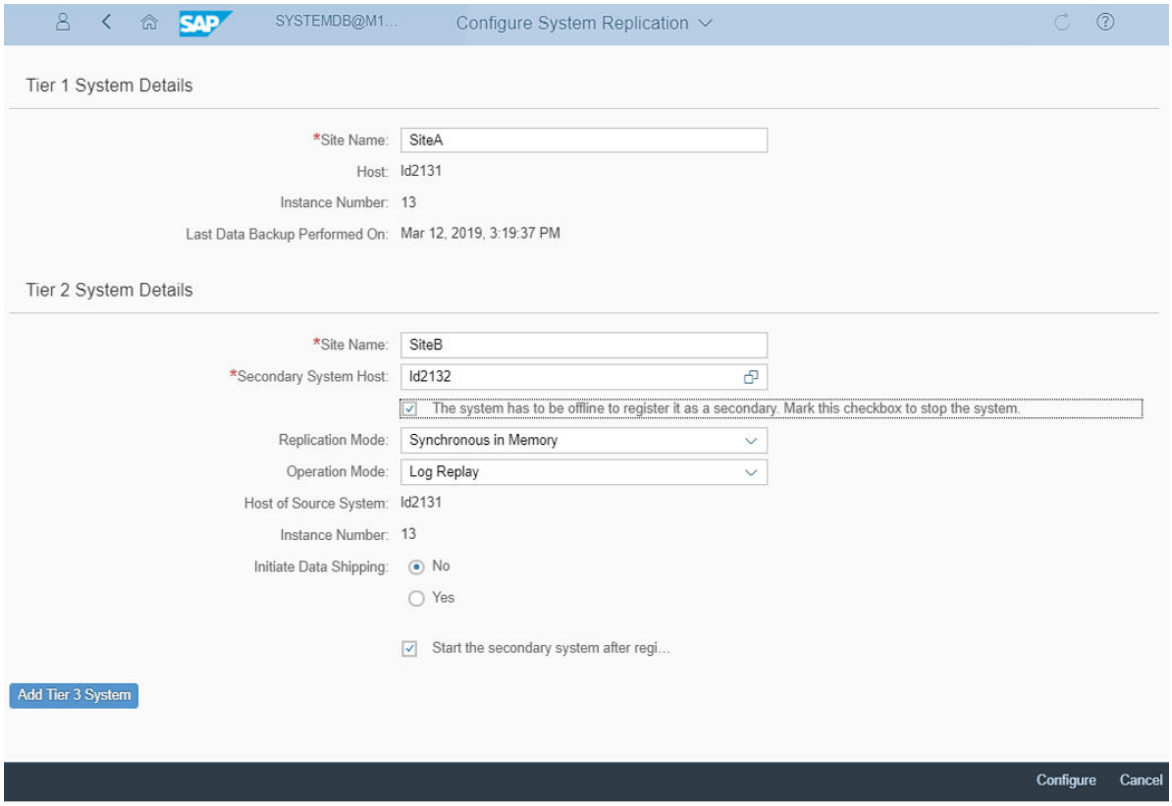
1. On the *Resource Directory* page, choose Site A or the SAP HANA system that will be the primary system.



2. On the *System Overview* page, choose the system replication tile.
3. Choose *Configure System Replication* on the top right to configure system replication.

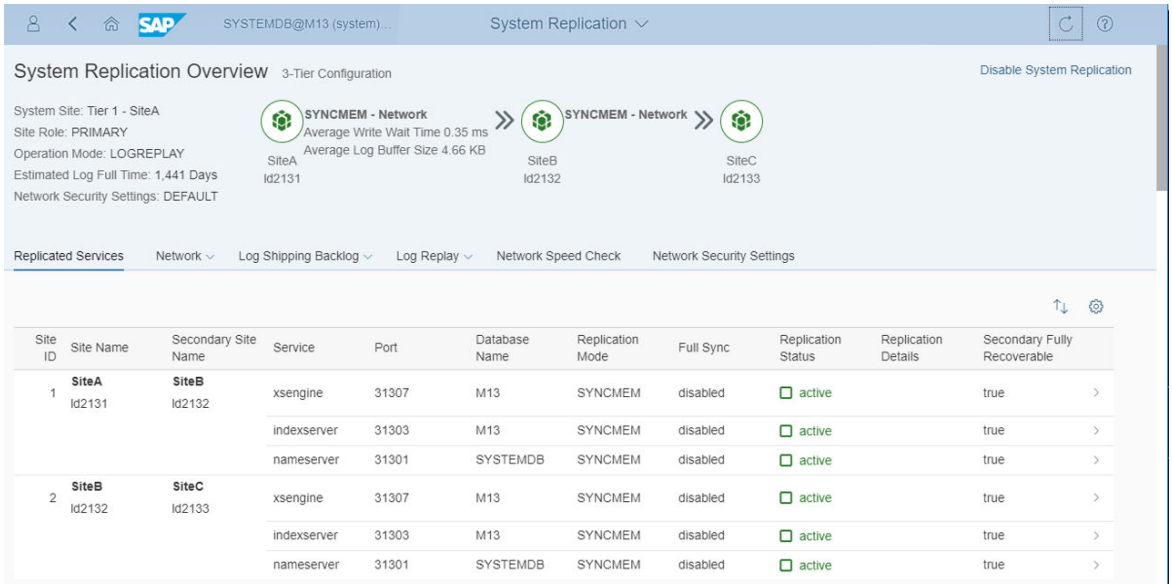


4. Enter the required information on the *System Replication Configuration* page.



5. Choose *Add 3 tier system* and enter the requiered information for the tier 3 secondary system.
6. Choose *Configure* on the bottom right.

   On the system replication overview, you should now see the tier 3 configuration.



Related Information

General Prerequisites for Configuring SAP HANA System Replication [page 36]
Supported Replication Modes Between Systems [page 136]
Operation Modes for SAP HANA System Replication [page 14]
Example: Configure SAP HANA System Replication with the SAP HANA Cockpit [page 50]

7.1.1.3 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 36]
Supported Replication Modes Between Systems [page 136]
Operation Modes for SAP HANA System Replication [page 14]

7.1.1.4 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.

> **i 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 36]
Supported Replication Modes Between Systems [page 136]

7.1.2 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 146]
Example: Restore the Original SAP HANA Multitier System Replication Configuration [page 147]

7.1.2.1 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 136]

7.1.2.2 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.

7.2 SAP HANA Multitarget System Replication

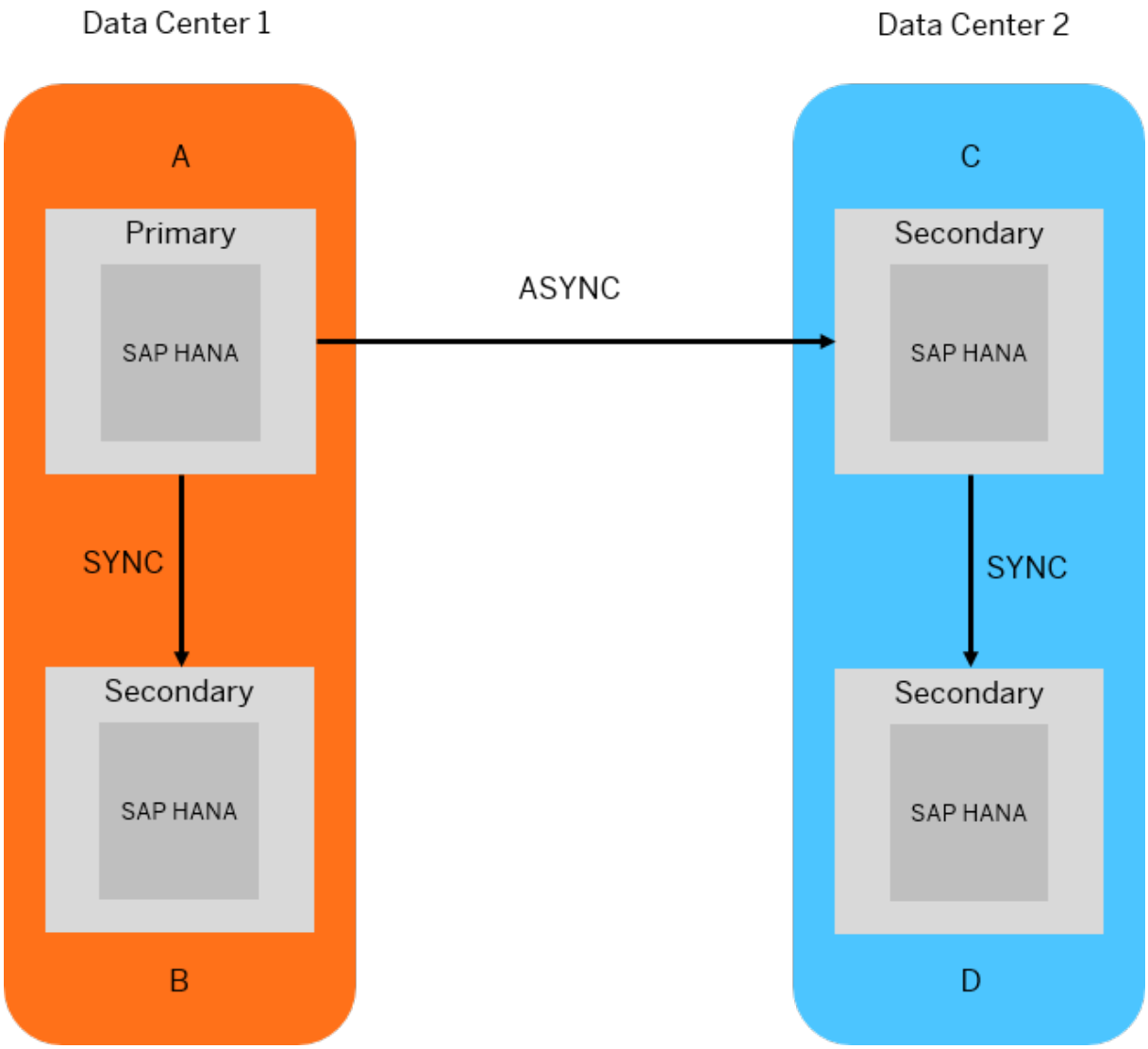
In a multitarget system replication, primary and secondary systems can replicate data changes to more than one system.

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 150]
Operation Modes for SAP HANA System Replication [page 14]

Disaster Recovery Scenarios for Multitarget System Replication [page 151]
Full Sync Option for SAP HANA System Replication [page 62]
Log Retention [page 21]
Log Retention and Multitarget System Replication [page 25]
Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites [page 247]
Use Multitarget System Replication for Near Zero Downtime Upgrades [page 210]
Active/Active (Read Enabled)

7.2.1 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

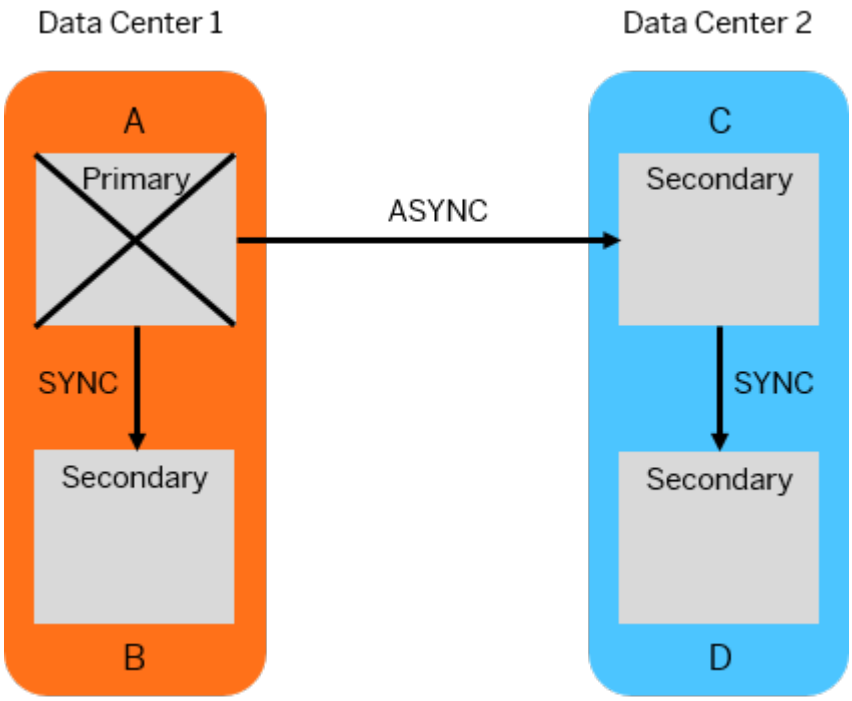
1. On primary system A in data center 1:
   a. Create backups.
   b. Enable system replication.
2. On the local secondary system B in data center 1:
   a. Stop the system.
   b. Register it to A.
   c. Start the system.
3. On the remote secondary system C in data center 2:
   a. Stop the system.
   b. 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.

7.2.2 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



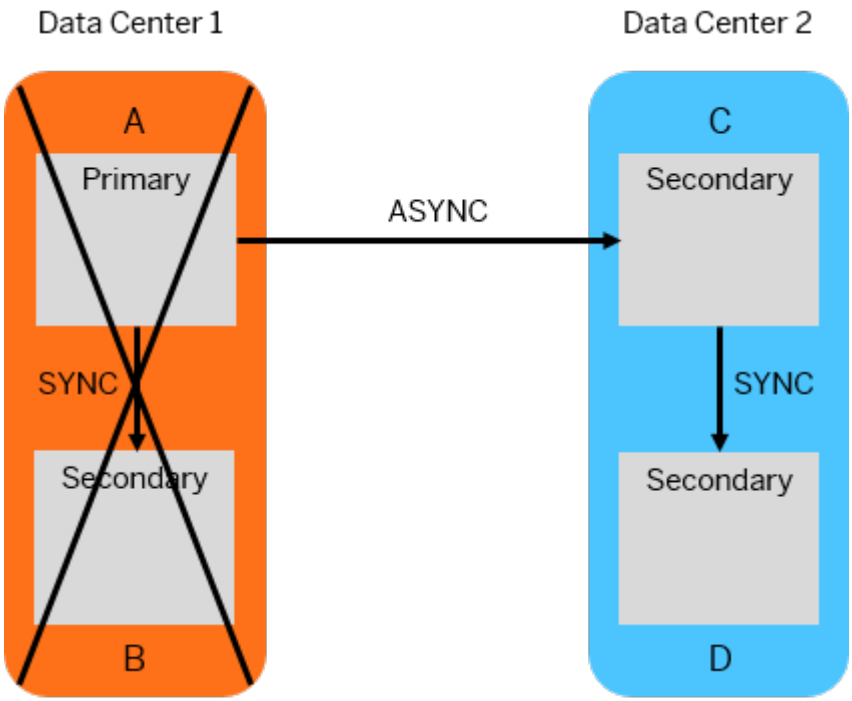
When primary system A fails, proceed as follows:

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.

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*.

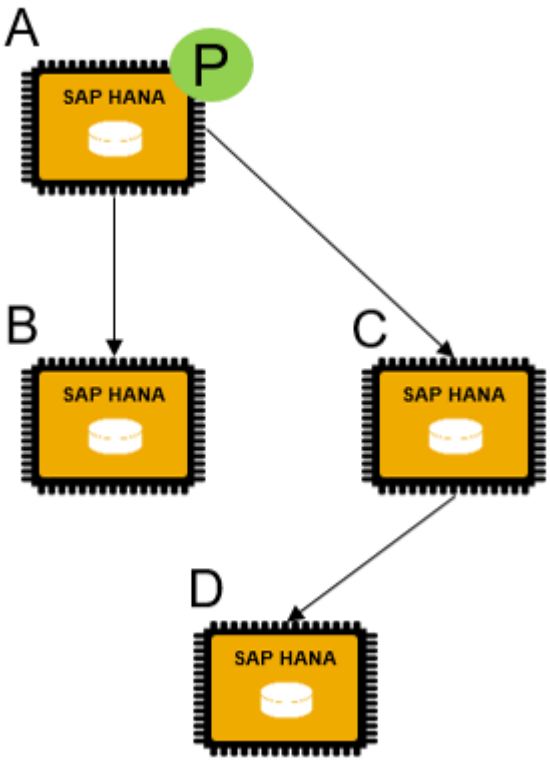
Related Information

Performing a Takeover
Performing a Failback
Full Sync Option for SAP HANA System Replication [page 62]
Log Retention [page 21]
Automatic Registration After Takeover [page 104]

7.2.3 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:



In the configuration parameter both site names are given in a comma-separated list. Optionally, you can also specify the replication mode for each site by appending it with a colon:

```
alternative_sources=SiteC:ASYNC,SiteB:SYNCMEM
```

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
```

8 SAP HANA System Replication: Operation and Maintenance

Learn how to monitor SAP HANA system replication.

How can I monitor system replication?

There are multiple ways to monitor system replication and to verify if the primary and secondary systems are in sync and are running correctly:

- Alerts on the primary and secondary systems warn you of potential problems.
- To ensure rapid takeover in the event of planned or unplanned downtime, you can check the status of the replication between the primary and the secondary system.
- You can monitor system replication using the SAP HANA cockpit, the SAP HANA studio, the `hdbnsutil` command line tool, or SQL queries.

This chapter also includes information about the network connection, how to copy or move tenants in a system replication configuration, or how to use SAP HANA system replication to update your SAP HANA systems.

Where can I find more information?

The following SAP Notes are relevant for a full understanding of the concepts described in this chapter:

SAP Notes

SAP Note	Title
1917938	Migration of the statistics server for Revision 74 or higher
2369981	Required configuration steps for authentication with HANA System Replication
2300936	Host Auto-Failover & System Replication Setup with SAP HANA extended application services, advanced model
1999880	FAQ: SAP HANA System Replication
1969700	SQL Statement Collection for SAP HANA
2494079	Near-Zero-Downtime-Upgrade to HANA 2 SPS02 or above when internal communicationSSL is used

SAP Note	Title
2081065	Troubleshooting SAP HANA Network
1984882	Using HANA System Replication for Hardware Exchange with minimum/zero Downtime
1984882	How-To Guides & Whitepapers For SAP HANA High Availability
2386973	Near Zero Downtime Upgrades for HANA Database 3-tier System Replication

Related Information

SAP HANA System Replication Details [page 155]
Alerts [page 160]
Checking the SAP HANA System Replication Status [page 166]
Monitoring System Replication [page 182]
System Replication Network Connection [page 194]
Copy or Move Tenants Within System Replication [page 201]
Copying a System Using System Replication [page 202]
Updating SAP HANA Systems with SAP HANA System Replication [page 203]
SAP HANA SQL and System Views Reference

8.1 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

Column	Description
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)
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.

Column	Description
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
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

Column	Description
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.
Max Replay Backlog Time (µs)	Specifies the maximum value of REPLAY_BACKLOG_TIME since the system startup.

Related Information

SQL and System View Reference

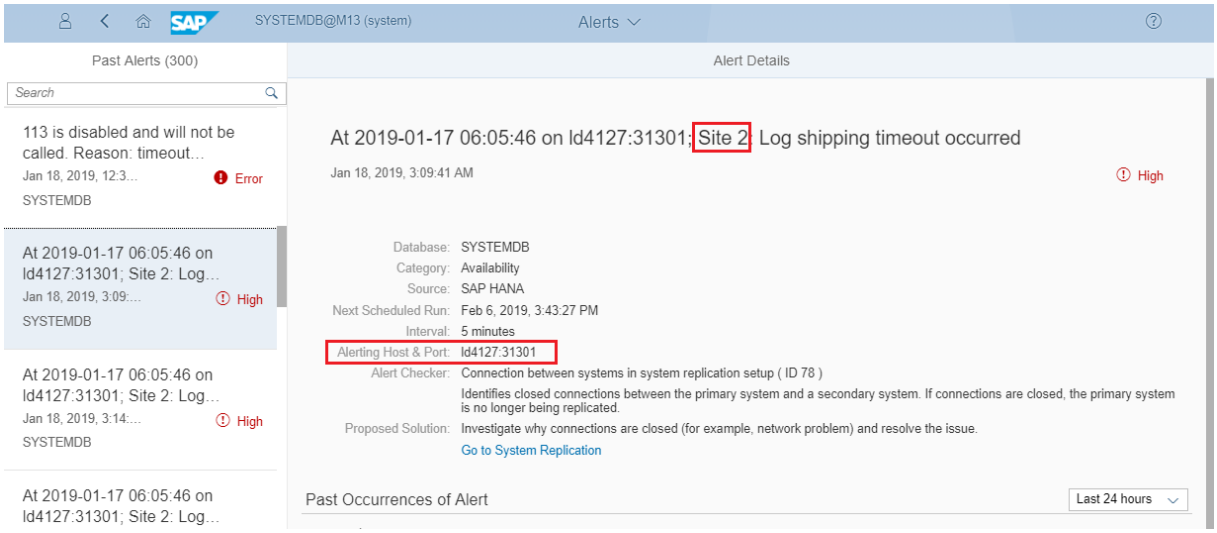
8.2 Alerts

Alerts on the primary and secondary systems warn you of potential problems.

For an overview of the alerts issued by the primary system, see *SAP HANA System Replication Alerts*.

On the system overview page of SAP HANA cockpit, choose *Show All* on the *Alerts* tile to see all alerts. Alerts occurring on the secondary system are reported on the primary system. In the SAP HANA cockpit, they are displayed as alerts and associated with the host where they occurred.

This is an example of an alert in the SAP HANA cockpit which occured on the secondary system (Id4127) and is reported on the primary system (Id4126):



In the SAP HANA studio, you can choose the *Alerts* tab to see all generated alerts.

If you want to receive an e-mail when an alert condition for all or specific checks is fulfilled, follow the steps described in *Configure E-Mail Notifications for Alerts* for SAP HANA studio or *Set Up E-mail Notification* for SAP HANA cockpit.

An alerting for the secondary systems was also established with the introduction of the so-called proxy views. Alerts occurring on the secondary hosts are shown on the primary system as alerts and associated with the host where they occurred. For more information, see *Monitoring Secondary Systems*.

Related Information

SAP HANA System Replication Alerts [page 161]
Monitoring Secondary Systems [page 163]
Configure E-Mail Notifications for Alerts
Set Up Email Notification Defaults
Proxy Schemas and Views [page 164]
Monitoring and Replicating INI File Parameter Changes [page 165]

8.2.1 SAP HANA System Replication Alerts

This is an overview of all alerts issued for the primary system.

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). **i 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*.

> **i** 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 163]
SAP HANA System Replication Configuration Parameters [page 65]
Create a Fallback Snapshot
Data and Log Compression [page 30]
SAP Note 1917938

8.2.2 Monitoring Secondary Systems

Remote SQL access on the primary system allows monitoring and reporting of the secondary system statistics.

There is a possibility to monitor the secondary system through proxy schemas and views. Proxy schemas and views are provided on the primary system. They extract the corresponding information from the 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.

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.

> **i 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.

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.

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.

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.
- When such a proxy view is queried and the secondary system is not started, no results are shown without the report of an SQL error.
- 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

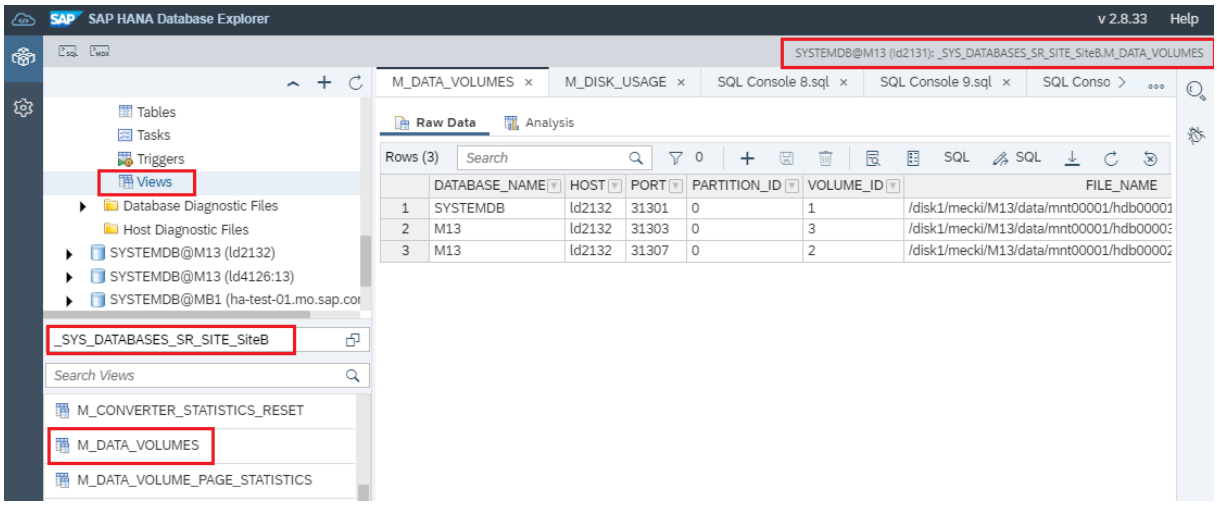
Alert Details

Proxy Schemas and Views [page 164]

8.2.2.1 Proxy Schemas and Views

You can monitor the secondary system through proxy schemas and views.

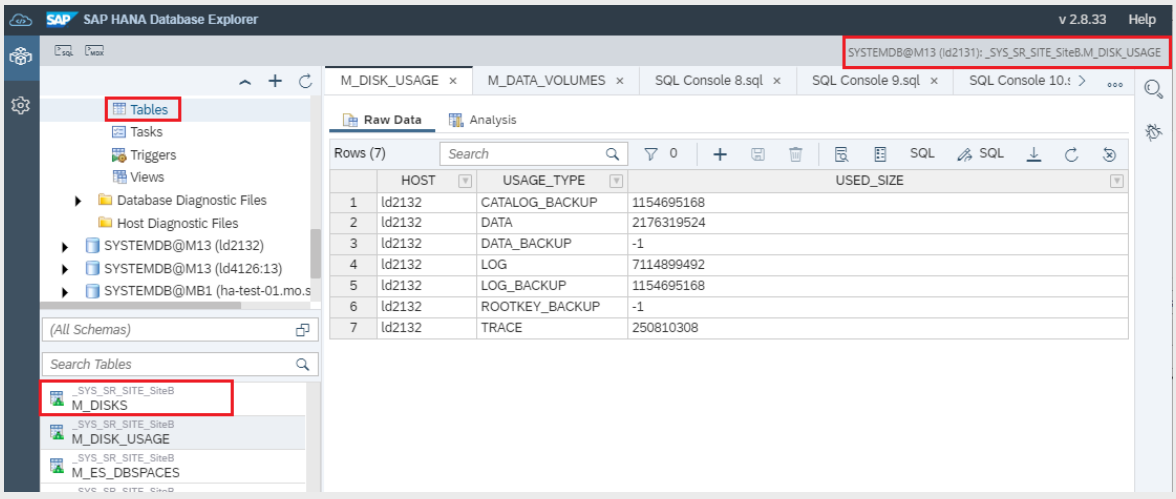
To see the proxy views of the secondary system's monitoring views, you can use the SAP HANA cockpit. On the system overview page of the primary system, choose *Execute SQL* to open the SAP HANA Database Explorer. Then open the primary's catalog and go to the corresponding schema. This is a schema example:



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";
```

8.2.3 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)
SAP HANA Configuration Parameter Reference

8.3 Checking the SAP HANA System Replication Status

To ensure rapid takeover in the event of planned or unplanned downtime, you can monitor the status of replication between the primary system and the secondary system.

There are several ways to gather information about the overall status of the sites and of the system replication:

- With the `landscapeHostConfiguration.py` script
  This script provides information only about the state of one SAP HANA database. A returned error state (for example, return code 1) could indicate that a takeover to the secondary should be considered.
- With the `systemReplicationStatus.py` script
  This script shows whether the secondary systems are in sync or not. This provides more confidence if a takeover is justified.
- With the `getTakeoverRecommendation` script
  This script shows the takeover recommendation based on the current system state.

  ### → Recommendation

  Rather than calling both `landscapeHostConfiguration.py` and `systemReplicationStatus.py` manually and calculate the required action based on return codes, you can use the `getTakeoverRecommendation.py` script.
- With HDB console
  Using the HDB console you can check the system replication status on all hosts and for all services.

- With predefined SQL statements

Related Information

Checking the Status with landscapeHostConfiguration.py [page 167]
Checking the Status with systemReplicationStatus.py [page 170]
Checking the Status with getTakeoverRecommendation.py [page 172]
Example: Checking the Status on the Primary and Secondary Systems [page 173]
Checking the Status with the HDB Console [page 177]
Checking the Status with Predefined SQL Statements [page 179]
Monitoring Status and Resource Usage of System Components

8.3.1 Checking the Status with landscapeHostConfiguration.py

Check the status of the primary system using `landscapeHostConfiguration.py`.

Check the overall status of the primary system using as <sid>adm user the `landscapeHostConfiguration.py` script located in `$DIR_INSTANCE/../exe/python_support`.

```
<sid>adm># python $DIR_INSTANCE/exe/python_support/landscapeHostConfiguration.py
| Host  | Host  | Host  | ... | NameServer | NameServer  | ...
|       | Active| Status|     | Config Role| Actual Role |
| ----- | ------| ------| --- | ---------- | ----------- | ----
| host1 | yes   | ok    | ... | master 1   | master      | ...
| host2 | yes   | ok    | ... | master 2   | slave       | ...
overall host status: ok
```

Use the `--sapcontrol=1` parameter, if you require a reliable and parsable output.

```
<sid>adm># python $DIR_INSTANCE/exe/python_support/landscapeHostConfiguration.py
--sapcontrol=1
SAPCONTROL-OK: <begin>
host/ld2131/hostActualRoles=worker
host/ld2131/removeStatus=
host/ld2131/nameServerConfigRole=master 1
host/ld2131/failoverStatus=
host/ld2131/hostConfigRoles=worker
host/ld2131/failoverActualGroup=default
host/ld2131/storageConfigPartition=1
host/ld2131/host=ld2131
host/ld2131/indexServerConfigRole=worker
host/ld2131/failoverConfigGroup=default
host/ld2131/storageActualPartition=1
host/ld2131/indexServerActualRole=master
host/ld2131/nameServerActualRole=master
host/ld2131/hostActive=yes
host/ld2131/hostStatus=ok
host/ld2131/storagePartition=0
host/ld2132/hostActualRoles=worker
host/ld2132/removeStatus=
host/ld2132/nameServerConfigRole=master 3
```

```
host/ld2132/failoverStatus=
host/ld2132/hostConfigRoles=worker
host/ld2132/failoverActualGroup=default
host/ld2132/storageConfigPartition=2
host/ld2132/host=ld2132
host/ld2132/indexServerConfigRole=worker
host/ld2132/failoverConfigGroup=default
host/ld2132/storageActualPartition=2
host/ld2132/indexServerActualRole=slave
host/ld2132/nameServerActualRole=slave
host/ld2132/hostActive=yes
host/ld2132/hostStatus=ok
host/ld2133/hostActualRoles=standby
host/ld2133/removeStatus=
host/ld2133/nameServerConfigRole=master 2
host/ld2133/failoverStatus=
host/ld2133/hostConfigRoles=standby
host/ld2133/failoverActualGroup=default
host/ld2133/storageConfigPartition=0
host/ld2133/host=ld2133
host/ld2133/indexServerConfigRole=standby
host/ld2133/failoverConfigGroup=default
host/ld2133/storageActualPartition=0
host/ld2133/indexServerActualRole=standby
host/ld2133/nameServerActualRole=slave
host/ld2133/hostActive=yes
host/ld2133/hostStatus=ignore
overall_status=ok
SAPCONTROL-OK: <end>
```

> **i** Note
>
> This script provides information about the state of one SAP HANA database only. A takeover is only recommended when the return code from the script is 1 (error).

The return codes of the script are:

Return code	Description
0	Fatal Internal script error, the state could not be determined
1	Error
2	Warning
3	Info
4	OK

```
Status Error and Return Code=1:
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> python
landscapeHostConfiguration.py
| Host | Host | Host | Failover | Remove | Storage | Storage | Failover |
Failover | NameServer | NameServer | IndexServer | IndexServer | Host | Host |
Worker | Worker |
| | Active | Status | Status | Status | Config | Actual | Config | Actual |
Config | Actual | Config | Actual | Config | Actual | Config | Actual |
| | | | | | Partition | Partition | Group | Group | Role | Role | Role | Role |
Roles | Roles | Groups | Groups |
```

```
| ------ | ------ | ------ | -------- | ------ | --------- | --------- |
-------- | -------- | ---------- | ---------- | ----------- | ----------- |
------- | ------- | ------- | ------- |
| ld2131 | yes | ok | | | 1 | 1 | default | default | master 1 | master | worker
| master | worker | worker | default | default |
| ld2132 | no | error | | | 2 | 2 | default | default | master 2 | slave |
worker | slave | worker | worker | default | default |
| ld2133 | no | ignore | | | 0 | 0 | default | default | master 3 | slave |
standby | standby | standby | standby | default | - |
overall host status: error
Return Code
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> echo $?
1
```

Status Warning and Return Code=2

```
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> python
landscapeHostConfiguration.py
| Host | Host | Host | Failover | Remove | Storage | Storage | Failover |
Failover | NameServer | NameServer | IndexServer | IndexServer | Host | Host |
Worker | Worker |
| | Active | Status | Status | Status | Config | Actual | Config | Actual |
Config | Actual | Config | Actual | Config | Actual | Config | Actual |
| | | | | | Partition | Partition | Group | Group | Role | Role | Role | Role |
Roles | Roles | Groups | Groups |
| ------ | ------ | ------- | -------------- | ------ | --------- | --------- |
-------- | -------- | ---------- | ---------- | ---------- | ---------- |
------- | ------- | ------- | ------- |
| ld2131 | yes | ok | | | 1 | 1 | default | default | master 1 | master | worker
| master | worker | worker | default | default |
| ld2132 | no | warning | waiting 30 sec | | 2 | 2 | default | default | master
2 | slave | worker | slave | worker | worker | default | default |
| ld2133 | yes | ignore | | | 0 | 0 | default | default | master 3 | slave |
standby | standby | standby | standby | default | - |
overall host status: warning
Return Code
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> echo $?
2
```

Status Info and Return Code=3:

```
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> python
landscapeHostConfiguration.py
| Host | Host | Host | Failover | Remove | Storage | Storage | Failover |
Failover | NameServer | NameServer | IndexServer | IndexServer | Host | Host |
Worker | Worker |
| | Active | Status | Status | Status | Config | Actual | Config | Actual |
Config | Actual | Config | Actual | Config | Actual | Config | Actual |
| | | | | | Partition | Partition | Group | Group | Role | Role | Role | Role |
Roles | Roles | Groups | Groups |
| ------ | ------ | ------- | -------- | ------ | --------- | --------- |
-------- | -------- | ---------- | ---------- | ---------- | ---------- |
------- | ------- | ------- | ------- |
| ld2131 | yes | ok | | | 1 | 1 | default | default | master 1 | master | worker
| master | worker | worker | default | default |
| ld2132 | yes | info | | | 2 | 0 | default | default | master 2 | slave |
worker | standby | worker | standby | default | - |
| ld2133 | yes | info | | | 0 | 2 | default | default | master 3 | slave |
standby | slave | standby | worker | default | default |
overall host status: info
Return Code
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> echo $?
3
```

Status OK and Return Code=4

```
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> python
landscapeHostConfiguration.py
```

```
| Host | Host | Host | Failover | Remove | Storage | Storage | Failover |
Failover | NameServer | NameServer | IndexServer | IndexServer | Host | Host |
Worker | Worker |
| | Active | Status | Status | Status | Config | Actual | Config | Actual |
Config | Actual | Config | Actual | Config | Actual | Config | Actual |
| | | | | | Partition | Partition | Group | Group | Role | Role | Role | Role |
Roles | Roles | Groups | Groups |
| ------ | ------ | ------ | -------- | ------ | --------- | --------- |
-------- | -------- | ---------- | ---------- | ---------- | ---------- |
------- | ------- | ------- | ------- |
| ld2131 | yes | ok | | | 1 | 1 | default | default | master 1 | master | worker
| master | worker | worker | default | default |
| ld2132 | yes | ok | | | 2 | 2 | default | default | master 2 | slave | worker
| slave | worker | worker | default | default |
| ld2133 | yes | ignore | | | 0 | 0 | default | default | master 3 | slave |
standby | standby | standby | standby | default | - |
overall host status: ok
Return Code
dx7adm@ld2131:/usr/sap/DX7/HDB07/exe/python_support> echo $?
4
```

In the event of a network split, a so called split brain scenario, the script cannot tell if the instance in the other half of the network is fully functional. Therefore, a takeover decision should not be based on this script alone.

Related Information

Example: Checking the Status on the Primary and Secondary Systems [page 173]

8.3.2 Checking the Status with systemReplicationStatus.py

Check the status of system replication using the `systemReplicationStatus.py` script.

`systemReplicationStatus.py` shows whether the secondary systems are in sync or not. This provides more confidence if a takeover is justified. If system replication was never in sync or is outdated, unexpected loss of data might occur.

Check the overall status of the system replication using as <sid>adm user the `systemReplicationStatus.py` script located in `$DIR_INSTANCE/../exe/python_support`.

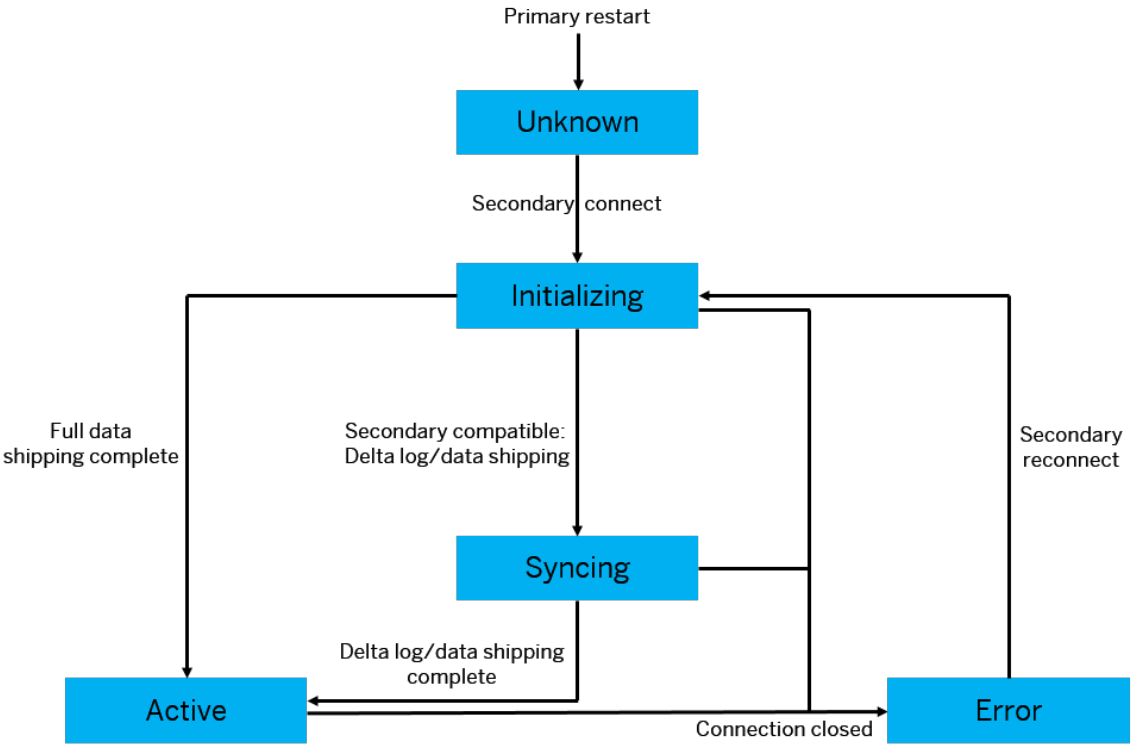
```
<sid>adm># python $DIR_INSTANCE/exe/python_support/systemReplicationStatus.py
| Host   | Service Name     | Site Name | Secondary | ...| Replication|
|        |                  |           | Host      |    | Status     |
| ------ | ---------------- | --------- | --------- | ---| -----------|
| ld7805 | indexserver      | WALLDORF  | ld8475    | ...| ACTIVE     |
| ld8513 | statisticsserver | WALLDORF  | ld8476    |    | ACTIVE     |
| ld8513 | xsengine         | WALLDORF  | ld8476    |    | ACTIVE     |
| ld8513 | nameserver       | WALLDORF  | ld8476    |    | ACTIVE     |
| ld8513 | indexserver      | WALLDORF  | ld8476    |    | ACTIVE     |
| ld8559 | indexserver      | WALLDORF  | NOT MAPPED|    |            |
status system replication site "2": ACTIVE
status system replication site "3": ACTIVE
overall system replication status: ACTIVE
Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
mode: PRIMARY
site id: 1
```

```
site name: WALLDORF
```

The return codes of the script are:

Return code	Description
10	No System Replication
11	Error Error occurred on the connection. Additional details on the error can be found in REPLICATION_STATUS_DETAILS.
12	Unknown The secondary system did not connect to primary since last restart of the primary system.
13	Initializing Initial data transfer is in progress. In this state, the secondary is not usable at all.
14	Syncing The secondary system is syncing again (for example, after a temporary connection loss or restart of the secondary).
15	Active Initialization or sync with the primary is complete and the secondary is continuously replicating. No data loss will occur in SYNC mode.

Use the following state-transition graphic to understand the return codes of the script.



Related Information

Monitoring and Replicating INI File Parameter Changes [page 165]
M_SERVICE_REPLICATION System View [page 257]
Example: Checking the Status on the Primary and Secondary Systems [page 173]
SAP HANA System Replication Alerts [page 161]
SAP HANA SQL and System Views Reference

8.3.3 Checking the Status with getTakeoverRecommendation.py

Check the status of system replication using the `getTakeoverRecommendation.py` script.

The `getTakeoverRecommendation` script shows the takeover recommendation based on the current system state. However, when the primary system faces any error situation, the system replication status cannot be determined anymore. Thus, the previous state should be saved and compared with the current state.

The script provides the following recommendations as takeover status based on the results of the other two scripts `landscapeHostConfiguration.py` and `systemReplicationStatus.py`. It also provides an overall status and a return code to match the overall host status.

landscapeHostConfiguration	systemReplicationStatus	Takeover Status	Reason for the takeover recommendation
Error/Fatal/Warning	NoHsr/Error/ Unknown/Initializing/Syncing/Active	Required	Primary system has errors
OK/Info/Ignore	NoHsr/Error/ Unknown/Initializing/Syncing	Cannot decide	Unknown system replication status
OK/Info/Ignore	Active	Possible	Primary system is up and system replication is in sync

This is a sample implementation of a python script that uses `getTakeoverRecommendation` to act as a minimalistic cluster manager.

Use the `--sapcontrol=1` parameter, if you require a reliable and parsable output.

```
import time
import subprocess
from getTakeoverRecommendation import TakeoverDecision
def main():
    wasInSync = False
    while True:
        recommendation =
subprocess.call(["python","getTakeoverRecommendation.py","--sapcontrol=1"])
        if not wasInSync and recommendation is TakeoverDecision.Required:
            print ("Primary defect & no sync => NO TAKEOVER")
        if wasInSync and recommendation is TakeoverDecision.Required:
            print ("Primary defect & sync => TAKEOVER")
        nowInSync = recommendation is TakeoverDecision.Possible
```

```
        wasInSync = nowInSync
```

The output depends on the previous state with the result of the current call of `getTakeoverRecommendation`. If no sync state is reached, a takeover is not advised. But once the systems are in sync, the next error of the primary system will suggest a takeover. Any subsequent negative return value will reset the sync state as it is no longer ensured that the replicated data is current.

Related Information

Example: Checking the Status on the Primary and Secondary Systems [page 173]

8.3.4 Example: Checking the Status on the Primary and Secondary Systems

Learn how to check the status on the primary and secondary systems with `landscapeHostConfiguration.py`, `systemReplicationStatus.py`, and `getTakeoverRecommendation`.

The examples below offer an overview of the outputs for the three scripts with and without "--sapcontrol=1". Use `echo $?` to receive the return code of the last executed script.

Primary System

```
m13adm@ld2131:/usr/sap/M13/HDB13/exe/python_support> python
systemReplicationStatus.py
| Database | Host   | Port  | Service Name | Volume ID | Site ID | Site Name |
Secondary | Secondary | Secondary | Secondary | Secondary     | Replication |
Replication | Replication    |
|          |        |       |              |           |         |           |
Host      | Port      | Site ID   | Site Name | Active Status | Mode        |
Status      | Status Details |
| -------- | ------ | ----- | ------------ | --------- | ------- | --------- |
--------- | --------- | --------- | --------- | ------------- | ----------- |
----------- | -------------- |
| SYSTEMDB | ld2131 | 31301 | nameserver   |         1 |       1 | SiteA     |
ld2132    |     31301 |         2 | SiteB2    | YES           | SYNCMEM     |
ACTIVE      |                |
| M13      | ld2131 | 31307 | xsengine     |         2 |       1 | SiteA     |
ld2132    |     31307 |         2 | SiteB2    | YES           | SYNCMEM     |
ACTIVE      |                |
| M13      | ld2131 | 31303 | indexserver  |         3 |       1 | SiteA     |
ld2132    |     31303 |         2 | SiteB2    | YES           | SYNCMEM     |
ACTIVE      |                |
status system replication site "2": ACTIVE
overall system replication status: ACTIVE
Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
mode: PRIMARY
site id: 1
site name: SiteA
```

```
================================================================================
================================================
m13adm@ld2131:/usr/sap/M13/HDB13/exe/python_support> python
systemReplicationStatus.py --sapcontrol=1
SAPCONTROL-OK: <begin>
service/ld2131/31301/SHIPPED_LOG_POSITION_TIME=2019-01-04 17:49:03.563462
service/ld2131/31301/LAST_LOG_POSITION_TIME=2019-01-04 17:49:03.563462
service/ld2131/31301/SHIPPED_FULL_REPLICA_DURATION=6191118
service/ld2131/31301/SHIPPED_LAST_DELTA_REPLICA_START_TIME=-
service/ld2131/31301/SHIPPED_FULL_REPLICA_SIZE=1577443328
service/ld2131/31301/SITE_ID=1
service/ld2131/31301/LAST_LOG_POSITION=79676672
service/ld2131/31301/SECONDARY_ACTIVE_STATUS=YES
service/ld2131/31301/LAST_SAVEPOINT_LOG_POSITION=79676418
service/ld2131/31301/FULL_SYNC=DISABLED
service/ld2131/31301/OPERATION_MODE=logreplay_readaccess
service/ld2131/31301/SHIPPED_LAST_FULL_REPLICA_START_TIME=2018-12-18
16:08:45.743753
service/ld2131/31301/LAST_SAVEPOINT_VERSION=6633
...
service/ld2131/31303/SITE_NAME=SiteA
service/ld2131/31303/SECONDARY_SITE_NAME=SiteB2
service/ld2131/31303/REPLAYED_LOG_POSITION_TIME=2019-01-04 17:49:07.122280
service/ld2131/31303/SHIPPED_LAST_FULL_REPLICA_END_TIME=2018-12-18
16:09:07.125423
service/ld2131/31303/CREATION_TIME=2018-12-18 13:37:49.825044
service/ld2131/31303/HOST=ld2131
service/ld2131/31303/SHIPPED_SAVEPOINT_VERSION=71
service/ld2131/31303/SECONDARY_HOST=ld2132
service/ld2131/31303/VOLUME_ID=3
service/ld2131/31303/SHIPPED_LAST_FULL_REPLICA_SIZE=1678131200
service/ld2131/31303/SHIPPED_LOG_BUFFERS_SIZE=11519025152
service/ld2131/31303/REPLICATION_MODE=SYNCMEM
service/ld2131/31303/DATABASE=M13
service/ld2131/31303/REPLAYED_LOG_POSITION=264306432
service/ld2131/31303/SECONDARY_RECONNECT_COUNT=1
service/ld2131/31303/SHIPPED_SAVEPOINT_START_TIME=2018-12-18 16:09:00.543785
service/ld2131/31303/SECONDARY_PORT=31303
service/ld2131/31303/SHIPPED_SAVEPOINT_LOG_POSITION=84321794
service/ld2131/31303/REPLICATION_STATUS=ACTIVE
service/ld2131/31303/SECONDARY_CONNECT_TIME=2018-12-18 16:09:00.513111
service/ld2131/31303/SHIPPED_LOG_BUFFERS_COUNT=2395097
service/ld2131/31303/SECONDARY_SITE_ID=2
site/2/SITE_NAME=SiteB2
site/2/SOURCE_SITE_ID=1
site/2/REPLICATION_MODE=SYNCMEM
site/2/REPLICATION_STATUS=ACTIVE
overall_replication_status=ACTIVE
site/1/REPLICATION_MODE=PRIMARY
site/1/SITE_NAME=SiteA
local_site_id=1
SAPCONTROL-OK: <end>
================================================================================
================================================
m13adm@ld2131:/usr/sap/M13/HDB13/exe/python_support> echo $?
15
********************************************************************************
********************************************
********************************************************************************
********************************************
m13adm@ld2131:/usr/sap/M13/HDB13/exe/python_support> python
landscapeHostConfiguration.py
| Host   | Host   | Host   | Failover | Remove | Storage   | Storage   |
Failover | Failover | NameServer | NameServer | IndexServer | IndexServer |
Host   | Host   | Worker  | Worker  |
|        | Active | Status | Status   | Status | Config    | Actual    |
Config   | Actual   | Config     | Actual     | Config      | Actual      |
Config | Actual | Config  | Actual  |
```

```
|        |        |        |          |        | Partition | Partition |
Group    | Group    | Role       | Role       | Role        | Role        |
Roles  | Roles  | Groups  | Groups  |
| ------ | ------ | ------ | -------- | ------ | --------- | --------- |
-------- | -------- | ---------- | ---------- | ----------- | ----------- |
------ | ------ | ------- | ------- |
| ld2131 | yes    | ok     |          |        |         1 |         1 |
default  | default  | master 1   | master     | worker      | master      |
worker | worker | default | default |
================================================================================
================================================
overall host status: ok
m13adm@ld2131:/usr/sap/M13/HDB13/exe/python_support> python
landscapeHostConfiguration.py --sapcontrol=1
SAPCONTROL-OK: <begin>
hostActualRoles=worker
removeStatus=
nameServerConfigRole=master 1
failoverStatus=
hostConfigRoles=worker
failoverActualGroup=default
storageConfigPartition=1
host=ld2131
indexServerConfigRole=worker
failoverConfigGroup=default
storageActualPartition=1
indexServerActualRole=master
nameServerActualRole=master
hostActive=yes
workerActualGroups=default
workerConfigGroups=default
hostStatus=ok
storagePartition=1
SAPCONTROL-OK: <end>
================================================================================
================================================
m13adm@ld2131:/usr/sap/M13/HDB13/exe/python_support> echo $?
4
```

Secondary System

```
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> python
systemReplicationStatus.py
this system is either not running or not the primary system replication site
Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
mode: SYNCMEM
site id: 2
site name: SiteB2
active primary site: 1
primary masters: ld2131
================================================================================
================================================
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> python
systemReplicationStatus.py --sapcontrol=1
SAPCONTROL-OK: <begin>
site/2/REPLICATION_MODE=SYNCMEM
site/2/SITE_NAME=SiteB2
site/2/SOURCE_SITE_ID=1
site/2/PRIMARY_MASTERS=ld2131
local_site_id=2
SAPCONTROL-OK: <end>
```

```
================================================================================
================================================
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> echo $?
12
********************************************************************************
********************************************
********************************************************************************
********************************************
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> python
landscapeHostConfiguration.py
| Host   | Host   | Host   | Failover | Remove | Storage   | Storage   |
Failover | Failover | NameServer | NameServer | IndexServer | IndexServer |
Host   | Host   | Worker  | Worker  |
|        | Active | Status | Status   | Status | Config    | Actual    |
Config   | Actual   | Config     | Actual     | Config      | Actual      |
Config | Actual | Config  | Actual  |
|        |        |        |          |        | Partition | Partition |
Group    | Group    | Role       | Role       | Role        | Role        |
Roles  | Roles  | Groups  | Groups  |
| ------ | ------ | ------ | -------- | ------ | --------- | --------- |
-------- | -------- | ---------- | ---------- | ---------- | ---------- |
------ | ------ | ------- | ------- |
| ld2132 | yes    | ok     |          |        |         1 |         1 |
default  | default  | master 1   | master     | worker      | master      |
worker | worker | default | default |
overall host status: ok
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> python
landscapeHostConfiguration.py --sapcontrol=1
SAPCONTROL-OK: <begin>
hostActualRoles=worker
removeStatus=
nameServerConfigRole=master 1
failoverStatus=
hostConfigRoles=worker
failoverActualGroup=default
storageConfigPartition=1
host=ld2132
indexServerConfigRole=worker
failoverConfigGroup=default
storageActualPartition=1
indexServerActualRole=master
nameServerActualRole=master
hostActive=yes
workerActualGroups=default
workerConfigGroups=default
hostStatus=ok
storagePartition=1
SAPCONTROL-OK: <end>
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> echo $?
4
********************************************************************************
********************************************
********************************************************************************
********************************************
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> python
getTakeoverRecommendation.py
================================================================================
================================================
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> python
getTakeoverRecommendation.py --sapcontrol=1
SAPCONTROL-OK: <begin>
landscapeHostConfiguration/code=4
landscapeHostConfiguration/description=ok
getTakeoverRecommendation/code=0
getTakeoverRecommendation/description=cannot decide
SAPCONTROL-OK: <end>
================================================================================
================================================
```

```
m13adm@ld2132:/usr/sap/M13/HDB13/exe/python_support> echo $?
0
```

8.3.5 Checking the Status with the HDB Console

Use the HDB console to check the system replication status on all hosts and for all services.

Using the HDB console for a status check can be especially useful in an ASYNC replication. Yon can see additional information currently not shown by the system view, because in this mode the primary systen does not wait for acknowledgement upon arrival of the shipped redo log buffer. In this case, you have to check the current log position on the secondary system on each node and for each persistency relevant service. The command is to be executed per service using the `hdbcons -p <servicePID> "replication into"` parameter.

In the following example, you can get information about the used replication and operation modes (mode, operation_mode). You can also see which IP address is used for data and log transfer (Log connection and Data connection) and – since this system replication example is running with operation mode `logreplay` – you can see how far the log replay is hanging behind the shipped log on this secondary (the delta between shippedLogPos and replayFinishLogPos). For all services the replication status should be *ReplicationStatus_Active*.

```
<sid>adm># hdbcons -p 54321 "replication info"
SAP HANA DB Management Client Console (type '\?' to get help for client commands)
Try to open connection to server process 'hdbindexserver' on system 'M19',
instance '19'
SAP HANA DB Management Server Console (type 'help' to get help for server
commands)
Executable: hdbindexserver (PID: 66110)
[OK]
--
listing default statistics for volume 3
System Replication Secondary Information
========================================
System Replication Secondary Configuration
[system_replication] site_id = 2
[system_replication] site_name = SiteA
 [system_replication] mode = sync
 [system_replication] operation_mode = logreplay
[system_replication] datashipping_min_logsize_threshold = 5368709120
[system_replication] datashipping_min_time_interval = 600
[system_replication] reconnect_time_interval = 30
[system_replication] enable_log_compression = false
[system_replication] preload_column_tables = true
[system_replication] ensure_backup_history = true
[system_replication] enable_ssl = off
[system_replication] keep_old_style_alert = false
[system_replication] enable_log_retention = 1
[system_replication] logshipping_max_retention_size = 1048576
Last Primary Host: ld2133
Last Primary Port: 32003
Log Connection
- ptr : 0x00007fd58931a400
- channel : NetworkChannel FD 25 [0x00007fd5ad064a98] {refCnt=3, idx=2}
10.96.4.20/65117_tcp->10.96.4.22/32003_tcp Connected,[r---]
- mode : ReplicationMode_Synchronous
- logSinceLastBackup : 663552 bytes
- timeSinceLastBackup : 67431655 microseconds
Data Connection
- ptr : 0x00007fd589315000
```

```
- channel : NetworkChannel FD 31 [0x00007fd5ad064c58] {refCnt=2, idx=3}
10.96.4.20/65118_tcp->10.96.4.22/32003_tcp Connected,[----]
Secondary Statistics
- Creation Timestamp : 08.12.2015-14.25.27 (1449584727282603)
- Last Reset Timestamp : 08.12.2015-14.25.27 (1449584727282603)
- Statistic Reset Count : 0
 - ReplicationMode : sync
 - OperationMode : logreplay
 - ReplicationStatus : ReplicationStatus_Active
- ReplicationStatusDetails :
 - ReplicationFullSync : DISABLED
 - shippedLogPos : 0x641cbb00
- shippedLogPosTimestamp : 08.12.2015-14.59.17 (1449586757965706)
- sentLogPos : 0x0
- sentLogPosTimestamp : 01.01.1970-00.00.00 (0)
- shippedLogBuffersCount : 11241
- shippedLogBuffersSize : 8335585280 bytes
- shippedLogBuffersSizeUsed : 8309875456 bytes (99.69%)
- shippedLogBuffersSizeNet : 8309875456 bytes (99.69%)
- shippedLogBufferDuration : 0 microseconds
- shippedLogBufferDurationMin : 0 microseconds
- shippedLogBufferDurationMax : 0 microseconds
- shippedLogBufferDurationSend : 0 microseconds
- shippedLogBufferDurationComp : 0 microseconds
- shippedLogBufferThroughput : 0.00 bytes/s
 - replayFinishLogPos : 0x641cbb00
- replayFinishLogPosTimestamp : 08.12.2015-14.59.17 (1449586757965706)
- replayStartLogPos : 0x641cbb00
- replayPushLogPos : 0x641cbb00
- replayRetentionLogPos : 0x62a66fcb
- replayStepCount : 61709
- replayLogSize : 8335581056 bytes
- replayDuration : 111608005 microseconds
- shippedSavepointVersion : 2252
- shippedSavepointLogPos : 0x5c595f82
- shippedSavepointTimestamp : 08.12.2015-14.25.28 (1449584728678668)
- shippedFullBackupCount : 1
- shippedFullBackupSize : 17884512256 bytes
- shippedFullBackupSizeNet : 17884512256 bytes (100.00%)
- shippedFullBackupDuration : 81098893 microseconds
- shippedFullBackupDurationComp : 0 microseconds
- shippedFullBackupThroughput : 220527205.67 bytes/s
- shippedLastFullBackupSize : 17884512256 bytes
- shippedLastFullBackupSizeNet : 17884512256 bytes (100.00%)
- shippedLastFullBackupStart : 08.12.2015-14.25.28 (1449584728678668)
- shippedLastFullBackupEnd : 08.12.2015-14.26.49 (1449584809777561)
- shippedLastFullBackupDuration : 81098893 microseconds
- shippedDeltaBackupCount : 0
- shippedDeltaBackupSize : 0 bytes
- shippedDeltaBackupSizeNet : 0 bytes (-nan%)
- shippedDeltaBackupDuration : 0 microseconds
- shippedDeltaBackupDurationComp : 0 microsecond
- shippedDeltaBackupThroughput : 0.00 bytes/s
- shippedLastDeltaBackupSize : 0 bytes
- shippedLastDeltaBackupSizeNet : 0 bytes (-nan%)
- shippedLastDeltaBackupStart : not set
- shippedLastDeltaBackupEnd : not set
- shippedLastDeltaBackupDuration : 0 microseconds
- Secondary sync'ed via Log Count : 0
- syncLogCount : 0
- syncLogSize : 0 bytes
- Secondary Backup History : complete
- shippedMissingLogCount : 0
- shippedMissingLogSize : 0 bytes
- backlogSize : 0 bytes
- backlogTime : 0 microseconds
- backlogSizeMax : 0 bytes
- backlogTimeMax : 0 microseconds
```

```
- Secondary Log Connect time : 08.12.2015-14.25.27 (1449584727296916)
- Secondary Data Connect time : 08.12.2015-14.25.27 (1449584727491743)
- Secondary Log Close time : not set
- Secondary Data Close time : not set
- Secondary Log Reconnect Count : 0
- Secondary Log Failover Count : 0
- Secondary Data Reconnect Count : 0
- Secondary Data Failover Count : 0
[OK]
--
[EXIT]
--[BYE]
```

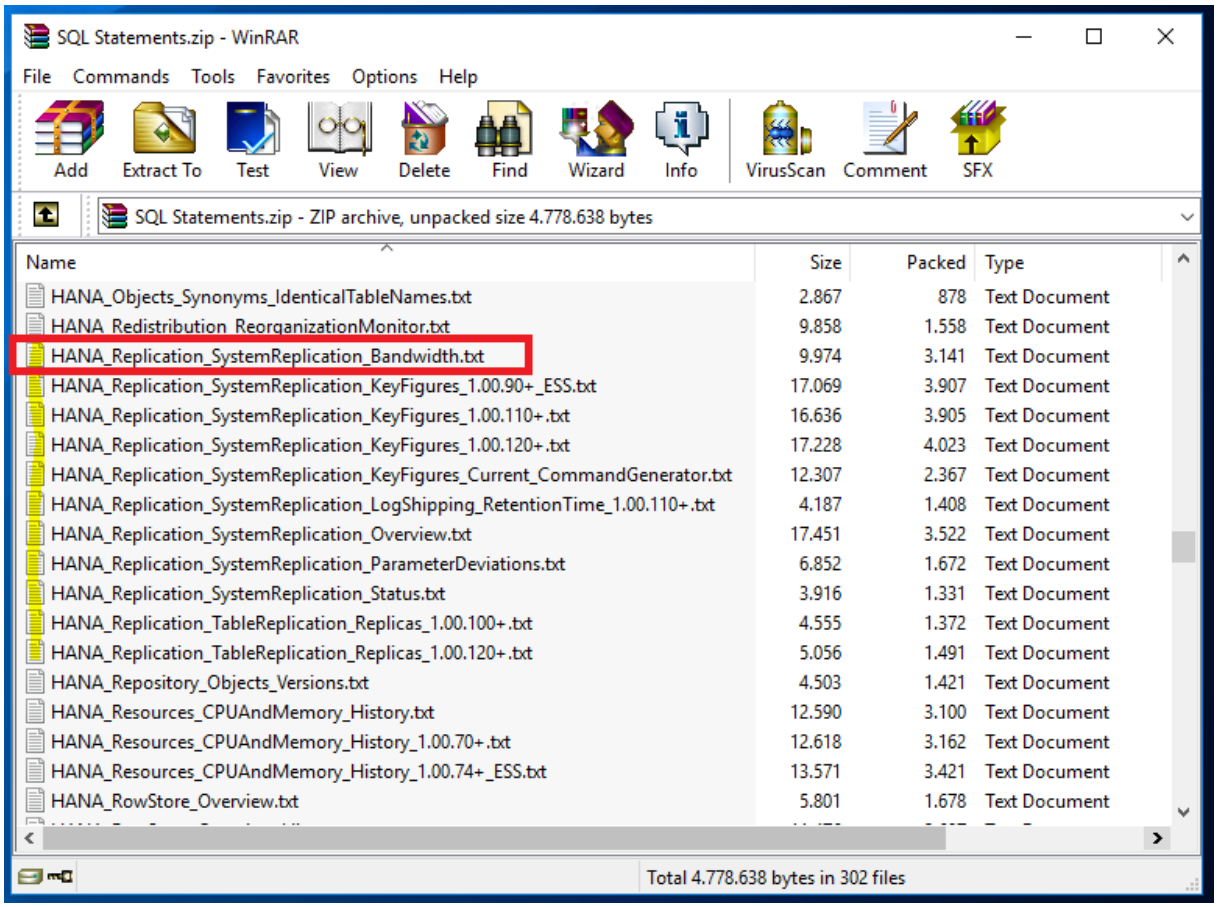
Related Information

SAP Note 1999880

8.3.6 Checking the Status with Predefined SQL Statements

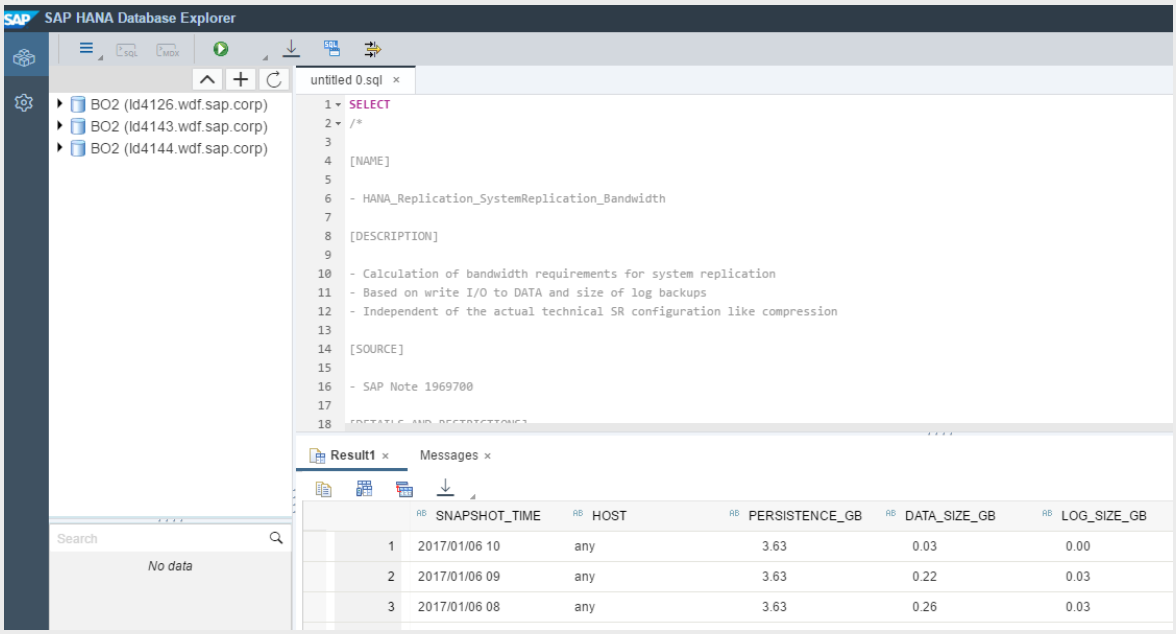
Use predefined SQL statements to check the system replication status.

Predefined SQL statements are attached to the *SAP Note 1969700*. You can copy and execute the statements contained in the text files in the SAP HANA Database Explorer or the SAP HANA studio:

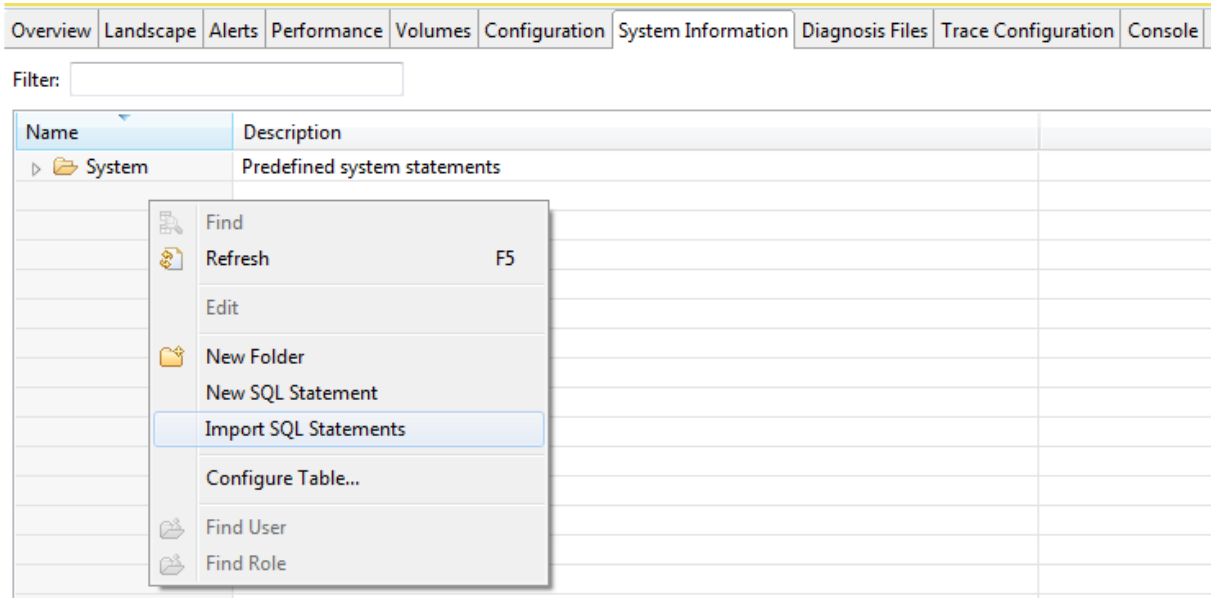


Example

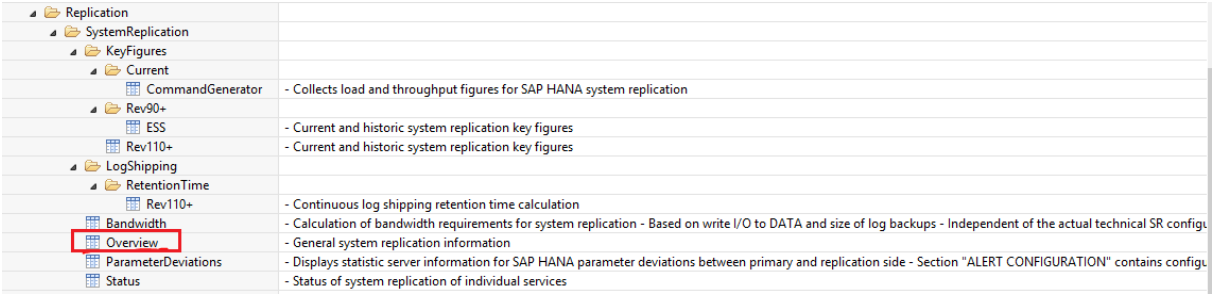
Copy and execute the *HANA_Replication_SystemReplication_Bandwidth.text* in the SAP HANA Database Explorer:



In the SAP HANA studio, go for the primary system to the *System Information* tab and right-click ▶ *Name* ▶ *Import SQL Statements* ▶:



Import the SQL statements downloaded from the SAP Note and right-click on the statements, then choose ▶ *Replication* ▶ *Overview* ▶.



This will open information about system replication and the replication state for each service:

Raw Data | Distinct values

Filter pattern

103 rows retrieved - 116 ms

REPLICATION_PATH	HOSTS	SERV...	KEY	VALUE
SITEA -> SITEB	ld2131 -> ld2132	indexserver	Replication mode	SYNCMEM
			Secondary connect time	2014/05/02 10:50:39
			Days since secondary connect time	0.02
			Used persistence size (GB)	4.98
			Log backup size / day (GB)	1.50
			Local log buffer write size (MB)	4623.18
			Shipped log buffer size (MB)	3593.53
			Avg. local log buffer write size (KB)	803.62
			Avg. shipped log buffer size (KB)	819.55
			Avg. local log buffer write time (ms)	6.60
			Avg. log buffer shipping time (ms)	6.71
			Local log buffer write throughput (MB/s)	118.86
			Log buffer shipping throughput (MB/s)	119.13
			Initial data shipping size (MB)	1122.84
			Initial data shipping time (s)	4.28
			Last delta data shipping size (MB)	1.01
			Last delta data shipping time (s)	0.00
			Delta data shipping size (MB)	3.24
			Delta data shipping time (s)	0.36
			Delta data shipping throughput (MB/s)	8.81
			Delta data shipping size / day (MB)	129.42
SITEA -> SITEB	ld2131 -> ld2132	nameserver	Replication mode	SYNCMEM
			Secondary connect time	2014/05/02 10:50:34
			Days since secondary connect time	0.02
			Used persistence size (GB)	0.00
			Log backup size / day (GB)	0.00

In a synchronous replication, it is interesting to compare *Local log buffer write throughput (MB/s)* with *Log buffer shipping throughput (MB/s)*. If these values differ too much, they could indicate network problems or a problem with the I/O on the secondary system.

For more information about how to interpret check results, see *SAP Note 1999993*.

8.4 Monitoring System Replication

You can monitor system replication using different tools.

- SAP HANA cockpit
  For more information, see *Monitoring System Replication with the SAP HANA Cockpit* and *Example: Monitoring System Replication with the SAP HANA Cockpit*.
- SAP HANA studio
  For more information, see *Monitoring System Replication with the SAP HANA Studio*.
- hdbnsutil
  For more information, see *Monitoring System Replication with hdbnsutil*.
- SQL query
  For more information, see *Monitoring System Replication with SQL query*.

Related Information

Monitoring SAP HANA System Replication in SAP HANA Cockpit [page 182]
Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit [page 184]
Monitoring SAP HANA System Replication with the SAP HANA Studio [page 192]
Monitoring SAP HANA System Replication with hdbnsutil [page 190]
Monitoring SAP HANA System Replication with SQL query [page 193]

8.4.1 Monitoring SAP HANA System Replication in SAP HANA Cockpit

To monitor SAP HANA system replication, you can use the *System Replication* card in the SAP HANA cockpit.

To open the *System Replication Overview* page, with the *Monitoring* or *All* view selected, click the *System Replication* card on the *Database Overview* page in the SAP HANA cockpit.

The *System Replication Overview* displays a graphical representation of the system replication landscape with the following information:

- The name and role of the system, as well as the selected operation mode
  For the operation modes `logreplay` and `logreplay_readaccess`, a retention time estimation is also displayed. The *Estimated log retention time* is an estimation of the time left before the primary system starts to overwrite the *RetainedFree* marked log segments and a full data shipping becomes necessary to get the primary and secondary systems back in sync after a disconnect situation. The *Estimated log full time* is an estimation of the time left before the primary system runs into a log full. The value shown in the header shows the situation into which the system could run first: log retention or log full.
- If the SQL ports of the secondary system are open for read access
- The replication mode used between the systems
- The current average redo log shipping time and the average size of shipped redo log buffers

Furthermore, detailed information on system replication is provided in the following tabs:

> **i Note**
>
> These tabs are displayed only if you configured a system replication before.

Tabs on the System Replication Overview

Tab Name	Description
Graphical Overview	The *Graphical Overview* tab provides information about each tier.
Related Alerts	The *Related Alerts* tab provides a description of any existing alerts, as well as their priority. This tab is only displayed when system replication-related alerts are available.
Replicated Services	The *Replicated Services* tab provides information on the replication status per site and service.
Network	The *Network* tab provides information on the time it took to ship the redo log to the secondary system and to write the redo log to the local log volume on disk. You can select the network connection that you want to analyze (for example, *Network Site 1 to 2* or *Network Site 2 to 3*). The graph displayed compares the local write wait time with the remote write wait time monitored over the last 24 hours.
Log Shipping Backlog	The *Log Shipping Backlog* tab provides a graphical representation on the history of the log shipping backlog.
Log Replay	The *Log Replay* tab provides a graphical representation on the delay of the secondary system. This tab is displayed if the chosen operation mode for the system replication landscape is `logreplay` or `logreplay_readaccess`. When this tab is activated for a secondary system the log replay delay is initially shown for the last 24 hours but a longer time period can be selected from a list. Furthermore, in this tab you can select to visualize the estimated log retention time as well as the estimated log full time for all system replication relevant services.
Network Speed Check	The *Network Speed Check* tab provides a way to measure the network speed of the system replication host-to-host network channel mappings.
Time Travel	The *Time Travel* tab provides an overview of the available snapshots, as well as the possibility to start a secondary time travel. This tab is only displayed on the overview of the secondary system.
Takeover History	The *Takeover History* tab provides an overview of the entire takeover and time travel history available.
Parameter Settings	The *Network Security Settings* option displays the specific network security details configured between the primary and the secondary systems. The *Invisible Takeover* option provides the possibility to turn on the invisible takeover. 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.

8.4.1.1 Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit

Learn how to monitor multiple SAP HANA systems with the SAP HANA cockpit.

In the SAP HANA cockpit, you can monitor system replication using the system replication tile, the system replication overview, and the tabs on the system replication overview.

System Replication Tile

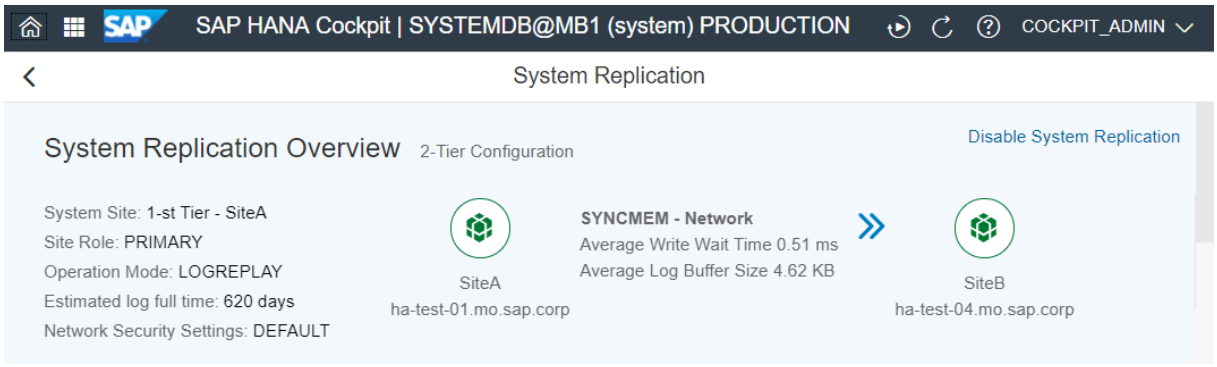
If system replication is configured, the corresponding tile appears on the main screen of the system overview page providing information about the type of landscape (tier 2 or tier 3), the replication modes between the primary and the secondary systems, the operation mode, as well as an overall replication status. For examples, see *System Replication Tile*. If all tiers are shown in green and the system replication tile indicates that all services are active and in sync, your system is doing well. Red tiers would indicate a problem with the replication.

If the tile does not show up, you have to grant the system replication specific role to the corresponding user.

System Replication Overview

To check the status of replication in detail, open the system replication tile. The application provides an overview on the system replication configuration and status. The “chain” of systems with their replication modes is also displayed containing further information about the sites and the network connections between them.

For example, the system replication overview can look like this:



A graphical representation of your system replication landscape is given. It tells you the chosen system names, the replication mode used between the systems, as well as how long it took on average to send redo log buffers to the secondary system based on measurements from the last 24 hours. For a synchronous replication, this is the round trip time for sending the redo log buffer and receiving the acknowledgement; for an asynchronous replication it refers to the time that it takes until the log buffer was sent after its creation.

Related Alerts Tab

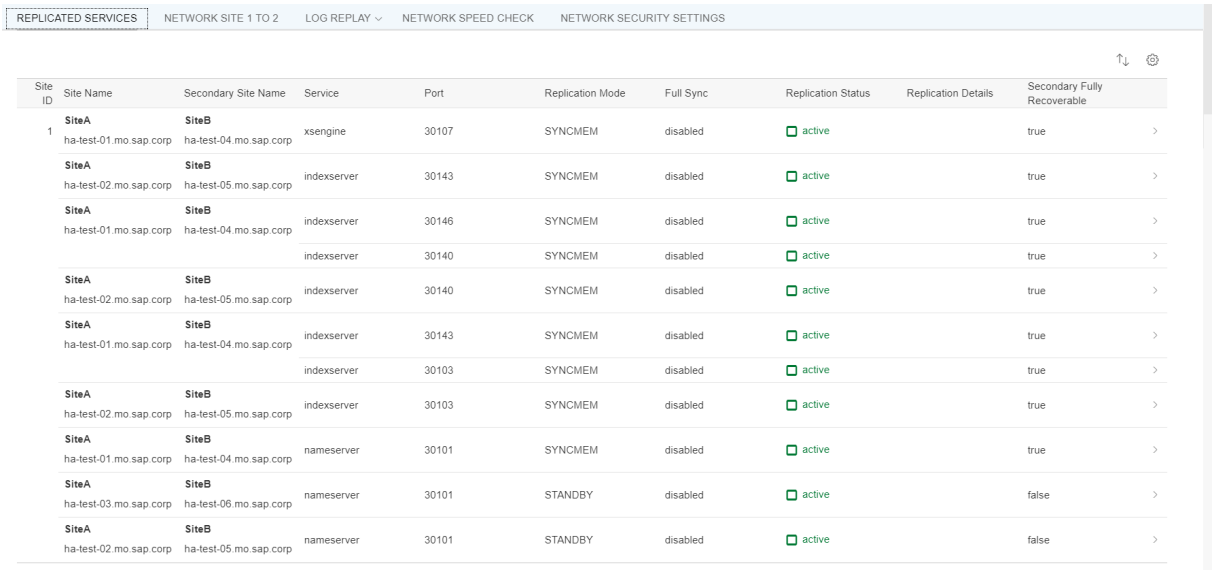
If a system replication relevant alert occurred, the first tab is the *Related Alerts* tab.

If no system replication relevant alert exists, this tab is not shown.

Replicated Services Tab

In the *Replicated Services* tab, an excerpt of the M_SERVICE_REPLICATION monitoring view is shown. The displayed table shows the replication state for each system and service.

For example, the tab can look like this:

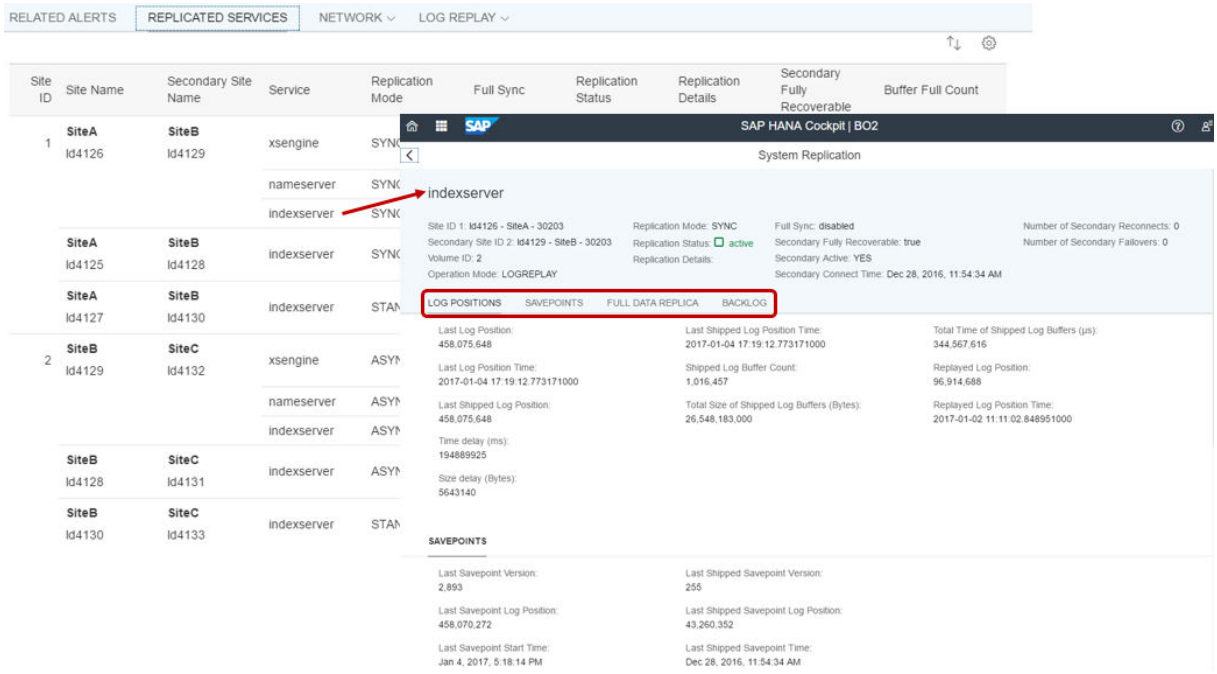


Site ID	Site Name	Secondary Site Name	Service	Port	Replication Mode	Full Sync	Replication Status	Replication Details	Secondary Fully Recoverable
1	SiteA ha-test-01.mo.sap.corp	SiteB ha-test-04.mo.sap.corp	xsengine	30107	SYNCMEM	disabled	active		true
	SiteA ha-test-02.mo.sap.corp	SiteB ha-test-05.mo.sap.corp	indexserver	30143	SYNCMEM	disabled	active		true
	SiteA ha-test-01.mo.sap.corp	SiteB ha-test-04.mo.sap.corp	indexserver	30146	SYNCMEM	disabled	active		true
			indexserver	30140	SYNCMEM	disabled	active		true
	SiteA ha-test-02.mo.sap.corp	SiteB ha-test-05.mo.sap.corp	indexserver	30140	SYNCMEM	disabled	active		true
	SiteA ha-test-01.mo.sap.corp	SiteB ha-test-04.mo.sap.corp	indexserver	30143	SYNCMEM	disabled	active		true
			indexserver	30103	SYNCMEM	disabled	active		true
	SiteA ha-test-02.mo.sap.corp	SiteB ha-test-05.mo.sap.corp	indexserver	30103	SYNCMEM	disabled	active		true
	SiteA ha-test-01.mo.sap.corp	SiteB ha-test-04.mo.sap.corp	nameserver	30101	SYNCMEM	disabled	active		true
	SiteA ha-test-03.mo.sap.corp	SiteB ha-test-06.mo.sap.corp	nameserver	30101	STANDBY	disabled	active		false
	SiteA ha-test-02.mo.sap.corp	SiteB ha-test-05.mo.sap.corp	nameserver	30101	STANDBY	disabled	active		false

> → Recommendation
>
> If you want to look at the trace files of all systems in your system replication landscape, choose the *View trace and diagnostic files* link on the primary system's overview page in the SAP HANA cockpit. This makes all diagnosis files from the trace directories of all systems visible in the browser.

To see the details for the corresponding service grouped thematically, choose one row like in the following example for one indexserver. Since this information is context aware, you only get the information required for this system. For example, because this system is running with the `logreplay` operation mode, no information on `delta data shipping` is shown here. However, the context-sensitive information about the log replay

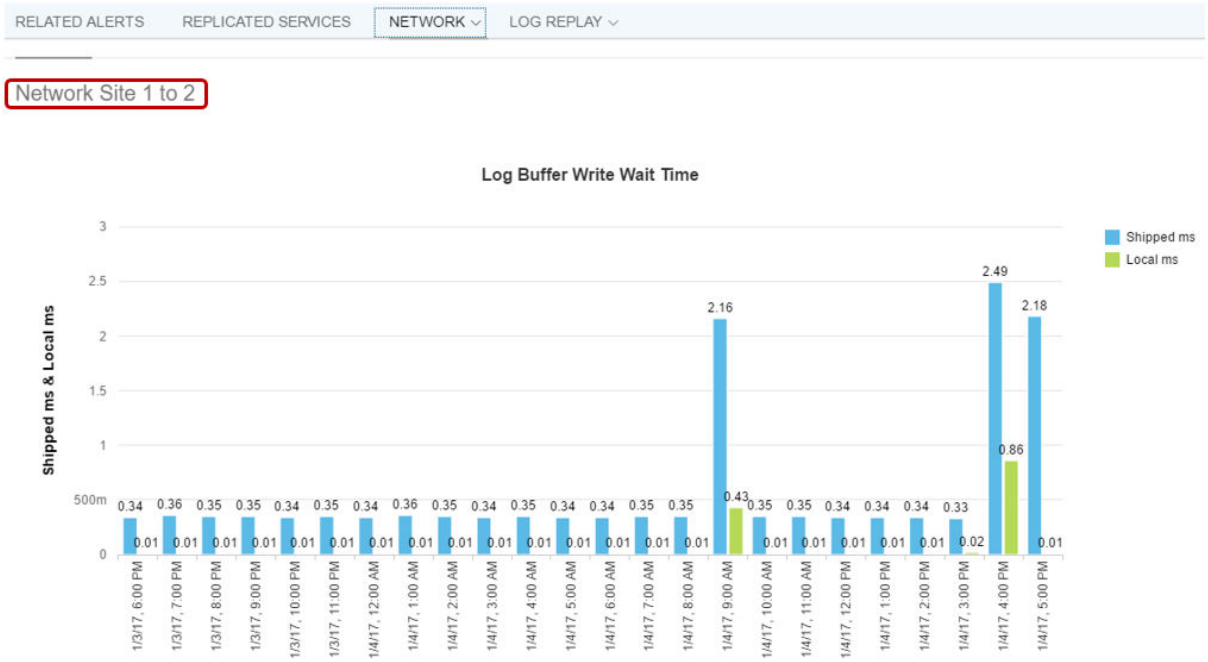
delay is displayed. The delta between *Last Log Position* and *Replayed Log Position* indicates how far the log replay is behind on the secondary system:



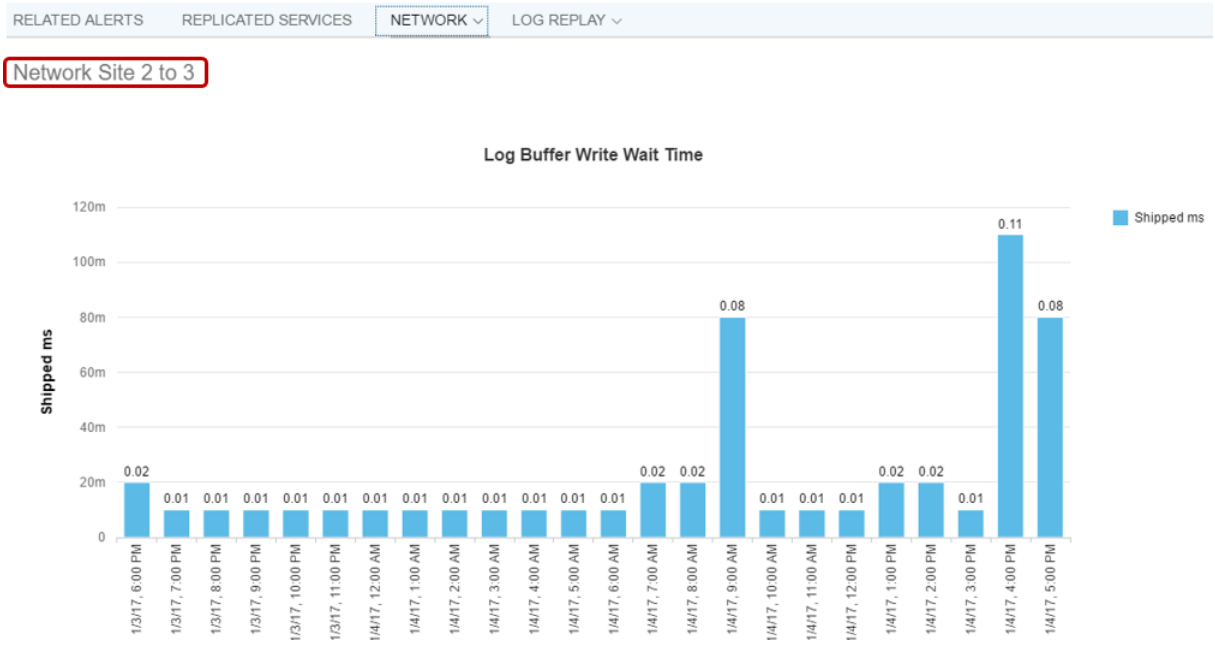
Network Tab

You can select the network connection that you want to analyze from the drop down menu. This can be *Network Site 1 to 2* or *Network Site 2 to 3*. The displayed graphic compares the local write wait time (writing redo log buffer into the local log volume) with the remote write wait time (shipping the redo log and receiving the acknowledgement) monitored over the last 24 hours. You can see if peak times occurred and how the network connection reacted.

For example:



If the ASYNC replication mode is configured between two systems, you also receive information about the network performance by selecting the corresponding connection between tier 2 and tier 3. For example:

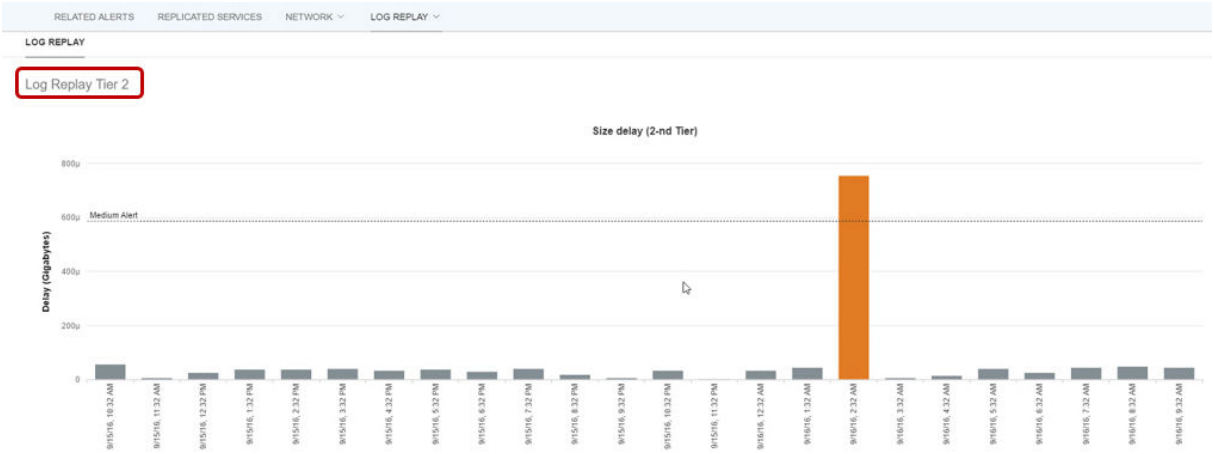


In this example, the average write wait time in ms describes the time it took from the creation of the redo log buffer (committing a write transaction) until the redo log buffer was sent out to the network. This value is an indicator for peak load phases and could point to network or I/O problems on the secondary system, which can influence the primary's performance as well.

For more information, see *Monitoring the Network Latency*.

Log Replay Tab

This tab is only visible, if `logreplay` or `logreplay_readaccess` is configured for the system replication landscape. The log replay delay on the secondary system is displayed as size in GB for the last 24 hours. If at some point in time, the threshold of the corresponding alert (for example, ID 94 System Replication Logreplay Backlog) was exceeded, this is indicated accordingly. For example:



For more information on how to analyze a high replay backlog, see *SAP Note 2409671: High replay backlog on HANA System Replication Secondary Site*.

Additionally, you can select the log retention information for your system. For more information, see *Estimating the Maximum Retention Time*.

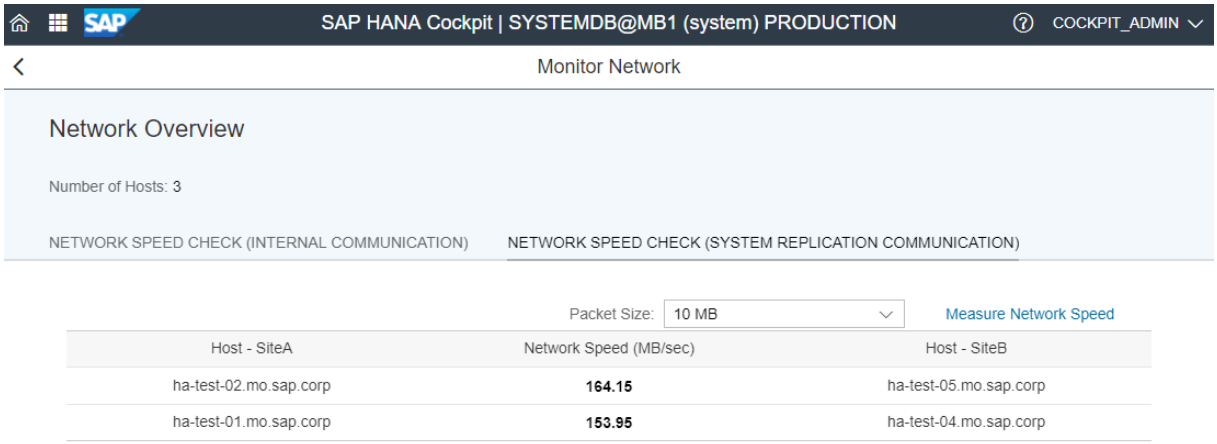
Network Speed Check

To measure the network speed for system replication, host-to-host mappings from the primary system to the secondary system (and even to a third tier) can be done using the *Network Speed Check (System Replication Communication)* tab.

To measure the network speed (MB/s), choose a package size from the drop-down menu and choose *Measure Network Speed*.

In this example, there are standby hosts configured on the primary (ha-test-03) and the secondary (ha-test-06) system. The network speed measurement cannot be done for standby hosts, because they do not

have access to the data and log volumes of the SAP HANA database and thus are not relevant for replication in this state:

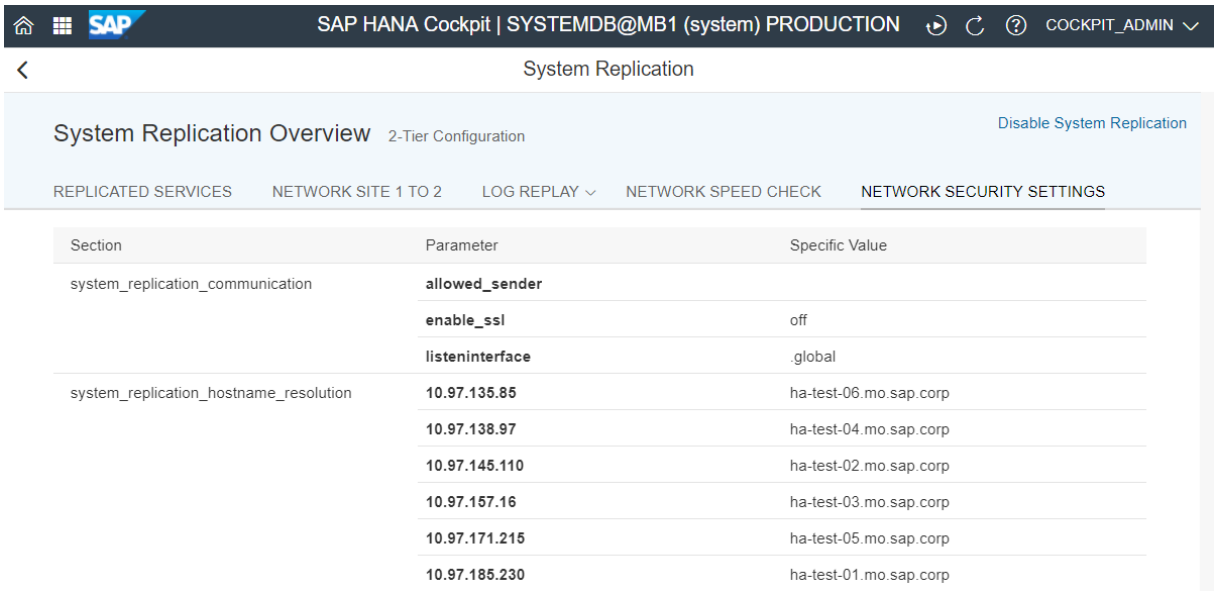


You can also measure the network speed for scale-out SAP HANA databases using the*Network Speed Check (Internal Communication)* tab. It measures the network speed in both directions for each host to each other host of the scale-out system. To do so, choose a package size from the drop-down menu and choose *Measure Network Speed*. The result is a list representing the network speed for every host in the scale-out system to every other host of the system. The measurement is done in both directions. Thus, every host-mapping is displayed twice in this list with switched roles of *Sender* and *Receiver*. The fastest connection is displayed first.

> **i Note**
>
> Measuring the network speed of your SAP HANA system replication communication channels or internode communication channels in your SAP HANA scale-out system will have an impact on your network performance and thus on your running systems for the time of the measurement.

Network Security Settings

This tab shows the security details configured between the primary and the secondary systems:



Related Information

Estimating the Maximum Retention Time [page 23]
System Replication Tile [page 42]
Monitoring the Network Latency [page 200]
SAP Note 2409671

8.4.2 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.

> **i 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

8.4.3 Monitoring SAP HANA System Replication with the SAP HANA Studio

You can monitor SAP HANA system replication using the SAP HANA studio.

You can monitor system replication in the administration editor of the primary system as follows:

- The general status is displayed on the *Overview* tab.

This should be *All services are active and in sync*.

- Detailed information is available on the *Landscape* > *System Replication* tab.
  Here you can see the system replication status. For all services, the REPLICATION_STATUS should be ACTIVE. Detailed information about shipped sizes and shipping times are also available.
  Since the secondary instance does not accept SQL connections while data replication is active, basic information about the secondary system is also shown.
  For more information about the meaning of the individual fields, see the M_SERVICE_REPLICATION system view. For more information on the system replication status, see *Checking the System Replication Status*.

Related Information

Checking the SAP HANA System Replication Status [page 166]
M_SERVICE_REPLICATION System View [page 257]

8.4.4 Monitoring SAP HANA System Replication with SQL query

Use SQL to directly get system replication specific information from the system views.

The M_SYSTEM_REPLICATION view provides general system replication relevant information about the whole system. For example, it gives information on the used replication mode, operation mode, and as which tier a system is configured.

Example

In this example SiteA with SITE_ID 1 is currently configured as TIER 1 as primary.

On the primary execute:

```
select * from "SYS"."M_SYSTEM_REPLICATION";
```

UT1 (SYSTEM) [Production System] Id2131 01

SQL Result

```
select * from "SYS"."M_SYSTEM_REPLICATION"
```

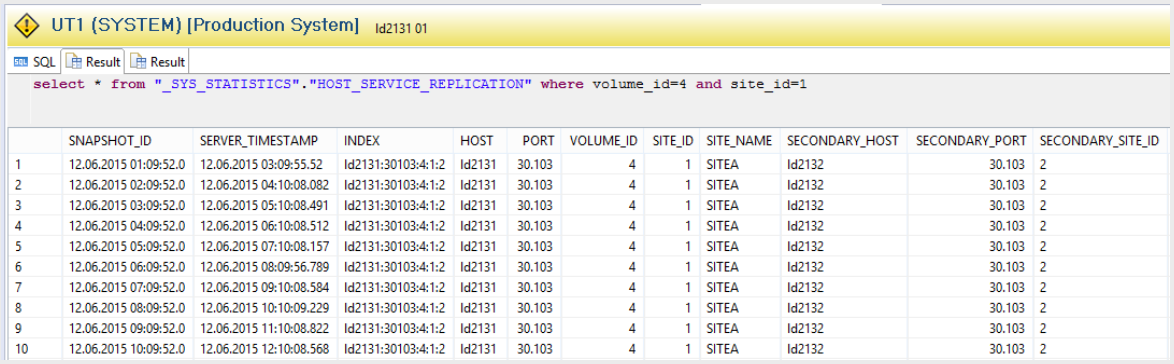
	SITE_ID	SITE_NAME	SECONDARY_SITE_ID	SECONDARY_SITE_NAME	REPLICATION_MODE	REPLICATION_STATUS	OPERATION_MODE	TIER
1	1	SiteA	2	SiteB	SYNC	ACTIVE	logreplay	1
2	2	SiteB	3	SiteC	ASYNC	ACTIVE	logreplay	2

The M_SERVICE_REPLICATION system view is another possibility to get system replication information. The contents of the M_SERVICE_REPLICATION view are collected by the statistics server every hour. Therefore, the history of the data and log replication can be viewed in the table.

Example

On the primary, execute the following command to view the data replicated by the index servers (volume 4 in this example) from the primary to the tier 2 secondary:

```
select * from "_SYS_STATISTICS"."HOST_SERVICE_REPLICATION"
where volume_id=4 and site_id=1;
```


UT1 (SYSTEM) [Production System] ld2131 01

SQL | Result | Result

```
select * from "_SYS_STATISTICS"."HOST_SERVICE_REPLICATION" where volume_id=4 and site_id=1
```

	SNAPSHOT_ID	SERVER_TIMESTAMP	INDEX	HOST	PORT	VOLUME_ID	SITE_ID	SITE_NAME	SECONDARY_HOST	SECONDARY_PORT	SECONDARY_SITE_ID
1	12.06.2015 01:09:52.0	12.06.2015 03:09:55.52	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
2	12.06.2015 02:09:52.0	12.06.2015 04:10:08.082	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
3	12.06.2015 03:09:52.0	12.06.2015 05:10:08.491	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
4	12.06.2015 04:09:52.0	12.06.2015 06:10:08.512	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
5	12.06.2015 05:09:52.0	12.06.2015 07:10:08.157	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
6	12.06.2015 06:09:52.0	12.06.2015 08:09:56.789	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
7	12.06.2015 07:09:52.0	12.06.2015 09:10:08.584	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
8	12.06.2015 08:09:52.0	12.06.2015 10:10:09.229	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
9	12.06.2015 09:09:52.0	12.06.2015 11:10:08.822	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2
10	12.06.2015 10:09:52.0	12.06.2015 12:10:08.568	ld2131:30103:4:1:2	ld2131	30.103	4	1	SITEA	ld2132	30.103	2

Related Information

M_SERVICE_REPLICATION System View [page 257]
M_SYSTEM_REPLICATION System View [page 261]
SAP HANA SQL and System Views Reference

8.5 System Replication Network Connection

The replication in a configured system replication uses either a public or a separate network channel between the involved data centers.

Learn in this section how to configure and monitor the system replication network connection. For detailed information about network speed chekcs with SAP HANA cockpit, see *Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit*.

For more information about security aspects, see also *Security Aspects for SAP HANA System Replication*.

For information about the distance between the data centers, network throughput, as well as data and log compression, see *Network Recommendations*.

Related Information

Secure Configuration of the Network Connection [page 195]

Encryption of the Connection [page 198]
Monitoring the Network Connection [page 199]
Monitoring the Network Latency [page 200]
Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit [page 184]
Security Aspects for SAP HANA System Replication [page 239]
Network Recommendations [page 27]

8.5.1 Secure Configuration of the Network Connection

By default, the primary and secondary systems establish communication using the internal host names.

With an `IPaddress~virtualHostname` mapping on the involved systems, the system replication host name resolution can be set configuring a separate network for system replication data traffic between the primary and the secondary system.

This is done in the `[system_replication_hostname_resolution]` section in global.ini, where all hosts of the primary and the secondary systems must be defined on each site:

```
global.ini/[system_replication_hostname_resolution]
<ip-address_same_site>=<internal_host_same_site>
<ip-address_other_site>=<internal_host_other_site>
```

This is also valid for a multitier system replication consisting of three sites (primary, tier 2 secondary, tier 3 secondary or more) because the roles can switch after takeovers and failbacks.

> **i Note**
>
> The parameters in the global.ini file must be set prior to registering the secondary system, because the `hdbnsutil -sr_register` command uses this mapping. Registration is one step in the process of configuring the secondary system.

The entries in the `[system_replication_hostname_resolution]` section are used in combination with the `listeninterface` parameter in the `[system_replication_communication]` section. The following combinations are possible:

[system_replication_communication] listeninterface	[system_replication_hostname_resolution]	Additional Information
.global	No mappings specified	Default, if nothing is specified. The default network route is used for system replication communication. **i Note** 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 SSL.

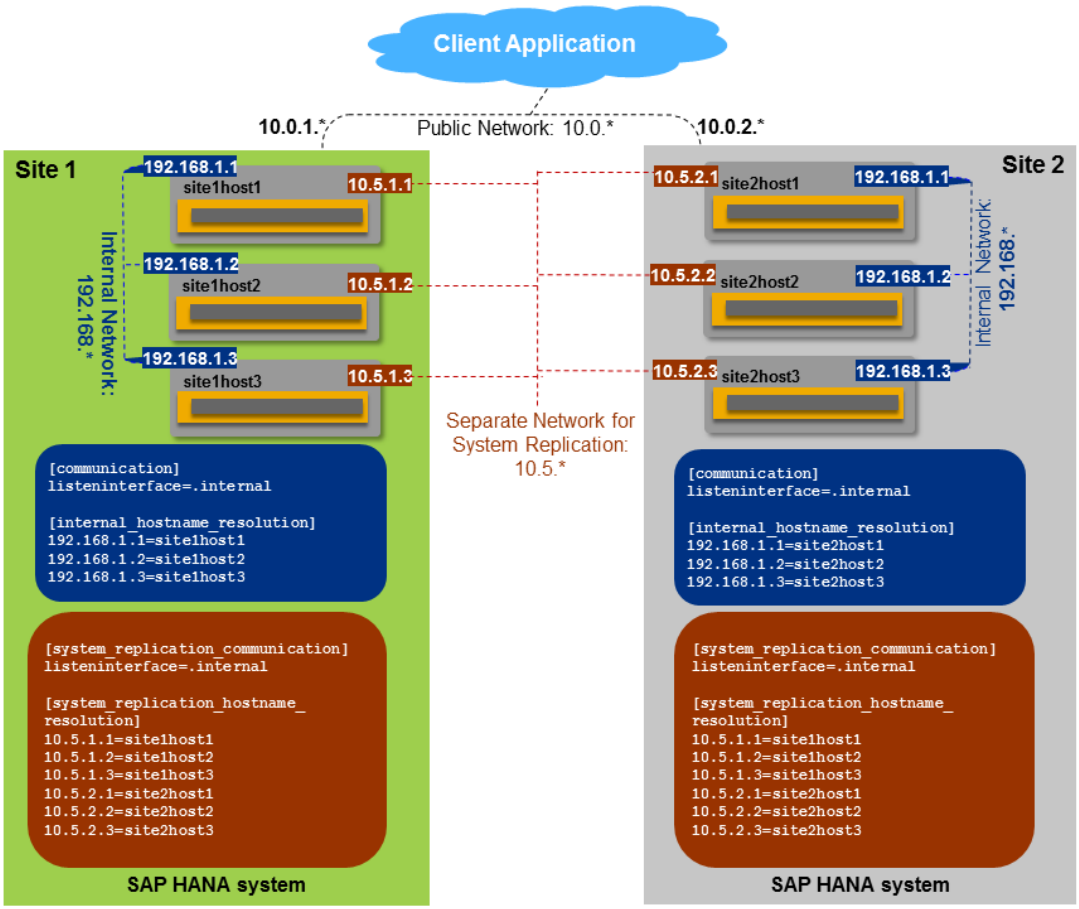
[system_replication_communication] listeninterface	[system_replication_hostname_resolution]	Additional Information
.global	Entries for the primary and secondary hosts (for all hosts in multitier setups)	A separate network is used for system replication communication. **→ Recommendation** In this way you can use a separate network for multitier system replication.
.internal	Entries for the primary and secondary hosts	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. **i Note** In SAP HANA 1.0 SPS 11, network communication for system replication with `listeninterface=.internal` is supported for tier 2 replication, but not for tier 3 setups.

There are two ways to activate `[system_replication_hostname_resolution]` in your system:

- Restart all sites after setting the parameter
- Temporarily resolve the system replication configuration (no restart of the primary is necessary):
  1. Stop secondary.
  2. Unregister secondary.
  3. Disable primary.
  4. Enable primary.
  5. Register secondary.
  6. Start secondary.

Example

Here is an example of the settings for a tier 2 system replication (3 node system) using a separate internal network per site and a separate connection for the system replication:



If for some reason, no separate network channel was configured for the SAP HANA system replication communication between the involved systems, the `allowed_sender` parameter could be used to restrict communication between primary and secondary to certain hosts. For this, the following settings can be configured in the global.ini file on the primary system:

```
global.ini/[system_replication_communication]
Parameter: allowed_sender
Value: <list of IP-addresses of secondary or CIDR-netmasks>
Example: 10.0.1.0/30
```

The default is "no restriction".

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

Host Name Resolution for System Replication
Security Aspects for SAP HANA System Replication [page 239]
SAP Community Blog: How to configure HANA network communication channels – Part 2: Internal network

8.5.2 Encryption of the Connection

The system replication connections for data, redo log, and metadata of the nameserver are secured with SSL by systemPKI without the need to switch on SSL for the internal communication.

All system replication connections are specified only by the setting `[system_replication_communication]/enable_ssl`. This simplifies security settings for system replication.

> **i** Note
>
> Before upgrading SAP HANA from a previous version to HANA 2.0 SPS02 or higher using near zero downtime upgrade (NZDU), see *SAP Note 2494079: Near-Zero-Downtime-Upgrade to HANA 2 SPS02 or above when internal communication SSL is used*.

SAP HANA System Replication supports secure network communication (SSL) for data and log shipping to the secondary system. The following settings can be configured in the global.ini file:

```
global.ini/[system_replication_communication]
Parameter: enable_ssl
Values:
off:     ssl is disabled for source and target replication channels (default)
on:      ssl is enabled for source and target replication channels
source:  ssl is enabled as source replication channel only
target:  ssl is enabled as target replication channel only
```

In this context, the 4xx06 port was added in system replication communication covering encrypted metadata communication of the nameserver. The following ports are available:

TCP Port	Service	Used For
4xx01	nameserver	Log and data shipping (System DB)
4xx02	nameserver	Metadata communication (System DB) – **unencrypted**
4xx06	nameserver	Metadata communication (System DB) – **encrypted via SSL**
4xx40 - 4xx97	indexserver	Log and data shipping (tenant DBs)
4xx40 - 4xx97	scriptserver	Log and data shipping (optional, tenant DBs)
4xx40 - 4xx97	docstore	Log and data shipping (optional, tenant DBs)

Related Information

Security Aspects for SAP HANA System Replication [page 239]
SAP Note 2494079

8.5.3 Monitoring the Network Connection

You can monitor the network connection using alerts or the HDB console.

The connection between the primary and the secondary system must be available for replication. If this is not the case for a certain time, the redo log cannot be shipped to the secondary system, the log segments start piling up on the primary, and the secondary system is not ready for takeover. For more information, see *Log Retention*.

The alert 78 (System Replication Connection Closed) is visible in the SAP HANA cockpit, the SAP HANA studio, and in the system view M_EVENTS to ensure that the primary system stays operational at all times, even if the connection is occasionally lost.

Additionally, the replication connection can be checked using the HDB console: `hdbcons -p <PID of replicating service> "replication info"` . The PID (process ID) of the replicating service can be obtained by using `HDB proc` as `<sid>`adm. For example: `hdbcons -p 12345 "replication info"`

The output delivers Log Connection information for the connection used by the provided service. It also shows errors, if the connection cannot be resolved properly:

```
…
Log Connection
- ptr : 0x00007fdb6e8e3410
- channel : NetworkChannel FD 158 [0x00007fdb6f1bbc90] {refCnt=3, idx=1}
10.68.91.226/3 0103_tcp->10.68.92.13/49537_tcp Connected,[r---]
...
```

To check, if the configured connection is actually used, use the OS command: `lsof -n -p <indexserver-pid>`.

For a detailed analysis of the network connection used for system replication, see *Troubleshoot System Replication* and *SAP Note 2081065: Troubleshooting SAP HANA Network*.

Related Information

Log Retention [page 21]
SAP HANA System Replication Alerts [page 161]
Troubleshoot System Replication [page 216]
SAP Note 2081065

8.5.4 Monitoring the Network Latency

You can monitor the network latency with the SAP HANA cockpit or the SAP HANA studio.

The latency for the redo log shipping is of interest in a running system replication for the synchronous replication modes SYNC or SYNCMEM regardless of the used operation mode.

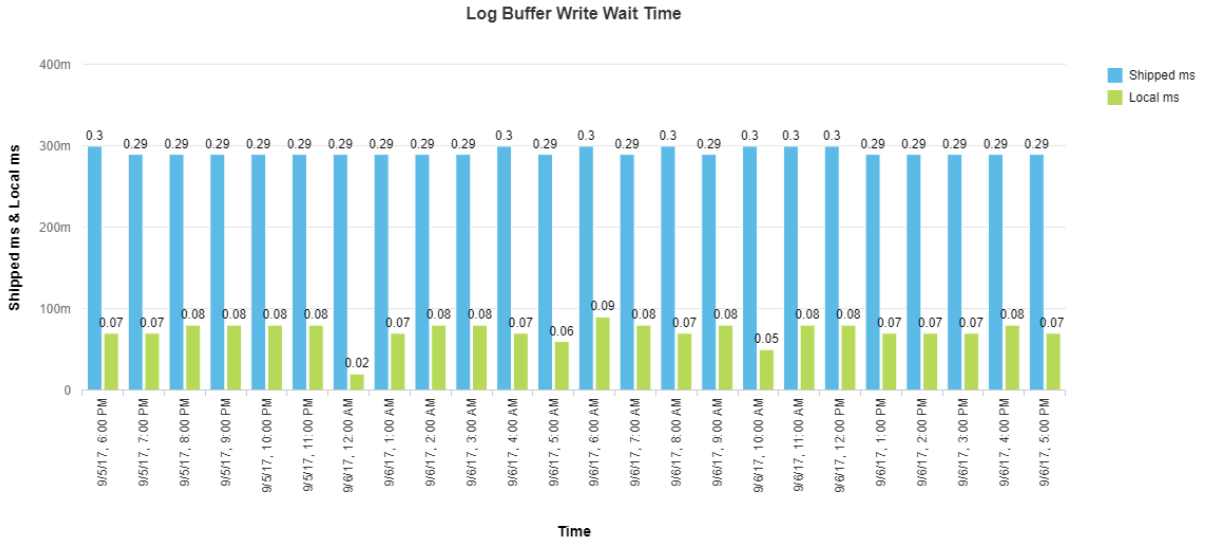
> **i Note**
>
> The redo log shipping wait time for 4 KB log buffers must be less than a millisecond or in a low single-digit millisecond range – depending on the application requirements (relevant for synchronous replication modes only).

All changes to data are captured in the redo log. The SAP HANA database asynchronously persists the redo log with I/O orders of 4 KB to 1 MB size into log segment files in the log volume of the primary system. A transaction writing a commit into the redo log waits until the buffer containing the commit has been written to the log volume. This wait time for 4 KB log buffers should be less than a millisecond or in a low single-digit millisecond range. Once you configured system replication, you can check the local and the remote log write wait times in the SAP HANA cockpit or collect them in the SAP HANA studio with the SQL statement *HANA_Replication_Overview* attached to the *SAP Note 1969700*.

Network Latency in the SAP HANA Cockpit

In the SAP HANA cockpit you receive an overview of your running system replication displaying the average log buffer size between the primary and the secondary, as well as the average redo log shipping wait time. Opening the *Network* tab, you can also have a look at the 24 hour history of these local log write wait times and the remote log shipping wait times. For more information, see *Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit*.

For example:



Network Latency in the SAP HANA Studio

You can gather these values also with the SAP HANA studio after importing the SQL statement collection (attached to *SAP Note 1969700*) to the SAP HANA studio: *System Information* > *Import SQL Statements*.

Right-click on the statements under *Replication* > *Overview* and choose *Execute* from the context menu. You will receive information about the system replication landscape and the replication state for each service.

The log write wait times on the "local" system (that is, on the persistence of the primary system) is returned by the mentioned SQL statement as *Average local log buffer write time (ms)* per service, where the index server is the one of interest.

Additionally, the mentioned SQL statement returns the redo log write latency for the shipping to the secondary, which can be slightly higher than the locally measured log write wait time. The returned value *Average local log buffer write time (ms)* represents the time between enqueueing and finishing a request.

Of interest is also the *Local log buffer write throughput (MB/s)* compared to the *Log buffer shipping throughput (MB/s)* in synchronous replication. If these two values differ too much, this could be an indication for network problems in a synchronous replication or a problem with the I/O on the secondary system (for SYNC). For information about ASYNC replication, see the *Check Network Configuration (Long Distance)* section in *Replication Performance Problems*.

Related Information

Example: Monitoring SAP HANA System Replication with the SAP HANA Cockpit [page 184]
Replication Performance Problems [page 223]
SAP Note 1969700

8.6 Copy or Move Tenants Within System Replication

You can copy or move a tenant in a SAP HANA tenant database system into or out of a primary system running in a system replication configuration.

There are three tenant copy or move scenarios that are applicable in a primary system:

Copy or move a tenant into a primary system

A tenant database is copied or moved into the primary system of a system replication configuration.

The copied or moved tenant starts replicating after the FINALIZE command was executed.

Copy or move a tenant from the primary system to a tenant database system (≠ secondary)

A tenant database is copied or moved out of the primary system into a target tenant database different from the secondary.

The copied or moved tenant arrives in the target tenant database system and is runnable after the FINALIZE command was executed.

Copy or move a tenant within the primary system

A tenant database is copied or moved within the primary system of a system replication.

The “cloned” tenant is created in the primary system and the replication starts after the FINALIZE command was executed.

Related Information

Copying and Moving Tenant Databases
SAP HANA System Replication with Tenant Databases [page 87]

8.7 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.

8.7.1 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.

8.8 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*.

> **i** 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 19]
Configure a User Under the SRTAKEOVER Key [page 209]
Update SAP HANA Systems Running in a System Replication Setup [page 204]
Use SAP HANA System Replication for Near Zero Downtime Upgrades [page 206]
Use Multitarget System Replication for Near Zero Downtime Upgrades [page 210]
SAP Note 2369981
SAP Note 1984882

8.8.1 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.

> i 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 206]
Prepare an Update for Flexible System Downtime

SAP Note 2407186 - How-To Guides & Whitepapers For SAP HANA High Availability

SAP Note 2599514 - Upgrade of Hana 1.0 to 2.0 in System Replication scenario without NZDU

8.8.2 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.

> **i** 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
```

> **i 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 209]
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 210]
Deploy a Multi-Target Application with Zero-Downtime Maintenance
Takeover with Handshake [page 104]
SAP Note 2369981 - Required configuration steps for authentication with HANA System Replication
SAP Note 1984882 - Using HANA System Replication for Hardware Exchange with minimum to zero Downtime
SAP Note 2386973 - Near Zero Downtime Upgrades for HANA database 3-tier System Replication
SAP Note 2494079 - Near-Zero-Downtime-Upgrade to HANA 2 SPS02 or above when internal communication SSL is used
SAP Note 2407186 - How-To Guides & Whitepapers For SAP HANA High Availability
SAP Note 2300936 - Host Auto-Failover & System Replication Setup with SAP HANA extended application services, advanced model

8.8.2.1 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>
   ```

   > **i 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
   ```

   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.

   For more information, see the *Secure User Store (hdbuserstore)* section in the *SAP HANA Security Guide*.

   > **i 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 206]
Secure User Store (hdbuserstore)

8.8.2.2 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.

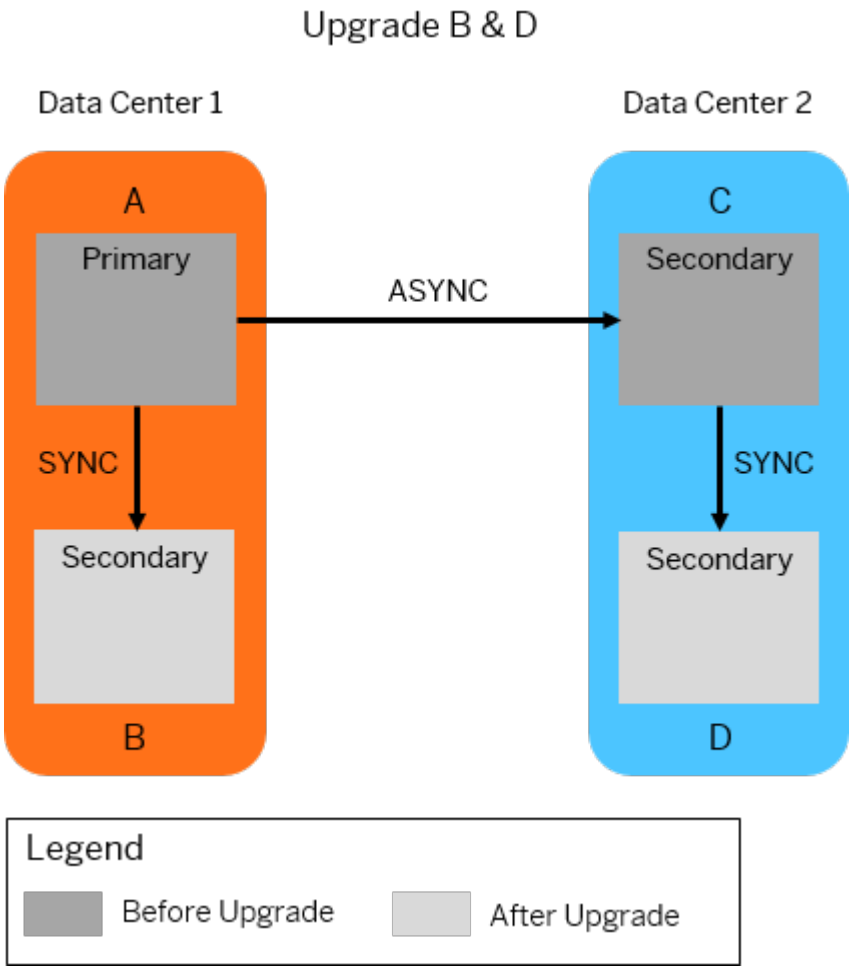
Context

We are using the following setup to exemplify the procedure:

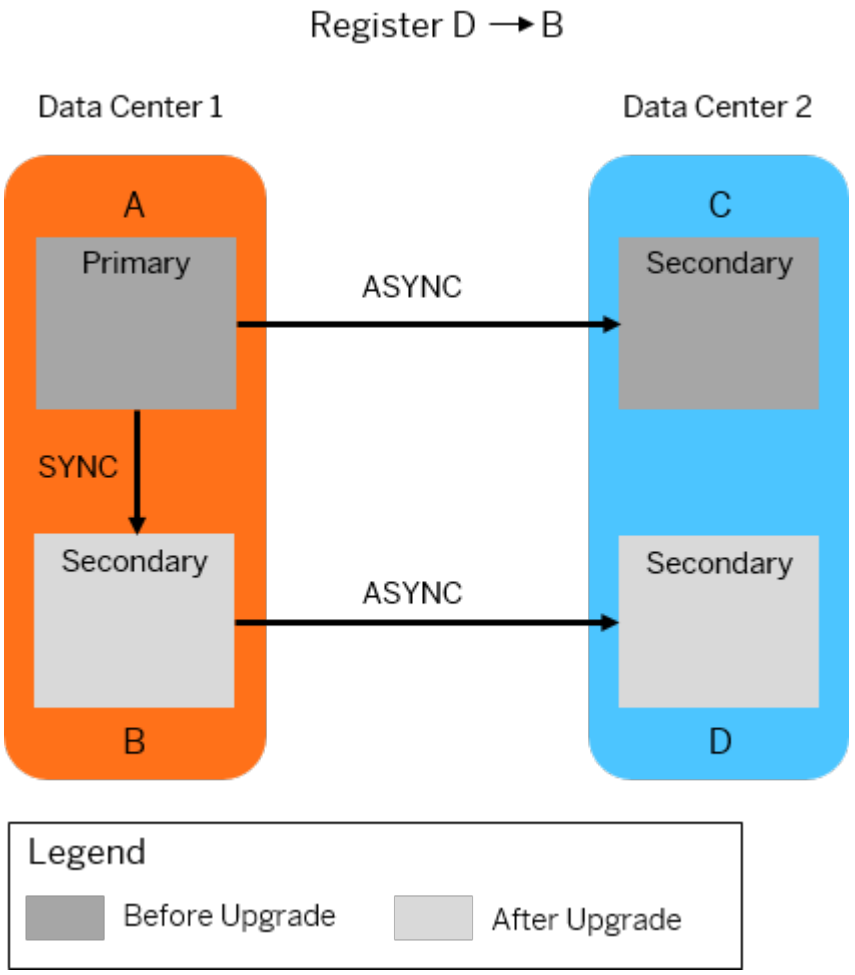
> ✤ Example
>
> Primary system A replicates data changes to secondary system B located in the same data center. Primary system A also replicates data changes to the secondary system C located in data center 2. Secondary system C is a source system for a further secondary system D located in the same 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

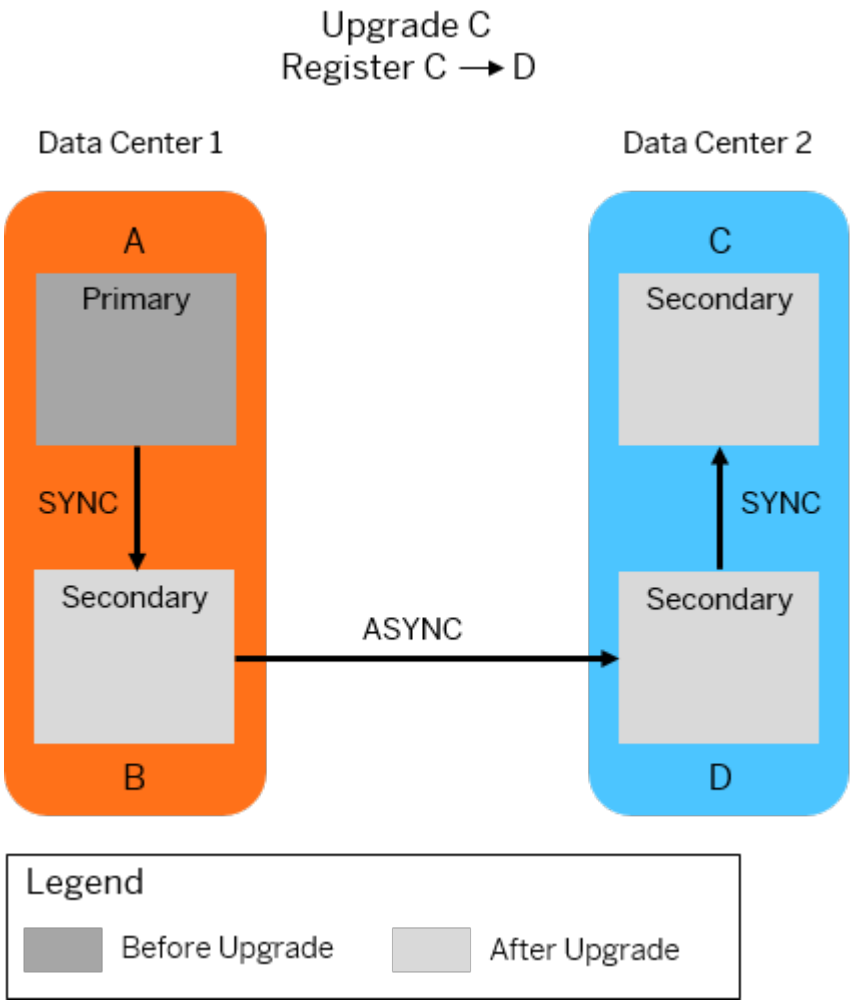
1. Upgrade secondary system B in data center 1 and secondary system D in data center 2.



2. Register secondary system D in data center 2 to secondary system B in data center 1.

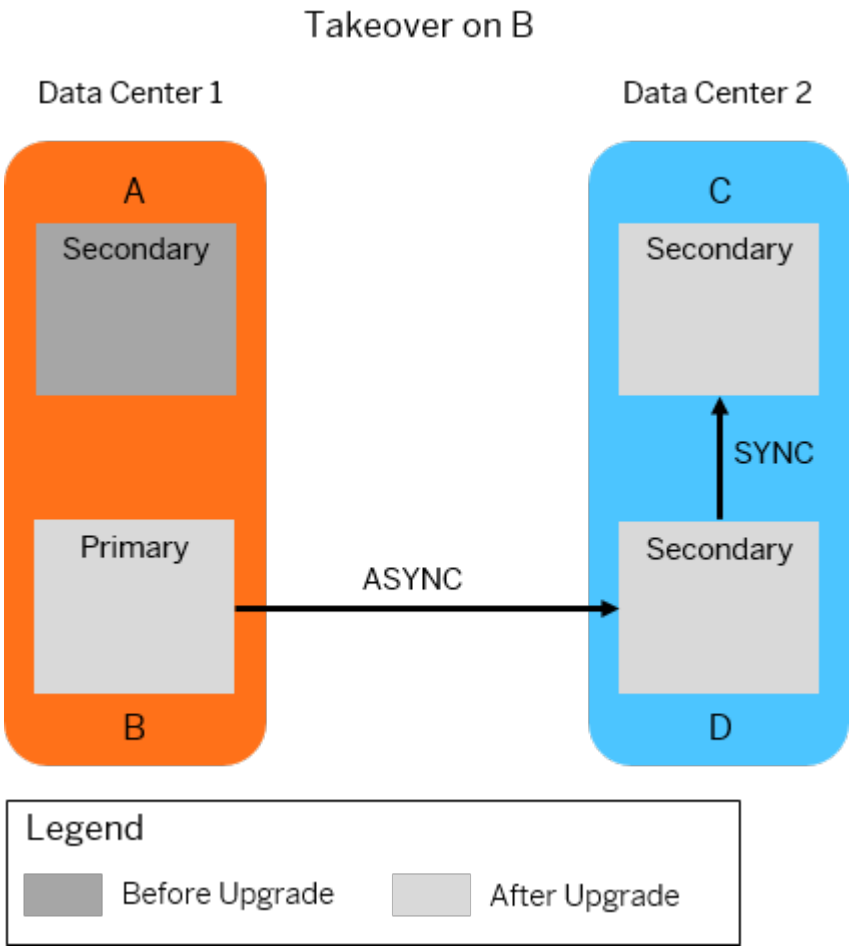


3. Upgrade secondary system C in data center 2. Then, register secondary system C to secondary system D in data center 2.

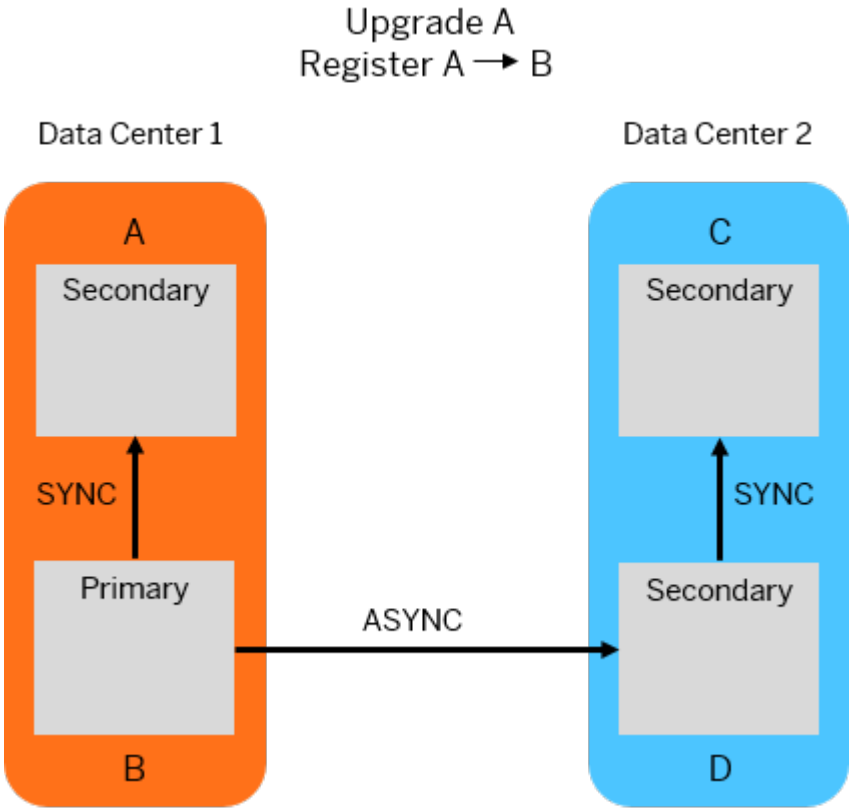


4. Take over on secondary system B in data center 1.

   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 148]

9 Troubleshoot System Replication

Learn how to analyze, avoid, and solve problems related to system replication.

Which problems can I avoid or solve related to system replication?

This chapter provides information about:

- Solving I/O performance problems related to system replication
- Determining the underlying cause for replication performance problems and solving them
- Solving configuration problems during the initial SAP HANA system replication setup
- Solving sporadic network interruptions causing problems in the SAP HANA system replication mechanism
- Avoiding log full situations and recovering from them
- Solving communication problems when they occur between the replication sites

Furthermore, the chapter also provides information on how to conduct a network stress test with NIPING and gives you an overview of all SAP HANA alerts.

Where can I find more information?

The following SAP Notes are relevant for troubleshooting system replication:

SAP Notes

SAP Note	Title
1930979	Alert: Sync/Async Read Ratio
1969700	Collection of SQL Statements for SAP HANA
1999880	FAQ: SAP HANA System Replication
2166157	Error: 'failed to open channel ! reason: connection refused' when setting up Disaster Recovery
2083715	Analyzing log volume full situations
1227116	Creating network traces
1679938	Log Volume is full
500235	Network Diagnosis with NIPING

Related Information

I/O Related Root Causes and Solutions [page 217]
Analyzing I/O Throughput and Latency [page 219]
Savepoint Performance [page 221]
Replication Performance Problems [page 223]
Setup and Initial Configuration Problems [page 228]
Intermittent Connectivity Problems [page 232]
LogReplay: Managing the Size of the Log File [page 234]
SAP HANA System Replication Communication Problems [page 237]
Stress Test with NIPING [page 238]
SAP HANA System Replication Alerts [page 161]
SAP HANA Troubleshooting Guide: Alerts Reference

9.1 I/O Related Root Causes and Solutions

This section covers troubleshooting of I/O performance problems. Although SAP HANA is an in-memory database, I/O still plays a critical role for the performance of the system.

From an end user perspective, an application or the system as a whole runs slowly, is unresponsive or can even seem to hang if there are issues with I/O performance. In the *Disk Volume Monitor* available in the *Disk Usage* tile in SAP HANA cockpit you can see the attached volumes and which services use which volumes. For details of the attached volumes, such as files and I/O statistics, select a row.

In certain scenarios data is read from or written to disk, for example during the transaction commit. Most of the time this is done asynchronously but at certain points in time synchronous I/O is done. Even during asynchronous I/O it may be that important data structures are locked.

Examples are included in the following table.

Scenario	Description
Savepoint	A savepoint ensures that all changed persistent data since the last savepoint gets written to disk. The SAP HANA database triggers savepoints in 5 minutes intervals by default. Data is automatically saved from memory to the data volume located on disk. Depending on the type of data the block sizes vary between 4 KB and 16 MB. Savepoints run asynchronously to SAP HANA update operations. Database update transactions only wait at the critical phase of the savepoint, which is usually taking some microseconds.
Snapshot	The SAP HANA database snapshots are used by certain operations like backup and system copy. They are created by triggering a system wide consistent savepoint. The system keeps the blocks belonging to the snapshot at least until the drop of the snapshot. Detailed information about snapshots can be found in the *SAP HANA Administration Guide*.

Scenario	Description
Delta Merge	The delta merge itself takes place in memory. Updates on column store tables are stored in the delta storage. During the delta merge these changes are applied to the main storage, where they are stored read optimized and compressed. Right after the delta merge, the new main storage is persisted in the data volume, that is, written to disk. The delta merge does not block parallel read and update transactions.
Write Transactions	All changes to persistent data are captured in the redo log. SAP HANA asynchronously writes the redo log with I/O orders of 4 KB to 1 MB size into log segments. Transactions writing a commit into the redo log wait until the buffer containing the commit has been written to the log volume.
Database restart	At database startup the services load their persistence including catalog and row store tables into memory, that is, the persistence is read from the storage. Additionally the redo log entries written after the last savepoint have to be read from the log volume and replayed in the data area in memory. When this is finished the database is accessible. The bigger the row store is, the longer it takes until the system is available for operations again.
Failover (Host Auto-Failover)	On the standby host the services are running in idle mode. Upon failover, the data and log volumes of the failed host are automatically assigned to the standby host, which then has read and write access to the files of the failed active host. Row as well as column store tables (the latter on demand) must be loaded into memory. The log entries have to be replayed.
Takeover (System Replication)	The secondary system is already running, that is the services are active but cannot accept SQL and thus are not usable by the application. Just like in the database restart (see above) the row store tables need to be loaded into memory from persistent storage. If table preload is used, then most of the column store tables are already in memory. During takeover the replicated redo logs that were shipped since the last data transport from primary to secondary have to be replayed.
Data Backup	For a data backup the current payload of the data volumes is read and copied to the backup storage. For writing a data backup it is essential that on the I/O connection there are no collisions with other transactional operations running against the database.
Log Backup	Log backups store the content of a closed log segment. They are automatically and asynchronously created by reading the payload from the log segments and writing them to the backup area.
Database Recovery	The restore of a data backup reads the backup content from the backup device and writes it to the SAP HANA data volumes. The I/O write orders of the data recovery have a size of 64 MB. Also the redo log can be replayed during a database recovery, that is the log backups are read from the backup device and the log entries get replayed.

In the following table the I/O operations are listed which are executed by the above-mentioned scenarios, including the block sizes that are read or written:

I/O pattern	Data Volume	Log Volume (redo log)	Backup Medium
Savepoint, **Snapshot,** **Delta merge**	**WRITE** 4 KB – 16 MB asynchronous bulk writes, up to 64 MB (clustered Row Store super blocks)		

I/O pattern	Data Volume	Log Volume (redo log)	Backup Medium
Write transactions		**WRITE** OLTP – mostly 4 KB log write I/O performance is relevant OLAP – writes with larger I/O order sizes	
Table load: **DB Restart,** **Failover,** **Takeover**	**READ** 4 KB – 16 MB blocks, up to 64 MB (clustered Row Store super blocks)	**READ**	
Data Backup	**READ** 4 KB – 16 MB blocks, up to 64 MB (clustered Row Store super blocks) are asynchronously copied to "[data] backup buffer" of 512 MB		**WRITE** in up to 64 MB blocks from "[data] backup buffer"
Log Backup		**READ** asynchronously copied to "[data] backup buffer" of 128 MB	**WRITE** in up to 64 MB blocks from "[data] backup buffer"
Database Recovery	**WRITE** 4 KB – 16 MB blocks, up to 64 MB (clustered Row Store super blocks)	**READ** Read block sizes from backup file headers and copy blocks into "[data] backup buffer" of size 512 MB	**READ** Read block sizes from backup file headers and copy blocks into "[data] backup buffer" of size 128 MB

9.1.1 Analyzing I/O Throughput and Latency

When analyzing I/O, the focus is on throughput and latency (time taken). A set of system views (with names beginning M_VOLUME_IO_*) is available to help you analyze throughput and examples are given here to illustrate how they can be used.

You can use the following example query to read I/O statistics data which will help you to analyze the throughput of the system (in this example the index server). The result of this query presents a set of columns including throughput in MB and trigger ratios (the relationship between trigger time and I/O time) for both read and write operations:

```
select v.host, v.port, v.service_name, s.type,
       round(s.total_read_size / 1024 / 1024, 3) as "Reads in MB",
       round(s.total_read_size / case s.total_read_time when 0 then -1 else
s.total_read_time end, 3) as "Read Throughput in MB",
```

```
       round(s.total_read_time / 1000 / 1000, 3) as "Read Time in Sec",
       trigger_read_ratio as "Read Ratio",
       round(s.total_write_size / 1024 / 1024, 3) as "Writes in MB",
       round(s.total_write_size / case s.total_write_time when 0 then -1 else
s.total_write_time end, 3) as "Write Throughput in MB",
       round(s.total_write_time / 1000 / 1000, 3) as "Write Time in Sec"   ,
       trigger_write_ratio as "Write Ratio"
from "PUBLIC"."M_VOLUME_IO_TOTAL_STATISTICS_RESET" s, PUBLIC.M_VOLUMES v
where s.volume_id = v.volume_id
and type not in ( 'TRACE' )
and v.volume_id in (select volume_id from m_volumes where service_name =
'indexserver')
order by type, service_name, s.volume_id;
```

Note that some of the system views for I/O can be used with a resettable counter so that you can gather data for just the most recent period since the counter was set. This example is based on the M_VOLUME_IO_TOTAL_STATISTICS system view but uses the 'reset' version of the view.

You can reset the statistics counter to analyze the I/O throughput for a certain time frame by running the following reset command:

```
alter system reset monitoring view M_VOLUME_IO_TOTAL_STATISTICS_RESET;
```

Multitier and Replication Scenarios

In a system using replication between primary and secondary sites it is possible to analyze throughput of the secondary remotely by running these queries on the primary site. This method uses the proxy schema of the secondary system on the primary and can be used in a 2-tier system replication setup as well as for multitier landscapes.

The proxy schema follows the naming convention _SYS_SR_SITE_<siteName>, where <siteName> is the name of the secondary site (case-sensitive). In the FROM clause of the example query given above the schema PUBLIC is used. In a system replication landscape replace this with the proxy schema as shown in the following example for a secondary with site name 'SiteB':

```
from "_SYS_SR_SITE_SiteB"."M_VOLUME_IO_TOTAL_STATISTICS_RESET" s,
"_SYS_SR_SITE_SiteB"."M_VOLUMES" v
```

Trigger Ratios

I/O calls are executed asynchronously, that is, the thread does not wait for the order to return. The trigger-ratio of asynchronous reads and writes measures the trigger time divided by the I/O time. A ratio close to 0 shows good performance; it indicates that the thread does not wait at all. A ratio close to 1 means that the thread waits until the I/O request is completed.

Refer to SAP Note 1930979 and SAP Notes for Alerts 60 and 61 for more information about the significance of the trigger ratio values.

Latency

The latency values are important for LOG devices. To analyze the latency, use the following example query which returns the log write wait time (for data of type LOG) with various buffer sizes written by the index server. The time values returned are the number of microseconds between enqueueing and finishing a request.

```
select host, port type,
       round(max_io_buffer_size / 1024, 3) "Maximum buffer size in KB",
       trigger_async_write_count,
       avg_trigger_async_write_time as "Avg Trigger Async Write Time in
Microsecond",
```

```
      max_trigger_async_write_time as "Max Trigger Async Write Time in
Microsecond",
       write_count, avg_write_time as "Avg Write Time in Microsecond",
        max_write_time as "Max Write Time in Microsecond"
from "PUBLIC"."M_VOLUME_IO_DETAILED_STATISTICS_RESET"
where type = 'LOG'
and volume_id in (select volume_id from m_volumes where service_name =
'indexserver')
and (write_count <> 0 or avg_trigger_async_write_time <> 0);
```

Related Information

SAP Note 1930979
M_VOLUME_IO_TOTAL_STATISTICS_RESET System View
Alerts Reference

9.1.2 Savepoint Performance

To perform a savepoint write operation, SAP HANA needs to take a global database lock. This period is called the “critical phase” of a savepoint. While SAP HANA was designed to keep this time period as short as possible, poor I/O performance can extend it to a length that causes a considerable performance impact.

Savepoints are used to implement backup and disaster recovery in SAP HANA. If the state of SAP HANA has to be recovered, the database log from the last savepoint will be replayed.

You can analyze the savepoint performance with this SQL statement:

```
select start_time, volume_id,
       round(duration / 1000000) as "Duration in Seconds",
       round(critical_phase_duration / 1000000) as "Critical Phase Duration in
Seconds",
       round(total_size / 1024 / 1024) as "Size in MB",
       round(total_size / duration) as "Appro. MB/sec",
       round (flushed_rowstore_size / 1024 / 1024) as "Row Store Part MB"
from m_savepoints
where volume_id in ( select volume_id from m_volumes where service_name =
'indexserver') ;
```

This statement shows how long the last and the current savepoint writes took/are taking. Especially the critical phase duration, in which savepoints need to take a global database lock, must be observed carefully.

The critical phase duration should not be longer than a second. In the example below the times are significantly higher due to I/O problems.

	START_TIME	VOLUME_ID	Duration in Seconds	Critical Phase Duration in Seconds	Size in MB	Appro. MB/sec	Row Store Part MB
1	Dec 16, 2013 9:28:38.469738 AM	2	23	3	3,822	177	874
2	Dec 16, 2013 9:23:17.984555 AM	2	20	3	3,845	197	796
3	Dec 16, 2013 9:17:49.383506 AM	2	29	3	3,910	143	1,065
4	Dec 16, 2013 9:11:33.777138 AM	2	76	3	10,072	140	1,957
5	Dec 16, 2013 9:05:57.694349 AM	2	36	2	3,879	113	1,362
6	Dec 16, 2013 9:00:38.347428 AM	2	19	2	3,321	180	1,026
7	Dec 16, 2013 8:56:23.056008 AM	2	69	4	4,226	64	1,324
8	Dec 16, 2013 8:48:07.902735 AM	2	154	3	11,558	79	4,650
9	Dec 16, 2013 8:09:16.575288 AM	2	167	4	20,150	127	5,032
10	Dec 16, 2013 8:05:58.861928 AM	2	49	4	2,762	59	809
11	Dec 16, 2013 8:01:32.889947 AM	2	7	3	1,253	201	161

Savepoints

The following SQL shows a histogram on the critical phase duration:

```
select
to_char(SERVER_TIMESTAMP,'yyyy.mm.dd') as "time",
sum(case when (critical_phase_duration <= 1000000) then 1 else 0
end) as "<= 1 s",
sum(case when (critical_phase_duration > 1000000 and critical_phase_duration
<=2000000) then 1 else 0
end) as "<= 2 s",
sum(case when (critical_phase_duration > 2000000 and critical_phase_duration
<=3000000) then 1 else 0
end) as "<= 3 s",
sum(case when (critical_phase_duration > 3000000 and critical_phase_duration
<=4000000) then 1 else 0
end) as "<= 4 s",
sum(case when (critical_phase_duration > 4000000 and critical_phase_duration
<=5000000) then 1 else 0
end) as "<= 5 s",
sum(case when (critical_phase_duration > 5000000 and critical_phase_duration
<=10000000) then 1 else 0
end) as "<= 10 s",
sum(case when (critical_phase_duration > 10000000 and critical_phase_duration
<=20000000) then 1 else 0
end) as "<= 20 s",
sum(case when (critical_phase_duration > 20000000 and critical_phase_duration
<=40000000) then 1 else 0
end) as "<= 40 s",
sum(case when (critical_phase_duration > 40000000 and critical_phase_duration
<=60000000) then 1 else 0
end) as "<= 60 s",
sum(case when (critical_phase_duration > 60000000 ) then 1 else 0
end) as "> 60 s",
count(critical_phase_duration) as "ALL"
from "_SYS_STATISTICS"."HOST_SAVEPOINTS"
where volume_id in (select volume_id from m_volumes where service_name =
'indexserver')
and weekday (server_timestamp) not in (5, 6)
group by to_char(SERVER_TIMESTAMP,'yyyy.mm.dd')
order by to_char(SERVER_TIMESTAMP,'yyyy.mm.dd') desc;
```

SQL | Result

```
select
to_char(SERVER_TIMESTAMP,'yyyy.mm.dd') as "time",
sum(case
when (critical_phase_duration <= 1000000) then 1 else 0
end) as "<= 1 s",
sum(case
when (critical_phase_duration > 1000000 and critical_phase_duration <=2000000) then 1 else 0
```

	time	<= 1 s	<= 2 s	<= 3 s	<= 4 s	<= 5 s	<= 10 s	<= 20 s	<= 40 s	<= 60 s	> 60 s	ALL
1	2013.10.17	0	0	0	3	12	56	32	3	0	0	106
2	2013.10.16	5	0	1	5	15	90	97	9	0	0	222
3	2013.10.15	1	0	3	9	6	56	25	1	2	1	104
4	2013.10.14	1	1	7	25	20	115	67	3	1	3	243
5	2013.10.11	21	1	2	0	3	15	85	12	2	4	145
6	2013.10.09	0	0	0	0	0	4	25	12	1	0	42
7	2013.10.08	0	0	0	0	2	27	133	34	1	2	199
8	2013.10.07	2	0	6	3	20	89	72	15	4	3	214
9	2013.10.04	1	0	0	1	0	16	45	21	1	1	86
10	2013.10.03	1	0	2	2	6	37	52	18	1	0	119
11	2013.10.02	2	2	9	5	2	24	53	19	1	0	117
12	2013.10.01	1	0	0	0	2	28	62	15	1	0	109
13	2013.09.30	1	1	6	16	15	61	35	7	0	0	142
14	2013.09.27	2	2	5	9	11	83	33	7	0	1	153
15	2013.09.26	1	0	3	3	6	24	47	17	1	0	102
16	2013.09.25	1	0	0	3	4	38	61	21	2	0	130
17	2013.09.24	1	0	1	4	7	51	71	13	1	0	149
18	2013.09.23	3	5	8	10	8	55	46	7	1	0	143
19	2013.09.20	4	28	30	18	28	71	4	2	0	0	185
20	2013.09.19	1	6	10	24	28	98	7	0	1	0	175

Savepoint Histogram

The performance of the backup can be analyzed with this statement:

```
select mbc.backup_id,
SECONDS_BETWEEN (mbc.sys_start_time, mbc.sys_end_time) seconds,
round(sum(backup_size) / 1024 / 1024 / 1024,2) size_gb,
round(sum(backup_size) / SECONDS_BETWEEN (mbc.sys_start_time, mbc.sys_end_time) /
1024 / 1024, 2) speed_mbs
from m_backup_catalog_files mbcf , m_backup_catalog mbc
where mbc.entry_type_name = 'complete data backup'
and mbc.state_name = 'successful'
and mbcf.backup_id = mbc.backup_id
group by mbc.backup_id, mbc.sys_end_time, mbc.sys_start_time order by
mbc.sys_start_time
```

9.2 Replication Performance Problems

If system replication appears to slow down transaction processing, you can check the network and disk I/O on the secondary site.

A slow-down related to system replication can occur in the following scenarios:

- ASYNC replication mode is configured over long distances;
- multi-tier system replication is configured and a tier 3 system is attached;
- SYNC/SYNCMEM replication mode is configured over short distances.

The following troubleshooting steps can help you determine and resolve the underlying cause.

Check If Log Can Be Shipped in Time

You can check the system replication KPI values to analyze the problem and verify that it is really related to system replication:

- check if log shipping is significantly slower than local log write (SYNC/SYNCMEM)
- check Async Buffer Full Count (ASYNC)

You can check system replication KPIs in SAP HANA cockpit (see *Monitoring SAP HANA System Replication* in the *SAP HANA Administration Guide*). You can also get an overview of basic system replication KPIs by running the query *HANA_Replication_SystemReplication_Overview_*_MDC.txt* (from *SAP Note 1969700 - SQL Statement Collection for SAP HANA*). This query is based on the system view M_SERVICE_REPLICATION and can be used to compare log shipping time to local log write time. For synchronous replication the following KPIs are shown:

Example output of SQL Statement HANA_Replication_SystemReplication_Overview.txt

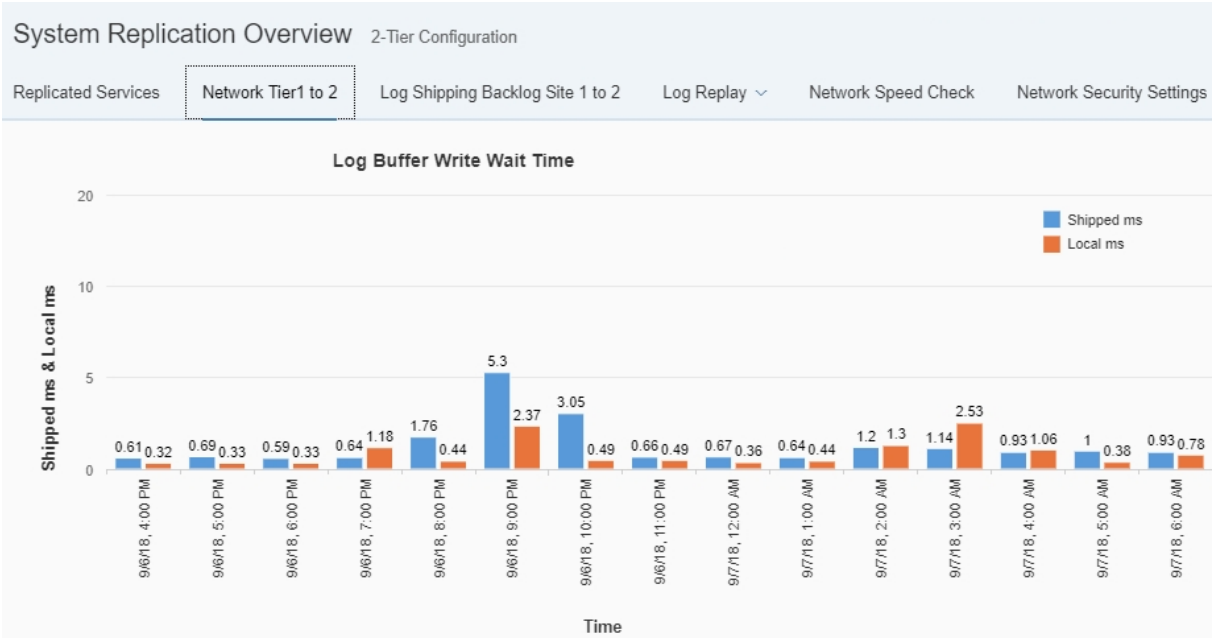
KEY	VALUE
Replication mode	SYNC
Secondary connect time	2016/10/18 14:02:36
Days since secondary connect time	1.26
Used persistence size (GB)	205.76
Log backup size / day (GB)	83.58
Local log buffer write size (MB)	101130.54
Shipped log buffer size (MB)	100258.52
Avg. local log buffer write size (KB)	6.14
Avg. shipped log buffer size (KB)	6.14
Avg. local log buffer write time (ms)	0.13
Avg. log buffer shipping time (ms)	0.24
Local log buffer write throughput (MB/s)	44.68
Log buffer shipping throughput (MB/s)	24.99
Initial data shipping size (MB)	0.00
Initial data shipping time (s)	0.00

KEY	VALUE
Last delta data shipping size (MB)	2736.00
Last delta data shipping time (s)	13.00
Delta data shipping size (MB)	758704.00
Delta data shipping time (s)	3538.20
Delta data shipping throughput (MB/s)	214.43
Delta data shipping size / day (MB)	n/a
Replication delay (s)	0.00

The following KPIs are of particular importance, the shipping time should not be significantly higher than the local write time:

- *Avg. local log buffer write time (ms)*
- *Avg. log buffer shipping time (ms)*

You can see a graphical comparison of these local and shipped values in the cockpit System Replication Overview (Network tab). The graph displayed compares the local write wait time with the remote write wait time monitored over the last 24 hours:



For asynchronous replication scenarios the redo log is written into an Asynchronous Log Buffer, which occasionally can run full in case the logs are not shipped in a timely manner to the secondary instance. This can lead to a performance overhead on the primary site as by default it waits with new COMMITS until there is free space in the buffer. This can be avoided by setting the parameter `logshipping_async_wait_on_buffer_full` in the system_replication section of the global.ini file to **`FALSE`**.

> **i Note**
>
> In order to maintain a stable connection during the initial data shipment this parameter should be set to true. This is recommended because if the log shipping connection is reset for any reason the data shipment connection is also reset and the initial data shipment has to start again from the beginning. In multi-tier scenarios a restarted full data shipment from primary to secondary site also results in a completely new full data shipment to a tertiary site. For the duration of the initial shipment, therefore, you may also increase the value of the `logshipping_timeout` parameter on the primary which has a default value of 30 seconds.

The size of the asynchronous log shipping buffer on the primary site is normally adequate; the default value of the `logshipping_async_buffer_size` parameter is 256MB for the indexserver (in the indexserver.ini) and 64MB for all other services (maintained in the global.ini). However, if additional free memory is available this value can also be increased for specific services with a high log generation (such as the indexserver). You should make such changes only in the service-specific ini files rather than in the global.ini file.

Once the asynchronous replication connection is established you can see how much is in the async buffer by checking the value of BACKLOG_SIZE in the system view M_SERVICE_REPLICATION. If there is no connection this column shows the number of log entries that have been generated on the primary but which have not yet reached the secondary. You can also see this information (the backlogSize) by running the following command on the primary with admin rights:

```
hdbcons 'replication info'
- backlogSize : 2781184 bytes
```

For further details of the most common performance issues caused by system replication and under which circumstances they occur, please refer to *SAP KBA 1999880 - FAQ: SAP HANA System Replication*.

Check If Data Load Can Be Handled by Network Link

To estimate the required bandwidth for Data/Log shipping, use *HANA_Replication_SystemReplication_Bandwidth.txt* (from *SAP Note 1969700 - SQL Statement Collection for SAP HANA*), which is based on the I/O statistics from the primary site. We recommend executing this SQL statement when system replication is disabled. The data returned will help you to estimate the amount of data/log shipped from the primary site and compare this to the available bandwidth.

You can also do a network performance test using, for example, the open source IPERF tool or similar, to measure the real application network performance. The recommended bandwidth is 10 Gbit/s.

If the network bandwidth is not adequate you can activate data and log compression which significantly reduces the shipment size by setting the following parameters in the system_replication section of `global.ini`:

- enable_log_compression = TRUE
- enable_data_compression = TRUE

Check Network Configuration (Long Distance)

Increasing the TCP window size can result in better network utilization and higher throughput. If the bandwidth can handle load, check if the network is shared and whether other applications may be interfering with performance. Collect network information on bandwidth and latency from the Linux kernel parameters as described here. For these values refer also to SAP Note 2382421 - *Optimizing the Network Configuration on HANA and OS-Level (Linux Kernel Parameters)* :

- Check the network utilization profile for the network link to see if the maximum capacity of the network has been reached.
- If the network is not fully utilized, check the linux kernel TCP configuration with `sysctl –a | egrep "net.core|net.ipv4.tcp" .`
- Check that window scaling is set to the default value of 1. `net.ipv4.tcp_window_scaling = 1`.
- Check whether the max size can be increased for `net.ipv4.tcp_wmem` and `net.ipv4.tcp_rmem`.
- Calculate the Bandwidth Delay Product (BDP): Bandwidth * Latency (for example, BDP = 50ms * 3 Gbps = 19.2 MB). The BDP tells you what TCP window size is needed to use the network link fully.

Check Disk I/O on a Secondary Site

Slow disk I/O on the secondary can postpone releasing log buffers on the primary, which results in wait situations on the primary. You can do the following:

- Use a Disk Performance Test Tool
  Execute fsperf on log volume, for example:
  `$ fsperf /usr/sap/TST/SYS/global/hdb/log/mnt00001/hdb00002`
- Check the *Monitoring and Administration* area
  If SQL is not available, use command line tools (this has to be done for each individual service), for example:
  `$ hdbcons "statreg print -n M_VOLUME_IO_TOTAL_STATISTICS -h"`
  A runtime dump also contains I/O statistics, which you can see with: `$ hdbcons "runtimedump dump"`.

  > ⚠ Caution
  >
  > Technical expertise is required to use hdbcons. To avoid incorrect usage, use `hdbcons` only with the guidance of SAP HANA development support.

- Check I/O relevant tables in the proxy schema of the corresponding secondary site, which provide SQL access on the primary site on statistic views of the secondary. For more information, see *Monitoring Secondary Sites* in the *SAP HANA Administration Guide*.

Related Information

SAP Note 1969700
SAP Note 1999880
SAP Note 2382421

Monitoring Secondary Systems [page 163]
Monitoring SAP HANA System Replication in SAP HANA Cockpit [page 182]

9.3 Setup and Initial Configuration Problems

This section outlines the analysis steps you need to take in case you face configuration problems during the initial HANA System Replication Setup.

The initial SAP HANA System Replication Setup steps are as follows:

- enabling the SAP HANA System Replication on the primary site with `sr_enable`
- registering the secondary system with `sr_register`

While there are no errors to be expected when you enable the primary site, the registration operation on the secondary site can fail due to various errors.

> **i Note**
>
> If you are in the process of setting up HANA System Replication for the first time, please make sure you have met all the prerequisites and performed the necessary preparation steps, outlined in the *SAP HANA Administration Guide*.
>
> Pay special attention to the following points:
>
> - Are the primary and secondary sites architecturally identical?
> - Are the network interface configurations identical on both sites? (refer to the SCN document *How to Configure Network Settings for HANA System Replication* for details).
> - Are the ports needed for system replication open and reachable from the primary and the secondary site?

Wrong Topology Information

Upon registering the secondary site, the following error is raised:

> **Output Code**
>
> ```
> > hdbnsutil -sr_register
> --remoteHost=primary_host --remoteInstance=<primary_instance_no> --
> mode=<replication_mode>
> --name=<logical_site_name>
> adding site ...
> checking for inactive nameserver ...
> nameserver primary_host:3xx01 not responding.
> collecting information ...
> error: source system and target system have overlapping logical hostnames;
> each site must have a unique set of logical hostnames.
> hdbrename can be used to change names;
> failed. trace file nameserver_primary_host.00000.000.trc may contain more
> error details.
> ```

The root cause for those issues is usually a wrong topology information. In this case, the secondary site contained the following landscape definition in the `nameserver.ini`:

Sample Code

```
[landscape]
id = <id>
master = <secondary_host>:3xx01
worker = <primary_host>
active_master = <secondary_host>:3xx01
roles_<primary_host> = worker
```

The `worker` property contained the hostname of the primary site, which was wrong. Therefore, the registration failed. The problem should disappear once the correct hosts are maintained in the `master` and `worker` (if any) properties. You need to check on both sites if the information maintained in the nameserver topology is consistent.

Resyncing the Secondary: Persistence Compatibility Checks

If the primary and secondary systems are disconnected for any reason, they must be resynced. If the persistencies (that is, the data and log volume snapshots) of the primary and secondary are compatible, it is possible to achieve a resync with only a delta data shipment or a log shipment; in this case full data shipping is not necessary. Even if the data snapshots are not compatible, the system will automatically attempt a full data shipment (*Resync Optimization*). If necessary, a full data shipment can be triggered manually using the following command:

```
hdbnsutil -sr_register --force_full_replica
```

Trace messages related to persistence which indicate that this is necessary include the following:

- Secondary persistence is not compatible with primary persistence.
- The persistence of at least one service is not initialized correctly.

Communication Problems with the Primary Site

The `sr_register` command on the secondary site is failing with:

Output Code

```
> hdbnsutil -sr_register --name=<logical_site_name> --
remoteHost=<primary_host --remoteInstance=<primary_instance_no> --
mode=<replication_mode> --force_full_replica --sapcontrol=1
unable to contact primary site host <primary_host>:3xx02. connection
refused,location=<primary_host>:3xx02
```

Possible Root Cause 1: Ports Not Open / Blocked by Firewall

This error usually indicates a general communication problem between the primary and secondary site. Mostly, this is caused by the primary host not listening on the required ports for various reasons. You can check

whether the required ports `3<instance_number>01` and `3<instance_number>02` (non-MDC scenarios) or `4<instance_number>02` (MDC scenarios) are listening on the required interfaces with the following command on OS level as privileged user (for example, root):

```
>netstat -apn | grep 3<instance_no>02
>netstat -apn | grep 4<instance_no>02
```

If you see that these ports are open and listening on the localhost interface only, you will not be able to reach them from the secondary site. You need to adjust the settings for `listeninterface` in the `global.ini` file from `.local` to `.global`:

Sample Code

```
[communication]
listeninterface=.global
```

With this setting, the following interface:port pairs should be visible in netstat:

```
tcp        0      0 0.0.0.0:30101           0.0.0.0:*               LISTEN      4273/hdbnameserver
tcp        0      0 10.          :30102     0.0.0.0:*               LISTEN      4273/hdbnameserver
tcp        0      0 127.0.0.2:30102         0.0.0.0:*               LISTEN      4273/hdbnameserver
tcp        0      0 127.0.0.1:30102         0.0.0.0:*               LISTEN      4273/hdbnameserver
tcp        0      0 0.0.0.0:30103           0.0.0.0:*               LISTEN      5278/hdbindexserver
tcp        0      0 0.0.0.0:30107           0.0.0.0:*               LISTEN      5280/hdbxsengine
```

Note

If the ports are open, check whether they are not filtered by your firewall. Often it is not sufficient to check the connectivity to remote hosts via ping, because ping uses the ICMP protocol for communication. You can easily verify the accessibility of remote hosts by issuing a telnet call. For example:

```
>telnet <primary_host> 30001
>telnet <primary_host> 30102
```

Possible Root Cause 2: SSL-Related Problems

Another cause for this error could be a wrongly implemented SSL configuration.

Note

If you do not secure the HANA network with SSL, do not implement any parameter changes related to SSL.

This can be revealed by activated corresponding traces on the primary site via SAP HANA cockpit:

- *Database Explorer* > *Trace Configuration* > *Database Trace* > *Search for "sr_nameserver"* > *Change from INFO to DEBUG* > *OK*
- *Database Explorer* > *Trace Configuration* > *Database Trace* > *Search for "trexnet"* > *Change from ERROR to INFO* > *OK*

Alternatively, the traces can be activated in the SQL console by issuing the following statements as a SYSTEM user:

Source Code

```
alter system alter configuration ('indexserver.ini','SYSTEM') SET
('trace','sr_nameserver')='debug' with reconfigure;
```

```
alter system alter configuration ('indexserver.ini','SYSTEM') SET
('trace','trexnet')='info' with reconfigure;
```

After the trace activation, the registration problem needs to be reproduced by re-running the `sr_register` command on the secondary. The nameserver trace on the primary site would reveal the following errors in the CommonCrypto Engine:

Output Code

```
Crypto/SSL/CommonCrypto/Engine.cpp:563: SSL handshake failed: SSL error
[536871970]: Unknown error, General error: 0x20000422 | SAPCRYPTOLIB |
SSL_accept
SSL API error
Version in SSLPlaintext.version field of currently received record differs
from
the one negotiated in the current or currently accomplished handshake.
0xa060023c | SSL | ssl3_accept
Version in SSLPlaintext.version field of currently received record differs
from
the one negotiated in the current or currently accomplished handshake.
0xa060023c | SSL | ssl3_get_record
Version in SSLPlaintext.version field of currently received record differs
from
the one negotiated in the current or currently accomplished handshake.
(ErrCode: 536871970)
```

Make sure the following parameters are consistent on both sites in the configuration file `global.ini`:

Sample Code

```
[communication]
ssl = systempki
..
...
[system_replication_communication]
enable_ssl = on
```

You need to ensure that the SSFS key and data files are stored on both sites. The following files must exist on both sites:

```
$DIR_INSTANCE/../global/security/rsecssfs/data/SSFS_<SID>.DAT
$DIR_INSTANCE/../global/security/rsecssfs/data/SSFS_<SID>.KEY
```

Possible Root Cause 3: Wrong Configuration of the Internal Hostname Resolution Parameters

Please check whether the internal hostname resolution information is consistent on both sites. The following how-to guides are a good source of information:

- *How to Configure Network Settings for SAP HANA System Replication*
- *How to Perform System Replication for SAP HANA*.

Possible Root-Cause 4: Wrong MTU Size Configured

A closer look at the nameserver trace file on the secondary site would reveal:

Output Code

```
error: unable to contact primary site; to <primary_host_ip> (<primary_host>):
3xx01; original error: timeout occured,location=<primary_host_ip> :3xx02. Was
MTU size set to 1500? (https://css.wdf.sap.corp/sap/support/notes/2142892);
```

This problem is discussed in full detail in *SAP Note 2166157 - Error: 'failed to open channel ! reason: connection refused' when setting up Disaster Recovery*.

Possible Root Cause 5: HANA Service Unavailability

Check the availability of the indexserver / nameserver process on the primary site. Often the services faced an intermittent restart, crash or reconfiguration which did not go unrecognized by the secondary site.

Related Information

SAP HANA Administration Guide
Configure Tracing in the SAP HANA Database Explorer
How to Perform System Replication for SAP HANA
How to Configure Network Settings for SAP HANA System Replication
SAP Note 2166157

9.4 Intermittent Connectivity Problems

This section discusses the mitigation strategies for sporadic network interruptions causing problems in the SAP HANA System Replication mechanism.

A common intermittent error is that the log buffer is not shipped in a timely fashion from the primary to the secondary site.

Log Shipping Timeout

Possible Root Cause 1: Log Area Is Full on the Secondary Site

If the System Replication Mode is set to `SYNC – full sync`, the commits on primary are halted as nothing can be written to the log area on secondary site any longer. On the secondary site, the trace files contain the following error:

Output Code

```
i EventHandler LocalFileCallback.cpp(00455) : [DISKFULL] (1st request) [W] ,
buffer= 0x00007f7eef8ae000, offset= 589299712, size= 0/524288, file= "<root>/
logsegment_000_00000508.dat

" ((open, mode= RW, access= rw-rw-r--, flags= DIRECT|LAZY_OPEN), factory=
(root= "/hana/log/<SID>/mnt00001/hdb00003/" (access= rw-rw-r--, flags=
AUTOCREATE_DIRECTORY, usage= LOG, fs= xfs,

config=
(async_write_submit_active=auto,async_write_submit_blocks=new,async_read_submi
t=off,num_submit_queues=1,num_completion_queues=1,size_kernel_io_queue=512,max
_parallel_io_requests=64,

min_submit_batch_size=16,max_submit_batch_size=64))) {shortRetries= 0,
fullRetries= 0 (0/10)}
```

To quickly mitigate the situation, you can disable the “full sync” option by running the following command:

```
>hdbnsutil -sr_fullsync --disable
```

Afterwards, the log area on the secondary site needs to be analyzed with regard to why the log segments are not freed up. This is usually caused by an erroneous log backup mechanism.

For further details refer to the following SAP Notes:

SAP Note 2083715 - Analyzing log volume full situations

SAP Note 1679938 - Log Volume is full

Possible Root Cause 2: Sporadic Communication Issues on the Network Layer

For more information about how to deal with communication problems between the primary and the secondary site, see *SAP HANA System Replication Communication Problems*.

Related Information

SAP Note 2083715

SAP Note 1679938

SAP HANA System Replication Communication Problems [page 237]

9.5 LogReplay: Managing the Size of the Log File

There is a risk in replication scenarios which use one of the logreplay operation modes of causing a disk full situation on the primary if the secondary system is not available for any reason; this can potentially lead to a complete freeze of the database.

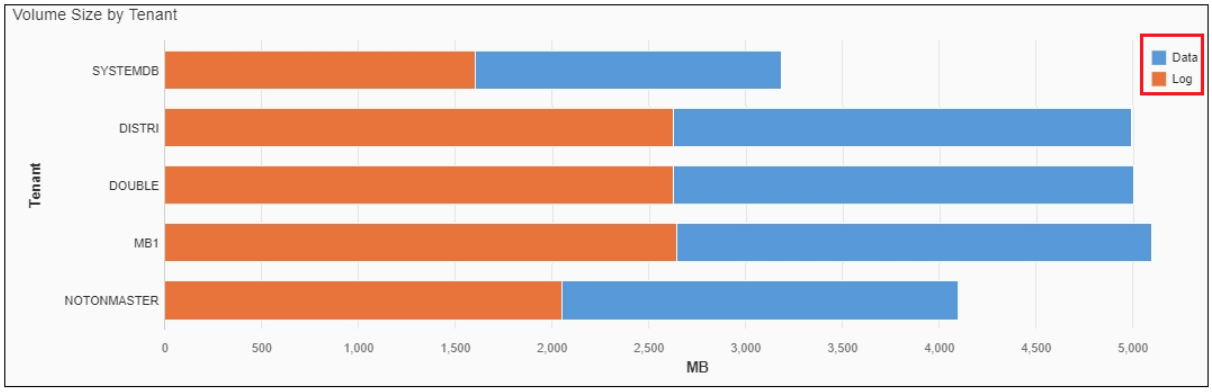
The *logreplay* modes (logreplay introduced in HANA 1.0 SPS10 and *logreplay_readaccess* introduced in HANA 2) require a log history on the primary so that a secondary system can be resynchronized without the need for a full data shipment. As long as a secondary system is registered the log file will continue to grow. When the secondary system synchronizes, then the log is automatically cleared down. However, if the replication environment changes, if for example, the secondary is separated because of network problems, manual intervention may be required to manage the log file or, in the worst case scenario, to recover from a disk full situation. This problem can also happen on a secondary system where a takeover has occurred.

The log replay modes are described in detail in the *SAP HANA Administration Guide* section *System Replication With Operation Mode Logreplay*. This section of the *SAP HANA Troubleshooting and Performance Analysis Guide* describes procedures to firstly prevent problems from occurring and secondly to resolve a disk full situation.

Log File Retention (RetainedFree Status)

If the secondary cannot be synchronized for any reason, then log segments continue to be written but are marked as *RetainedFree*. You can check for RetainedFree log segments either in SAP HANA cockpit or from the command line.

1) To check using SAP HANA cockpit, start from the Disk Usage app and open *Monitor Disk Volume*. The graph shows usage for log and data volumes; you can filter the display for a specific volume and server (for example indexserver). Check the *State* column of the log files for RetainedFree log segments as shown here:



Volume ID	Service	Type	Size (MB)	Used (MB)	Used (%)	State	Files	Path	Host	Database Name
3	indexserver	LOG	1,024.0	1,024.0	100.0	Writing	1	/hana/log/MB1/mnt00002/hdb00003.00004	ha-test-02.mo.sap.corp	DISTRI
2	xsengine	LOG	8.0	8.0	100.0	Writing	1	/hana/log/MB1/mnt00001/hdb00002.00003	ha-test-01.mo.sap.corp	MB1
2	indexserver	LOG	1,024.0	1,024.0	100.0	RetainedFree	1	/hana/log/MB1/mnt00002/hdb00002.00006	ha-test-02.mo.sap.corp	NOTONMASTER
1	nameserver	LOG	64.0	64.0	100.0	RetainedFree	1	/hana/log/MB1/mnt00001/hdb00001	ha-test-01.mo.sap.corp	SYSTEMDB
3	indexserver	LOG	1,024.0	1,024.0	100.0	Writing	1	/hana/log/MB1/mnt00001/hdb00003.00003	ha-test-01.mo.sap.corp	MB1
2	indexserver	LOG	1,024.0	1,024.0	100.0	Writing	1	/hana/log/MB1/mnt00001/hdb00002.00005	ha-test-01.mo.sap.corp	DOUBLE

2) To check using the command line, execute the following command as <sid>adm for a specific log volume (hdb00003 in this example – the log volume of one indexserver):

```
#>hdblogdiag seglist $DIR_INSTANCE/../SYS/global/hdb/log/mnt00001/hdb00003
```

The result shows details of each log segment including status information. Look for any segments with status RetainedFree as shown here:

```
LogSegment[0/2:0xec98740-0xecb6000(0x763000B)/
GUID=759DC14B-00D7-20161122-134436-39A00002ED/
PrevGUID=759DC14B-00D7-20161122-134436-39A00002EC,TS=2016-11-30 06:55:18.008741,
Hole=0xec98740/RetainedFree/0x0]@0x00007f34cb32a010
```

Maximum Number of Segments

A further possible cause of a log full event is if the maximum number of allocated log segments is reached. By default, the maximum number is 10240 segments, if this limit is reached then log writing is blocked and the system may hang as though the disk is full even though disk space is still available. You can change the maximum number of log segments using the `hdblogdiag` tool as described in SAP Note 2072410 *Enlarge limitation of log segment number of LogSegment Directory*.

How to Avoid Log Full Situations

Unregister an Unused Secondary

If the secondary is disconnected for a prolonged period and if it is not to be used as a replication server anymore, then unregister the secondary site and disable the primary functionality. This will stop the RetainedFree log entries from being written:

1. Unregister the secondary; this is normally done from the secondary site but can be done from the primary if the secondary is not available anymore:
   `hdbnsutil –sr_unregister`
2. Disable the primary (from the primary site):
   `hdbnsutil –sr_disable`
3. Execute reconfigure on the primary site:
   `hdbnsutil –reconfig`

You can use this same procedure for a primary which, after a takeover, will no longer be used for failback.

Set a Maximum Retention Size

Another option to manage the log size is to set a value for the `logshipping_max_retention_size` parameter. If the log size reaches this limit, then RetainedFree log entries will be overwritten. Note the following points in this case:

- If any RetainedFree log entries are lost, then synchronization by logreplay will no longer be possible and a full data shipment will be necessary to resynchronize the secondary.
- It is not possible to switch back to delta mode to resynchronize - only a full data shipping is possible.

→ Tip

As a further general precaution, to prevent any disk full situation from arising you can reserve a portion of the disk with an emergency placeholder file (containing any dummy values), for example, occupying 5 – 10 % of the file system. This file can then be deleted if ever necessary to quickly solve disk full situations.

How to Recover From Log Full Situations

Secondary Has Been Taken out of Service

If the secondary has been permanently taken out of service, then these log entries will never be required. In this case the secondary can be unregistered and the log volume cleaned up:

1. Unregister the secondary (same steps as previous subsection: unregister, disable and reconfigure).
2. Delete the Free marked log segments from the command line for each of the persistent relevant services (nameserver, indexserver, xsengine, ...). To do this run hdbcons with the `log release` parameter as <sid>adm. In a multi-database system the -p switch is required with the process ID of the service (such as indexserver):
   `hdbcons -p <PID_of_service> "log release"`

Secondary Still Required

If the secondary is still required, then restart it and allow it to resynchronize. When this has completed, the RetainedFree log segments on the primary will be marked as Free, you can then clean up the log as described above by running hdbcons with the `log release` parameter.

Log Full Has Caused Complete Database Freeze

If the log full has caused a complete database freeze, you can try to move the log to another linked file system and replay the log from there. Essentially, this is a three step procedure, refer to SAP Note 1679938 *Log Volume is full* for complete details:

- Stop the primary system.
- Mount the log volumes of the primary via symbolic link to another file system.
- Start the primary and the secondary and allow them to resynchronize.

When this has completed, you can then clean up the log by running hdbcons with the `log release` parameter as described above.

Related Information

SAP Note 1679938
SAP Note 2072410

9.6 SAP HANA System Replication Communication Problems

Problems during initial setup of the system replication can be caused by incorrect configuration, incorrect hostname resolution or wrong definition of the network to be used for the communication between the replication sites.

Context

System replication environments depend on network bandwidth and stability. In case communication problems occur between the replication sites (for example, between SITE A and SITE B), the first indication of a faulty system replication setup will arise.

Procedure

1. Check the nameserver tracefiles.

   The starting point for the troubleshooting activity are the nameserver tracefiles. The most common errors found are:

   Sample Code

   ```
   e TNS TNSClient.cpp(00800) : sendRequest dr_secondaryactivestatus to
   <hostname>:<system_replication_port> failed with NetException.
   data=(S)host=<hostname>|service=<service_name>|(I)drsender=2|
   e sr_nameserver TNSClient.cpp(06787) : error when sending request
   'dr_secondaryactivestatus' to <hostname>:<system_replication_port>:
   connection broken,location=<hostname>:<system_replication_port>
   e TrexNetBuffer BufferedIO.cpp(01151) : erroneous channel ### from #####
   to <hostname>:<system_replication_port>: read from channel failed;
   resetting buffer
   ```

   Further errors received from the remote side:

   Sample Code

   ```
   Generic stream error: getsockopt, Event=EPOLLERR - , rc=104: Connection
   reset by peer
   Generic stream error: getsockopt, Event=EPOLLERR - , rc=110: Connection
   timed out
   ```

   It is important to understand that if those errors suddenly occur in a working system replication environment, they are often indicators of problems on the network layer. From an SAP HANA perspective, there is nothing that could be toggled, as it requires further analysis by a network expert. The investigation, in this case, needs to focus on the TCP traffic by recording a tcpdump in order to get a rough understanding how TCP retransmissions, out-of-order packets or lost packets are contributing to the overall network traffic. How a tcpdump is recorded is described in *SAP Note 1227116 - Creating network*

traces. As these errors are not generated by the SAP HANA server, please consider consulting your in-house network experts or your hardware vendor before engaging with SAP Product Support.

2. Set the parameter `sr_dataaccess` to debug.

   In the *DB Administration* area of the SAP HANA cockpit open the *Configuration of System Properties* monitor. In the *[trace]* section of the `indexserver.ini` file set the parameter `sr_dataaccess` = **debug**. This parameter enables a more detailed trace of the components involved in the system replication mechanisms. For more information about how to change parameters, see *Memory Information from Logs and Traces*.

Related Information

SAP Note 1227116
Memory Information from Logs and Traces

9.7 Stress Test with NIPING

The SAP NIPING tool is a powerful tool which can be used to perform specific network stability tests.

Prerequisites

You must have OS level access to the SAP HANA host and the client host.

Procedure

Read *SAP Note 500235 - Network Diagnosis with NIPING*.

A stress test with SAP's NIPING tool may be performed in order to confirm the high network latency (or bandwidth exhaustion).

Related Information

SAP Note 500235

10 Security Aspects for SAP HANA System Replication

Learn about the secure operation and configuration of SAP HANA system replication.

Which security aspects are addressed in this guide?

Communication between sites in a system replication scenario is always authenticated. In addition, it is possible to secure internal network communication between primary and secondary systems using TLS/SSL. You can learn how to configure TLS/SSL on communication channels between primary and secondary systems using the system public key infrastructure (PKI).

To integrate SAP HANA securely into your network environment, several general recommendations apply. Learn how to protect the channels used in a system replication scenario.

Where can I find more information?

The following SAP Notes are relevant for a full understanding of the concepts described in this chapter:

SAP Notes

SAP Note	Title
2175672	Migration steps from manual SSL configuration for internal communication to automatic configuration using system PKI
2356851	SAP HANA Dynamic Tiering Support for SAP HANA System Replication
2447994	SAP HANA Dynamic Tiering Support for SAP HANA System Replication

Related Information

Secure Internal Communication [page 240]
Secure Internal Communication Between Sites in System Replication Scenarios [page 244]
Legacy Configuration of Secure Internal Communication [page 245]
Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites [page 247]
Communication Channels [page 249]
Network Security [page 251]
Internal Application Encryption Service [page 254]

10.1 Secure Internal Communication

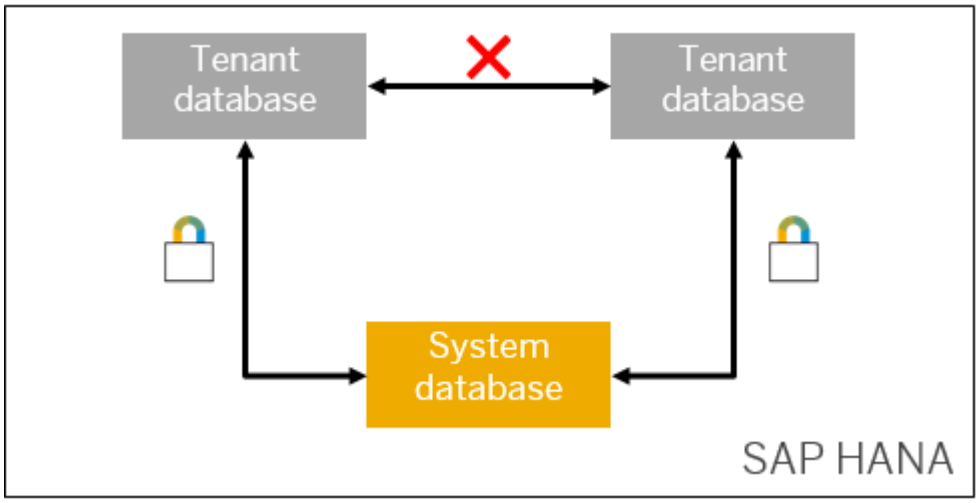
All internal SAP HANA communication can be secured using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol. A simple public-key infrastructure (PKI) is set up during installation for this purpose.

Internal Communication Channels

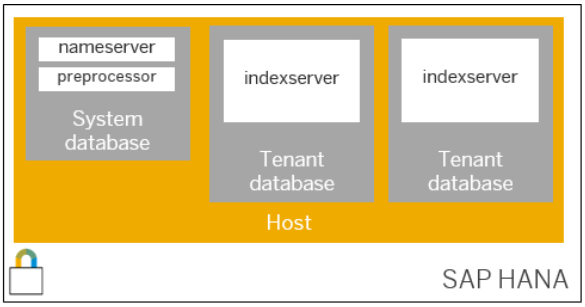
The internal communication channels shown below can be secured using TLS/SSL.

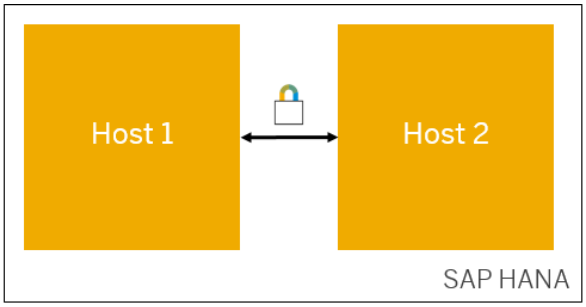
> **i** Note
>
> SAP HANA internal communication has sometimes been unofficially referred to as TREXNet communication. However, the term TREXNet is not valid in the context of SAP HANA.

Communication between databases



Communication between the hosts in a multiple-host system and between processes on a single host

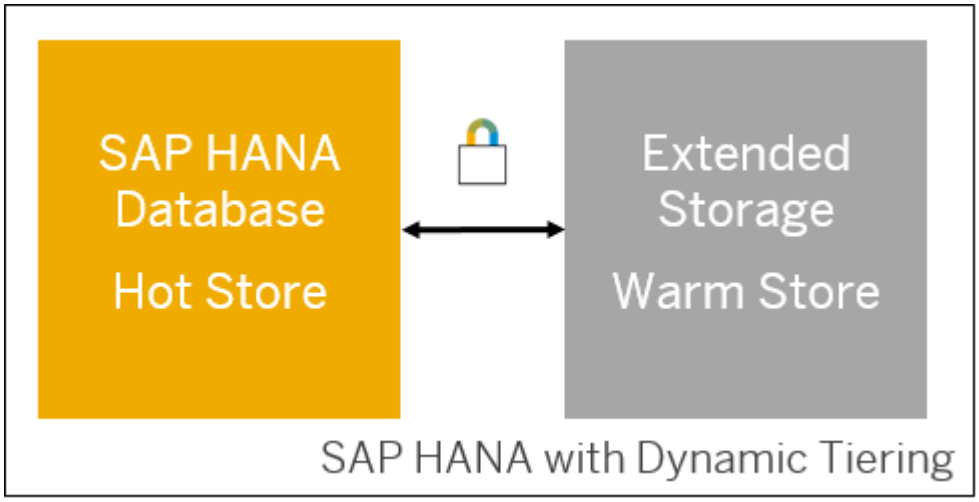




Communication between the sites in a system with system replication enabled

Communication between the SAP HANA database and additional server components

For example: Extended storage server (SAP HANA dynamic tiering) or a streaming analytics server (SAP HANA Streaming Analytics).



System Public Key Infrastructure

A dedicated PKI is created for internal communication automatically during system installation. Every host on which a database server and optional component server is running, as well as every tenant database in the system, are integrated into this PKI, which uses CommonCryptoLib as the cryptographic library.

Each host and database receive a public and private key pair and a public-key certificate for mutual authentication. These certificates are all signed by a dedicated trusted certificate authority (CA) that is unique to the SAP HANA instance. The root personal security environment (PSE) file is stored in the system PKI SSFS (secure store in the file system). All other PSEs are encrypted with an automatically generated random PIN and stored in the file system. Certificates are automatically renewed when they expire.

> **i** Note
>
> A unique master key that protects the system PKI SSFS is generated during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that

you change it immediately after handover to ensure that it is not known outside of your organization. For more information about how to change the SSFS master keys, see the *SAP HANA Administration Guide*.

To secure internal communication between hosts and sites, you can set up and configure your own PKI, but we recommend you use the system PKI. The system PKI is always used to secure communication within tenant databases and communication with optional server components.

> **i** Note
>
> If high isolation is configured for tenant databases, the system PKI **must** also be used to secure communication between hosts.

For more information about migrating to the system PKI from a manually configured PKI, see SAP Note 2175672.

TLS/SSL Configuration Using System PKI

No interaction is required to set up the system PKI, but you may need to explicitly enable TLS/SSL depending on the channel as follows:

Communication Channel	Configuration Required to Enable TLS/SSL
Communication between the processes of individual databases	Configure the system for high isolation. High isolation requires that the processes of individual databases run under dedicated operating system (OS) users in dedicated OS groups. In addition, it enables certificate-based authentication so that only the processes belonging to the same database can communicate with each other. If you also want data communication within databases to be encrypted, you must change the value of the property `[communication] ssl` in the `global.ini` from **`off`** to **`systemPKI`**. If the property `ssl` is not visible (for example, in the SAP HANA studio), add the key **`ssl`** with the value **`systemPKI`** to the section `communication`. → Remember Change (or add) the property in the system database in the `SYSTEM` layer of the configuration file. **i** Note If cross-database access is enabled, communication between configured tenant databases is allowed. For more information about how to configure a system for high isolation, see the *SAP HANA Administration Guide*.

Communication Channel	Configuration Required to Enable TLS/SSL
Communication between hosts in a multiple-host system and localhost communication	Enable TLS/SSL manually. In the `global.ini` configuration file, change the value of the property `[communication] ssl` to **`systemPKI`**. This configuration ensures that only hosts belonging to the same system can communicate with each other and that all data communication between hosts is encrypted. **i Note** In a system that is not configured for high isolation, you can still enable secure communication between hosts. Remember to change the property in the system database in the `SYSTEM` layer. Enabling secure communication between hosts automatically enables secure communication between processes on the same host without any further configuration. Note the following: • If you are operating a single-host and require secure localhost communication, you must still enable TLS/SSL for inter-host communication as described above. • If you have enabled TLS/SSL for inter-host communication as described above, but do not require secure localhost communication, you can change the value of the property `[communication] ssl_local` from **`on`** to **`off`**.
Communication between sites in a system with system replication enabled	Several steps are required to enable TLS/SSL for the communication channel used for system replication. For more information, see *Secure Internal Communication Between Sites in System Replication Scenarios*.
Communication between the SAP HANA database and additional server components	No configuration required TLS/SSL is automatically enabled and cannot be disabled.

Related Information

Server-Side Secure Stores
Change the SSFS Master Keys
Database Isolation
Increase the System Isolation Level
Server-Side TLS/SSL Configuration Properties for Internal Communication
Secure Internal Communication Between Sites in System Replication Scenarios [page 244]
Legacy Configuration of Secure Internal Communication [page 245]
SAP Note 2175672

10.2 Secure Internal Communication Between Sites in System Replication Scenarios

Communication between sites in a system replication scenario is always authenticated. In addition, it is possible to secure internal network communication between primary and secondary sites using TLS/SSL.

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 (or systems), including in-memory loading, system replication facilitates rapid failover in the event of a disaster. Production operations can be resumed with minimal downtime.

Communication between the sites in a system replication landscape must be secured. The system PKI (public key infrastructure) that is automatically created during system installation is the default and recommended mechanism for communication. No interaction is required to set up the system PKI. However, you can also set up and configure your own PKI (see *Legacy Configuration of Secure Internal Communication*).

> → Remember
>
> 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.

Configuring Authentication Between Sites

To ensure that only configured systems in a system replication landscape can communicate with each other, SAP HANA uses certificate-based authentication based on the system PKI. To establish trust between systems, you must copy the system PKI SSFS data file and key file from the primary system to the same location on the secondary system(s). These files can be found at the following locations:

- `$DIR_INSTANCE/../SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT`
- `$DIR_INSTANCE/../SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY`

Copy the files before you register the secondary system with the primary system.

Configuring TLS/SSL-Secured Communication

In addition to authenticated communication, it is also possible to secure the following communication channels between primary and secondary systems 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

For more information on how to enable TLS/SSL on these communication channels, see the *SAP HANA Administration Guide*.

> **i Note**
>
> On SAP HANA systems with dynamic tiering, the same configuration applies. No additional steps are required. However, before you configure communication for dynamic tiering, see SAP Note 2356851.
>
> Be aware that you need additional licenses for SAP HANA options and capabilities. For more information, see Important Disclaimer for Features in SAP HANA.

Additional Network Security

You can further secure communication between sites by configuring SAP HANA to use exclusively a separate network dedicated to system replication for communication between primary and secondary sites.

It is also recommended that you configure the IP addresses of those hosts that are allowed to connect to the ports required for system replication.

For more information, see also the section on network security and the section on host name resolution in the *SAP HANA Administration Guide*.

Related Information

Legacy Configuration of Secure Internal Communication [page 245]
Server-Side TLS/SSL Configuration Properties for Internal Communication
Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites [page 247]
Host Name Resolution for System Replication
Network Security [page 251]
Secure Internal Communication [page 240]
SAP Note 2356851

10.3 Legacy Configuration of Secure Internal Communication

Although it is recommended that you use the system PKI (public key infrastructure) that is automatically created during system installation to secure internal communication channels, you can set up and configure your own PKI. This manually configured PKI is also used if system replication is configured for the system.

TLS/SSL Configuration for Communication Between Hosts

Since a host can both initiate a connection with another host (client role) as well as be the target of a connection initiated by another host (server role), every host in the system requires a public and private key

pair, and a public-key certificate (server certificate) with which it can identify itself to other hosts. Each host also needs the certificate or certificates with which it can validate the identity of other hosts. Typically, this is the root certificate or the certificate of the certification authority (CA) that signed the other hosts' certificates.

> **i** Note
>
> SAP HANA dynamic tiering does not support legacy configuration (using a manually configured PKI). If you are using SAP HANA dynamic tiering, use the system PKI configuration.

Use CommonCryptoLib as the cryptographic library. It is installed by default as part of SAP HANA server installation.

To manually configure secure communication between hosts:

1. Create a CA for the SAP HANA installation using external tools, for example, the OpenSSL command line tool.
   We recommend that you use a dedicated CA to sign all certificates used. We recommend storing your CA certificate in `$DIR_INSTANCE/ca`. This is typically the root certificate.

   > → Recommendation
   >
   > Create one private CA for each SAP HANA host. Do not use public CA for securing internal SAP HANA communication.

2. On every host, create the required server certificates.
   Every host is verified with its fully qualified domain name (FQDN). The common name (CN) must be the FQDN of the host you get by reverse DNS look-up. The other fields describe your organization.
3. Sign the certificates with the CA.
4. On every host, create a local keystore named `sapsrv_internal.pse` in directory `$SECUDIR` and import the private key and certificate, and the CA certificate (or root certificate).
   In the `communication` section of the file `global.ini`, create the property `ssl` with the value **`on`**.

TLS/SSL Configuration for Cross-Site Communication in System Replication Scenarios

In a system with system replication enabled, communication between sites (metadata and data channels) can be secured using the same configuration described above. For the data communication, you also need to enable SSL with the property `[system_replication_communication] enable_ssl` in the `global.ini` configuration file. For more information, see *Secure Internal Communication Between Sites in System Replication Scenarios*.

Keystore Configuration

The `[communication] sslInternalKeyStore` parameter in the `global.ini` configuration file specifies the path to the keystore file that contains the certificates for the following internal communication channels:

- Communication between hosts

- Communication between sites in system replication scenarios (data communication channel).

The default value is `$SECUDIR/sapsrv_internal.pse`.

Related Information

Secure Internal Communication Between Sites in System Replication Scenarios [page 244]

10.4 Configure Secure Communication (TLS/SSL) Between Primary and Secondary Sites

Configure TLS/SSL on communication channels between primary and secondary 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

The following communication channels between primary and secondary systems can be secured 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.

> **i** Note
>
> On SAP HANA systems with dynamic tiering, the following steps also enable the system PKI for internal system replication communication. No additional steps are required. 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

1. Shut down all systems.

   If you want to avoid downtime when enabling TLS/SSL, disable system replication. You can enable or disable TLS/SSL without downtime only if the primary system is not enabled.

2. In the primary and secondary system, enable TLS/SSL for the data channel.

   In the `global.ini` file, configure the property `[system_replication_communication] enable_ssl`. The following values are possible:

Value	Description
off (default)	TLS/SSL is disabled for replication source and target systems
on	TLS/SSL is enabled for replication source and target systems

> **i Note**
>
> You must enable SSL for the whole system, that is, in the global.ini file of the system database. Setting this feature for single tenant databases is not supported.

For a simple system replication scenario involving two systems, it is sufficient to 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 site_name 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
```

> **i Note**
>
> To avoid communication failure between systems, TLS/SSL must be enabled on all systems at the same time. TLS/SSL won't be used unless the secondary system reconnects with the primary. To do this either restart the primary and secondary systems, or re-setup system replication .

3. As `<sid>`adm, restart the `sapstartsrv` service on the secondary system(s):
   a. `sapcontrol -nr <instance_no> -function StopService`
   b. `/usr/sap/<sid>/HDB<instance_no>/exe/sapstartsrv pf=/usr/sap/<sid>/SYS/profile/<sid>_HDB<instance_no>_<host> -D -u <sid>adm`
4. Restart all systems.

Related Information

SAP Note 2447994 - SAP HANA Dynamic Tiering Support for SAP HANA System Replication

10.5 Communication Channels

The network communication channels used by SAP HANA can be categorized into those used for database clients connecting to SAP HANA and those used for internal database communication. SAP recommends using encrypted communication channels where possible.

The following is an overview of the network communication channels used by SAP HANA.

To support the different SAP HANA scenarios and set-ups, SAP HANA has different types of network communication channels:

- Channels used for external access to SAP HANA functionality by end-user clients, administration clients, application servers, and for data provisioning through SQL or HTTP
- Channels used for SAP HANA internal communication within the database, between hosts in multiple-host systems, and between systems in system-replication scenarios

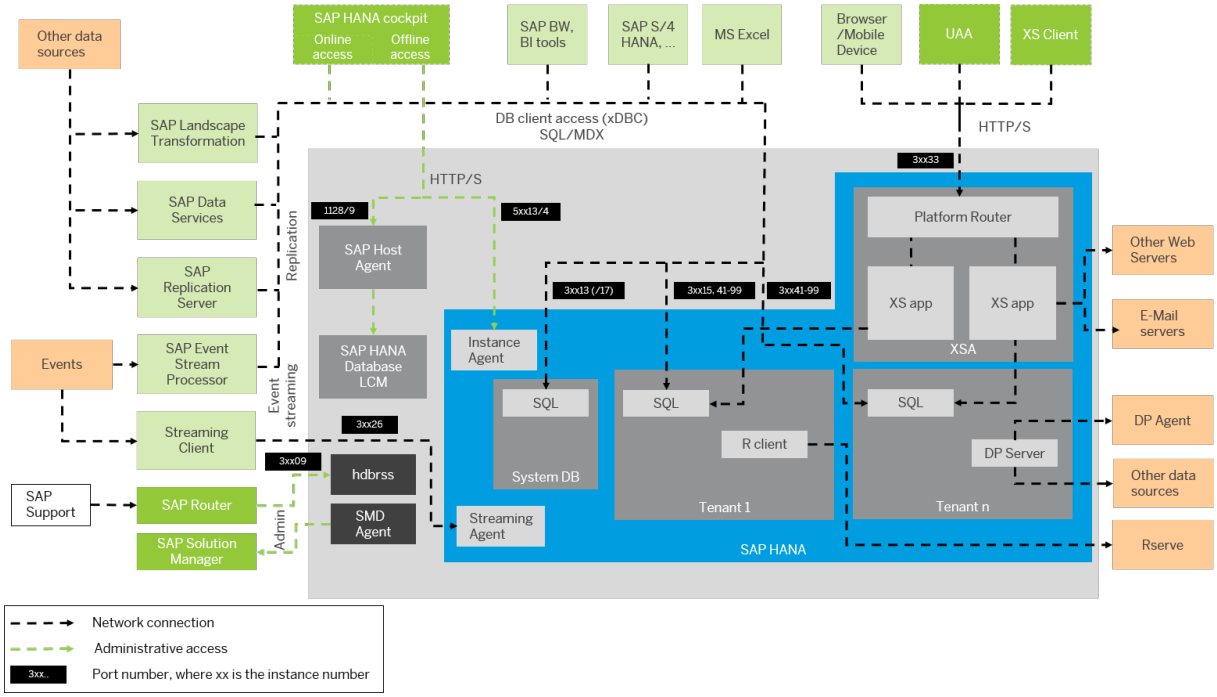
> **i Note**
>
> SAP HANA internal communication has sometimes been unofficially referred to as TREXNet communication. However, the term TREXNet is not valid in the context of SAP HANA.

The connections between SAP HANA and external components and applications come under these categories:

- Connections for administrative purposes
- Connections 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 figure below. Network connections are depicted by dotted arrows. The direction of each arrow indicates which component is the initiator and which component is the listener. 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 number of your SAP HANA instance.

The figure below shows all the network channels used by SAP HANA. For the purposes of illustration, a single-host installation with two tenant databases is depicted. However, the connections shown apply equally to a distributed scenario.



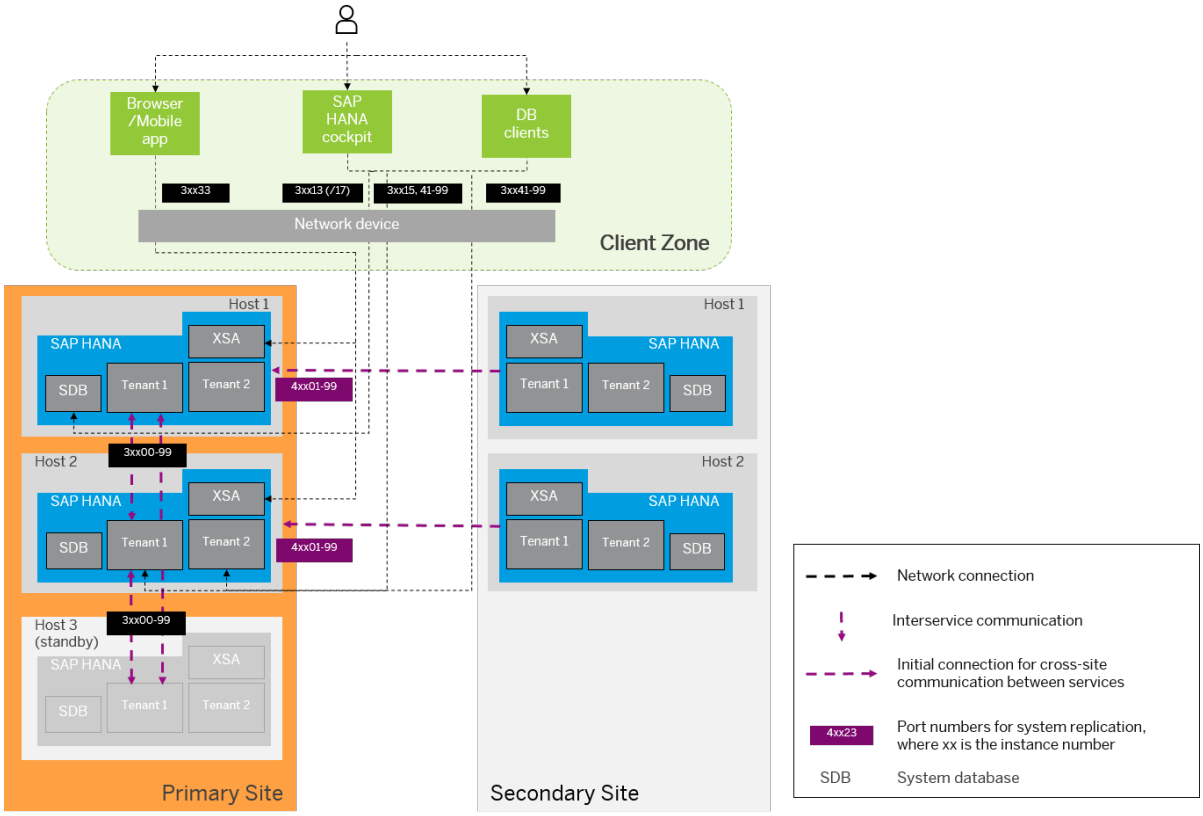
Connections Between SAP HANA and External Components

> **i Note**
>
> Some components depicted in the figure are supported on Intel-based hardware platforms only (for example, SAP HANA Streaming Analytics). Refer to the Product Availability Matrix (PAM).

In addition, the different components of SAP HANA, as well as the hosts in a distributed scenario, communicate with each other over internal SAP HANA connections. These connections are also used in system replication scenarios for communication between a primary site and secondary site(s) to ensure high availability in the event of a data center failure.

The following figure shows an example of a distributed SAP HANA system with two active hosts and an extra standby host, both fully replicated to a secondary site to provide full disaster recovery support.

SAP HANA Internal Connections



Related Information

Securing Data Communication
Network Administration
Product Availability Matrix

10.6 Network Security

To integrate SAP HANA securely into your network environment, several general recommendations apply.

The components of an SAP HANA landscape communicate through different network communication channels. It is recommended security practice to have a well-defined network topology to control and limit network access to SAP HANA to only those communication channels needed for your scenario, and to apply appropriate additional security measures, such as encryption, where necessary. This can be achieved through different means, such as separate network zones and network firewalls, and through the configuration options provided by SAP HANA (for example, encryption). The exact setup depends on your environment, your implementation scenario, and your security requirements and policies.

The detailed network set-up and recommendations are described in network administration section of the *SAP HANA Administration Guide*. This section contains some additional security-relevant information.

> ⚠ Caution
>
> It is strongly recommended that you apply the measures described in this section to protect access to the SAP HANA database's internal communication channels and to mitigate the risk of unauthorized access to these services.

Network Zones

- We recommend that you operate the different components of the SAP HANA platform in separate network zones.
  To prevent unauthorized access to the SAP HANA environment and the SAP HANA database through the network, use network firewall technology to create network zones for the different components and to restrictively filter the traffic between these zones implementing a "minimum required communication" approach. The relevant network zones depend on your specific application scenario and your network infrastructure. For more information about network zones, see the *SAP HANA Administration Guide*.
- We recommend that you operate SAP HANA in a protected data-center environment.Allow only dedicated authorized network traffic from other network zones (for example, user access from the client network zone) to follow these rules:
  - Clients accessing external standard database functionality, for example by SQL, have only access to the database client access port.
  - Clients (for example, browser applications) accessing the SAP HANA environment through the HTTP access feature of SAP HANA Extended Application Services, classic model (XS classic), for example SAP HANA UI Toolkit for Info Access, have only access to the SAP HANA XS ports.
  - Some administrative functions (for example, starting and stopping the SAP HANA instance) have access to the administrative ports.
  - XS classic exposes some administrative applications (for example, administration of Security Assertion Markup Language (SAML) for user authentication). We recommend using URL filtering (for example, reverse proxy) to control the exposure of different applications to different network zones.

Internal Communication

Database internal communication channels are only used for the following:

- Communication within the database
- Communication between hosts in distributed (multiple-host) scenarios
- Communication between multiple sites in system replication (high-availability) scenarios
- Communication between the SAP HANA database and server components, such as extended storage (SAP HANA dynamic tiering)

Note the following network security considerations for single-host, multiple-host, and system replication (high-availability) scenarios.

Single-Host Scenario

In a single-host scenario, access to the network ports for database internal communication from other network hosts is blocked by default. We recommend that you do not change this setting. The internal communication ports are bound to `localhost`.

> **i** 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. Note that this does not apply to dynamic tiering. The dynamic tiering service is bound to all interfaces, although the internal communication between SAP HANA database and dynamic tiering uses the `localhost` interface.

Multiple-Host Scenario

In a distributed scenario (that is, one instance of SAP HANA on multiple hosts), internal network communication takes place between the hosts at one site via internal. Certified SAP HANA hosts contain either dedicated or virtualized network interfaces that are configured as part of a private network using separate IP addresses and ports.

We recommend operating all hosts in a dedicated sub-network.

To prevent unauthorized access to the database via the internal communication channels in distributed systems, we recommend that you prevent access to these network channels and ports from outside the system. There are a number of ways to isolate internal network ports from the client network:

- Using the SAP HANA configuration option to route communication between the hosts of a distributed environment onto a specified network and binding those internal network services exclusively to the network interface (**recommended option**)
  For more information about configuring inter-service communication, see the *SAP HANA Administration Guide*.

  > **i** Note
  >
  > In system replication scenarios, you can use this feature in the presence of a secondary site. 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.
- Using operating system commands (for example, `iptables`), and/or network device configuration
- Using network firewall functions to block access to internal ports in specific network zones

If your setup does not permit isolating internal network communication, consider using encryption to protect the internal communication. For more information, see the section on securing internal communication.

System Replication Scenario

In a system replication scenario, you can protect the channels used in the following ways:

- Configuring SAP HANA to use exclusively a separate network dedicated to system replication for communication between primary and secondary site
- Configuring secure communication using the TLS/SSL protocol for encryption and mutual authentication between sites
- Specifying the IP addresses allowed to connect to system replication ports

Additional Measures for Securing Internal Communication

We recommend that you protect internal communication further by applying additional mechanisms. 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 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 additional network security measures needed will depend on your specific environment. For more information about network administration, see the *SAP HANA Administration Guide*

SAP HANA Extended Application Services, Advanced Model

Security mechanisms are applied to protect the communication paths used by the SAP HANA XS advanced server infrastructure. SAP provides network topology recommendations to restrict access at the network level. For more information, see the section on SAP HANA XS advanced security.

Data Replication Technologies

Additional network configurations may be required depending on the implemented data replication technology. For more information, see the section on security for SAP HANA replication technologies.

Related Information

Secure Internal Communication [page 240]
Network Administration
Network Zones
Host Name Resolution for System Replication
Configuring SAP HANA Inter-Service Communication
Security for SAP HANA Extended Application Services, Advanced Model
Security for SAP HANA Replication Technologies

10.7 Internal Application Encryption Service

The internal encryption service is used internally by applications requiring data encryption.

> **i** Note
>
> In the SAP HANA 1.0 documentation, the internal application encryption service was referred to as the internal data encryption service.

The internal application encryption service is used in the following contexts:

- **Secure internal credential store**
  This service stores credentials required by SAP HANA for outbound connections. It is used, for example, when data is retrieved from remote data sources using SAP HANA smart data access. It is also used during HTTP destination calls from SAP HANA XS classic applications.
  For more information, see the section on the secure internal credential store.
- **SAP HANA secure store**
  SAP HANA provides a set of database procedures to store and manage encrypted values individually per database user. A user who can execute these procedures can create and maintain encrypted values in an internal table that is only available to this user.
  SAP HANA XS advanced applications can use the SAP HANA secure store to store sensitive data such as credentials in a secure manner even if data volume encryption has not been enabled. For more information, see *Maintain Values in the SAP HANA Secure Store* in the *SAP HANA Developer Guide for SAP HANA XS Advanced Model*.
- **Secure stores for SAP HANA XS classic applications**
  Application developers can maintain values can define secure stores using the SAP HANA XS classic `$.security.Store` API.
  For more information, see *Using the Server-Side JavaScript APIs* in the *SAP HANA Developer Guide (For SAP HANA Studio)* and *Class:Store* in the *SAP HANA XS JavaScript API Reference*:
- **Private key store**
  This service stores the private keys of the SAP HANA server required for secure client-server communication, if the relevant personal security environment (PSE) is stored in the database. PSEs stored in the database are called certificate collections.
  For more information, see the section on SSL configuration on the SAP HANA server and certificate management in SAP HANA.

Every consumer of the service has its own system-internal application encryption key. These keys are generated as follows:

- The application key for the internal credential store is generated randomly during the first startup.
- A user-specific key for the SAP HANA secure store is created when a user inserts the first value into the SAP HANA secure store.
- Application keys for XS classic secure stores are created at the same time as the XS classic secure store.
- The application key for the private key store is created when the first private key is set for a certificate collection.

Application encryption keys are encrypted with the application encryption service root key.

SAP HANA generates unique root keys on installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, we recommend that you change the root key of the internal application encryption service to ensure it is not known outside your organization. We recommend that you do this immediately after system installation or handover from your hardware or hosting partner.

> **i** 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.

The system database and all tenant database have their own individual application encryption service root key.

Related Information

Secure Internal Credential Store
Certificate Management in SAP HANA
Maintain Values in the SAP HANA Secure Store
Using the Server-Side JavaScript APIs
Class: Store

11 SQL and System View Reference

Use this chapter to find the system views relevant for system replication.

Related Information

M_SERVICE_REPLICATION System View [page 257]
M_SYSTEM_REPLICATION System View [page 261]
M_SYSTEM_AVAILABILITY System View [page 263]
M_SYSTEM_REPLICATION_MVCC_HISTORY System View [page 265]
M_SYSTEM_REPLICATION_TAKEOVER_HISTORY System View [page 266]
M_SYSTEM_REPLICATION_TIMETRAVEL System View
M_LOG_SEGMENTS System View [page 268]
M_SNAPSHOTS System View [page 271]

11.1 M_SERVICE_REPLICATION System View

Provides information about replicated services.

Structure

Column name	Data type	Description
HOST	VARCHAR(64)	Displays the host name.
PORT	INTEGER(10)	Displays the internal port.
VOLUME_ID	INTEGER	Displays the volume ID.
SITE_ID	INTEGER	Displays the generated site ID.
SITE_NAME	VARCHAR(256)	Displays the logical site name.
SECONDARY_HOST	VARCHAR(64)	Displays the secondary host name.
SECONDARY_PORT	INTEGER	Displays the secondary port.

Column name	Data type	Description
SECONDARY_SITE_ID	INTEGER	Displays the generated ID of the secondary site.
SECONDARY_SITE_NAME	VARCHAR(256)	Displays the secondary logical site name.
SECONDARY_ACTIVE_STATUS	VARCHAR(16)	Displays the secondary active status.
SECONDARY_CONNECT_TIME	TIMESTAMP	Displays the time the connection was established from the secondary site.
SECONDARY_RECONNECT_COUNT	INTEGER	Displays the secondary reconnect count.
SECONDARY_FAILOVER_COUNT	INTEGER	Displays the secondary failover count.
SECONDARY_FULLY_RECOVERABLE	VARCHAR(5)	Displays that the secondary system can be fully recovered with a backup from the primary system. If this value is FALSE, then the backup history is broken. If there is a takeover at that time, then start a new data backup once the takeover is finished.
REPLICATION_MODE	VARCHAR(16)	Displays the replication mode.
REPLICATION_STATUS	VARCHAR(16)	Displays the replication status.
REPLICATION_STATUS_DETAILS	VARCHAR(1.024)	Displays the replication status details.
FULL_SYNC	VARCHAR(16)	Displays the full sync status: • DISABLED: the full sync is not configured • ENABLED: the full sync is configured, but it is not yet active (transactions do not block in this state) • ACTIVE: the full sync mode is configured and active
LAST_LOG_POSITION	BIGINT	Displays the current log position.
LAST_LOG_POSITION_TIME	TIMESTAMP	Displays the current log position timestamp.
LAST_SAVEPOINT_VERSION	INTEGER	Displays the current savepoint version.
LAST_SAVEPOINT_LOG_POSITION	BIGINT(19)	Displays the current savepoint log position.

Column name	Data type	Description
LAST_SAVEPOINT_START_TIME	TIMESTAMP	Displays the current savepoint timestamp.
SHIPPED_LOG_POSITION	BIGINT	Displays the shipped log positon.
SHIPPED_LOG_POSITION_TIME	TIMESTAMP	Displays the shipped log position timestamp.
SHIPPED_LOG_BUFFERS_COUNT	BIGINT	Displays the shipped log buffers count.
SHIPPED_LOG_BUFFERS_SIZE	BIGINT	Displays the shipped log buffers size in bytes.
SHIPPED_LOG_BUFFERS_DURATION	BIGINT	Displays the shipped log buffer duration in microseconds.
REPLAYED_LOG_POSITION	BIGINT	Displays the log end position of the last known replayed log buffer on the secondary site.
REPLAYED_LOG_POSITION_TIME	TIMESTAMP	Displays the timestamp of the last known replayed log buffer on the secondary site.
SHIPPED_SAVEPOINT_VERSION	INTEGER	Displays the shipped savepoint version.
SHIPPED_SAVEPOINT_LOG_POSITION	BIGINT	Displays the shipped savepoint log position.
SHIPPED_SAVEPOINT_START_TIME	TIMESTAMP	Displays the shipped savepoint start time.
SHIPPED_FULL_REPLICA_COUNT	BIGINT	Displays the shipped full replica count.
SHIPPED_FULL_REPLICA_SIZE	BIGINT	Displays the shipped full replica size in bytes.
SHIPPED_FULL_REPLICA_DURATION	BIGINT	Displays the shipped full replica duration in microseconds.
SHIPPED_LAST_FULL_REPLICA_SIZE	BIGINT	Displays the shipped last full replica size in bytes.
SHIPPED_LAST_FULL_REPLICA_START_TIME	TIMESTAMP	Displays the shipped last full replica start time.
SHIPPED_LAST_FULL_REPLICA_END_TIME	TIMESTAMP	Displays the shipped last full replica end time.

Column name	Data type	Description
SHIPPED_DELTA_REPLICA_COUNT	BIGINT	Displays the shipped delta replica count.
SHIPPED_DELTA_REPLICA_SIZE	BIGINT	Displays the shipped delta replica size in bytes.
SHIPPED_DELTA_REPLICA_DURATION	BIGINT	Displays the shipped delta replica duration in microseconds.
SHIPPED_LAST_DELTA_REPLICA_SIZE	BIGINT	Displays the shipped last delta replica size in bytes.
SHIPPED_LAST_DELTA_REPLICA_START_TIME	TIMESTAMP	Displays the shipped last delta replica start time.
SHIPPED_LAST_DELTA_REPLICA_END_TIME	TIMESTAMP	Displays the shipped last delta replica end time.
ASYNC_BUFFER_FULL_COUNT	BIGINT	Displays the number of times the asynchronous replication buffer was full.
BACKLOG_SIZE	BIGINT	Displays the current replication backlog in bytes.
MAX_BACKLOG_SIZE	BIGINT	Displays the maximum replication backlog in bytes.
BACKLOG_TIME	BIGINT	Displays the current replication backlog in microseconds.
MAX_BACKLOG_TIME	BIGINT	Displays the maximum replication backlog in microseconds.
REPLAY_BACKLOG_SIZE	BIGINT	Displays the size, in bytes, of all log buffers that have been shipped to the secondary site but have not yet been replayed on the scondary site.
MAX_REPLAY_BACKLOG_SIZE	BIGINT	Displays the maximumsize, in bytes, of the REPLAY_BACKLOG_SIZE since the system startup.
REPLAY_BACKLOG_TIME	BIGINT	Displays the time difference, in microseconds, between the time of the last shipped log buffer and the last replayed log buffer on the secondary site.

Column name	Data type	Description
MAX_REPLAY_BACKLOG_TIME	BIGINT	Displays the maximum time, in microseconds, of the REPLAY_BACKLOG_TIME since the system startup.

11.2 M_SYSTEM_REPLICATION System View

Monitors system replication information.

Structure

Column name	Data type	Description
SITE_ID	INTEGER	Displays the generated site ID.
SITE_NAME	VARCHAR(256)	Displays the site name.
SECONDARY_SITE_ID	INTEGER	Displays the generated site ID of the secondary site.
SECONDARY_SITE_NAME	VARCHAR(256)	Displays the secondary logical site name.
REPLICATION_MODE	VARCHAR(7)	Displays the configured replication mode: • SYNC: the synchronous replication that acknowledges when a buffer has been written to a disk. • SYNCMEM: the synchronous replication that acknowledges when a buffer has arrived in the memory. • ASYNC: the asynchronous replication. • UNKNOWN: set if the replication mode cannot be determined (for example, if there is a communication error when getting status information from a service).
REPLICATION_STATUS	VARCHAR(12)	Displays the replication status.
OPERATION_MODE	VARCHAR(32)	Displays the operation mode.

Column name	Data type	Description
SECONDARY_READ_ACCESS_STATUS	VARCHAR(16)	Indicates whether the secondary system is read-enabled and if read access is activated: • NOT CONFIGURED: Displays that an operation mode is used that does not allow read access. • STOPPED: Displays that the secondary is running in operation mode logreplay_readaccess, but it is currently stopped. • VERSION MISMATCH: Displays that the secondary system is running in operation mode logreplay_readaccess but read access is internally disabled on the secondary system because it is on a different SAP HANA version than the primary system. • INITIALIZING: Displays that the secondary site is running in operation mode logreplay_readaccess but read access is not yet completely initialized. SQL connections to the secondary system fail in this state. • CONSISTENT: Displays that log replay on the secondary site has reached a global consistent state, but the SQL port is not open. • ACTIVE: Displays that the secondary system is running in operation mode logreplay_readaccess and is initialized for read access. SQL connections are possible in this state.
TIER	INTEGER	Displays the tier on which the source site is located.

Related Information

ALTER SYSTEM {REGISTER | UNREGISTER} SYSTEM REPLICATION SITE Statement (System Management)
ALTER SYSTEM {ENABLE | DISABLE} SYSTEM REPLICATION Statement (System Management)

11.3 M_SYSTEM_AVAILABILITY System View

Monitors the availability of the system.

Structure

Column name	Data type	Description
EVENT_TIME	TIMESTAMP	Displays the time that this event was originally traced.
GUID	NVARCHAR (36)	Displays the event guide.
IS_ORIGIN	VARCHAR (5)	Displays the original entry.
TRACE_HOST	VARCHAR (64)	Displays the host the trace file was read from.
EVENT_NAME	VARCHAR (32)	Displays the event name: • database_add • database_remove • failover_begin • failover_end • host_remove_prepare • host_remove_reorg • host_remove_abort • host_remove • ping • recovery_begin • recovery_end • service_remove • service_remove_abort • service_remove_prepare • service_remove_reorg • service_started • service_starting • service_stopped • service_stopping • service_unknown
EVENT_DETAIL	VARCHAR (256)	Displays any additional information.
ERROR_MESSAGE	VARCHAR (256)	Displays the error message.

Column name	Data type	Description
SYSTEM_ACTIVE	VARCHAR (16)	Displays the system active status: NO, YES, UNKNOWN, STARTING, or STOPPING.
SYSTEM_STATUS	VARCHAR (16)	Displays the system status: • ERROR • IGNORE • INFO • OK • WARNING
HOST	VARCHAR (64)	Displays the host that traced the event.
HOST_ACTIVE	VARCHAR (16)	Displays the host active status.
HOST_STATUS	VARCHAR (16)	Displays the host status.
DATABASE_NAME	VARCHAR (256)	Displays the database name.
DATABASE_ACTIVE	VARCHAR (16)	Displays the database active status.
SERVICE_NAME	VARCHAR (32)	Displays the service name.
PORT	INTEGER	Displays the service port.
VOLUME_ID	INTEGER	Displays the volume ID.
SERVICE_ACTIVE	VARCHAR (16)	Displays the service active status.
HOST_CONFIG_ROLES	VARCHAR (64)	Displays the configured host roles.
HOST_ACTUAL_ROLES	VARCHAR (64)	Displays the actual host roles.
STORAGE_CONFIG_PARTITION	INTEGER	Displays the configured storage partition.
STORAGE_ACTUAL_PARTITION	INTEGER	Displays the actual storage partition.
TARGET_HOST	VARCHAR (64)	Displays the failover target host.
TARGET_HOST_CONFIG_ROLES	VARCHAR (64)	Displays the target host configuration roles.
TARGET_HOST_ACTUAL_ROLES	VARCHAR (64)	Displays the target host actual roles.
TARGET_STORAGE_CONFIG_PARTITION	INTEGER	Displays the target storage configuration partition.

Column name	Data type	Description
TARGET_STORAGE_ACTUAL_PARTITION	INTEGER	Displays the target storage actual partition.
SITE_ID	INTEGER	Displays the system replication site ID.

Related Information

Multiple-Host (Distributed) Systems
The System Database
M_SYSTEM_DATA_STATISTICS System View
M_SYSTEM_INFORMATION_STATEMENTS System View
M_SYSTEM_OVERVIEW System View

11.4 M_SYSTEM_REPLICATION_MVCC_HISTORY System View

Displays the global multi-version concurrency control (MVCC) timestamp history in the secondary site for system replication. The global MVCC timestamp of the secondary site is updated after a chunk of logs from the primary site is replayed on the secondary site.

Structure

Column name	Data type	Description
GLOBAL_MVCC_TIMESTAMP	BIGINT	Displays the global MVCC timestamp
SECONDARY_SITE_TIME	TIMESTAMP	Displays the global MVCC timestamp updated time of the secondary site
PRIMARY_SITE_TIME	TIMESTAMP	Displays the global MVCC updated time of the primary site
SECONDARY_SITE_UPDATE_DURATION	BIGINT	Displays the global MVCC update duration of the secondary site in milliseconds

Related Information

M_SYSTEM_REPLICATION System View [page 261]
M_MVCC_OVERVIEW System View
M_MVCC_SNAPSHOTS System View
M_MVCC_TABLES System View
ALTER SYSTEM {ENABLE | DISABLE} SYSTEM REPLICATION Statement (System Management)

11.5 M_SYSTEM_REPLICATION_TAKEOVER_HISTORY System View

Provides access to a history of HSR takeover executions.

Structure

Column name	Data type	Description
TAKEOVER_START_TIME	TIMESTAMP	Displays the start time of the takeover command. This value matches tenant takeovers that are executed within the same system takeover process.
TAKEOVER_END_TIME	TIMESTAMP	Displays the end time of the takeover command.
EXECUTION_START_TIME	TIMESTAMP	Displays the execution start time for takeover of the transaction domain.
EXECUTION_END_TIME	TIMESTAMP	Displays the execution end time for takeover of the transaction domain.
SITE_ID	INTEGER	Displays the generated ID of the secondary site at takeover time.
SITE _NAME	VARCHAR(64)	Displays the logical name provided by the site administrator at takeover time.
MASTER_NAMESERVER_HOST	VARCHAR(64)	Displays the master nameserver host at takeover time.
VERSION	VARCHAR(32)	Displays the SAP HANA version for the site that is executing the takeover.

Column name	Data type	Description
SOURCE_SITE_ID	INTEGER	Displays the generated ID of the source site at takeover time.
SOURCE_SITE _NAME	VARCHAR(64)	Displays the logical name for the source site provided by the site administrator at takeover time.
SOURCE_MASTER_NAME-SERVER_HOST	VARCHAR(64)	Displays the source site master name-server host at takeover time.
SOURCE_VERSION	VARCHAR(32)	Displays the source site SAP HANA version.
TAKEOVER_TYPE	VARCHAR(10)	Displays how the system went online: • ONLINE: an online takeover • OFFLINE: an offline takeover • TIMETRAVEL: a timetravel takeover
REPLICATION_MODE	VARCHAR(16)	Displays the replication mode at takeover time.
OPERATION_MODE	VARCHAR(32)	Displays the operation mode at takeover time.
REPLICATION_STATUS	VARCHAR(16)	Displays the replication status at takeover time.
LOG_POSITION	BIGINT	Displays the master log position, that has been reached by takeover.
LOG_POSITION_TIME	TIMESTAMP	Displays the time reached by the takeover.
SHIPPED_LOG_POSITION	BIGINT	Displays the highest master log position that has been shipped before executing takeover.
SHIPPED_LOG_POSITION_TIME	TIMESTAMP	Displays the time of the last shipped log buffer before executing takeover.
COMMENTS	NVARCHAR(5000)	Displays a comment for the remote subscription.

Related Information

M_SYSTEM_OVERVIEW System View

M_SYSTEM_REPLICATION System View [page 261]

11.6 M_LOG_SEGMENTS System View

Provides log segment statistics.

Structure

Column name	Data type	Description
HOST	VARCHAR(64)	Displays the host name.
PORT	INTEGER	Displays the internal port.
VOLUME_ID	INTEGER	Displays the persistence volume ID.
PARTITION_ID	BIGINT	Displays the partition ID. Returns the following: • For partitioned tables, the part ID is equal to the sequential number of the partition, starting at 1. • In the case of replicated tables, the part ID is 1 for the original table and subsequent part IDs are assigned to replica tables. • The part ID is 0 for tables that are not partitioned. • A part ID value of -1 indicates that a modification of the table schema is in progress.
SEGMENT_ID	BIGINT	Displays the log segment ID within the partition.
FILE_NAME	VARCHAR(512)	Displays the log segment file name.
FILE_OFFSET	BIGINT	Displays the start position of the log segment in the file.

Column name	Data type	Description
STATE	VARCHAR(16)	Displays the log segment state: **Formatting** The log segment is being formatted and not yet used. **Preallocated** The log segment has been preallocated, but never used. **Writing** The log segment is currently being written. **Closed** The log segment is closed, not backed up and is still required for restart. **Truncated** The log segment is not required for restart, but has not been backed up. **BackedUp** The log segment has been backed up, but is still required for restart. **RetainedFree** The log segment has been backed up and is not required for restart, but is required to resync the system replication sites. **Free** The log segment has been backed up, it is not required for restart and can be reused.
MIN_POSITION	BIGINT	Displays the first position contained in the log segment.
MAX_POSITION	BIGINT	Displays the position behind the last log record in the log segment. This value is for closed log segments only.
HOLE_POSITION	BIGINT	Displays the start position of the log hole before the log segment. This value is equal to MIN_POSITION if there is no hole.

Column name	Data type	Description
USED_SIZE	BIGINT	Displays the used log segment size in bytes.
TOTAL_SIZE	BIGINT	Displays the total log segment size in bytes.
IN_BACKUP	VARCHAR(5)	Indicates whether or not the flag for the log segment is in the backup: TRUE/ FALSE
LAST_COMMIT_TIME	TIMESTAMP	Displays the timestamp of the last commit in the log segment.
ENCRYPTION_KEY_HASH	VARCHAR(64)	Displays the hash of the key used for the log segment encryption.

Additional Information description

This view describes each allocated log segment and shows its current state and log position range that is currently contained in the segment.

This view has a resettable counterpart; you can see the values since the last reset in the M_LOG_SEGMENTS_RESET system view. To reset the view, execute the following statement: `ALTER SYSTEM RESET MONITORING VIEW SYS.M_LOG_SEGMENTS_RESET;`.

Related Information

M_LOG_SEGMENTS_RESET System View
M_LOG_BUFFERS System View
M_LOG_PARTITIONS System View
M_VOLUME_IO_TOTAL_STATISTICS System View

11.7 M_SNAPSHOTS System View

Provides information about existing snapshots.

Structure

Column name	Data type	Description
HOST	VARCHAR(64)	Displays the host name.
PORT	INTEGER	Displays the internal port.
VOLUME_ID	INTEGER	Displays the persistence volume ID.
ID	BIGINT	Displays the snapshot ID.
TIMESTAMP	TIMESTAMP	Displays the creation time.
PURPOSE	VARCHAR(24)	Displays why the snapshot was executed.
FOR_BACKUP	VARCHAR(5)	Displays if the snapshot was created for backup: TRUE/FALSE.
ANCHOR	BIGINT	Displays the anchor.
REDO_LOG_POSITION	BIGINT	Displays the redo log position corresponding to the snapshot.
LAST_COMMIT_TIME	TIMESTAMP	Displays the timestamp of the last commit of the snapshot.

Related Information

Monitoring and Managing Tenant Databases
M_DATABASES System View

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 a 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

THE BEST RUN SAP