PUBLIC
2017-10-26

Redis Service

THE BEST RUN SAP

Content

1 Overview

Redis is a popular in-memory data structure store used as a database, cache, and message broker. It supports data structures such as strings, hashes, lists, sets, and so on.

The benefits of using Redis-as-a-Service are:

- Ensures uninterrupted business transactions by providing high availability.
- Multi-utility tool that supports intermittent data storage. It also supports caching and message-queues.

  > ✤ Example
  >
  > You can use it to store Web application sessions.

- Restore the data of a service instance on to another instance within the same CF space. If your service instance is deleted by any chance, you can also restore the data from the deleted instance on to another instance. See Restore Data [page 6].

Third Party Notice

1.1 High Availability

Redis Sentinel provides high availability. Redis contains an in-built replication mechanism based on the master-slave model.

If the master server stops functioning because of unexpected reasons, Sentinel starts a failover process and promotes a slave as a master. The other slaves are reconfigured to use the new master. Sentinel is also a source of authority for clients to discover the master (service provider). Clients connect to Sentinel to get the address of the current Redis master responsible for providing service. If a failover occurs, Sentinel reports the new address of the Redis master.

1.2 Access the Service

To use the Redis service, you need to create an instance of the service and bind the service with your application.

Prerequisites

- Install Cloud Foundry Command Line Interface on your system. For more information, see http://docs.cloudfoundry.org/devguide/installcf/install-go-cli.html .
- You have deployed the application that requires Redis to SAP Cloud Platform. For more information, see http://docs.cloudfoundry.org/devguide/deploy-apps/deploy-app.html .

Procedure

1. Open the Cloud Foundry console client.
2. Check the Service Marketplace to see if the Redis service is listed using the command: `cf marketplace`.
3. Create an instance of the service using the command: `cf create-service redis <service plan name> <instance name>`.
4. Bind the service instance to the application using the command: `cf bind-service <application name> <instance name>`.
5. Restage the application using the command: `cf restage <application name>`.
6. Check if the instance is successfully bound to the application using the command: `cf env <application name>`

> **i Note**
>
> If the instance is successfully bound, you will find the Redis details along with its credentials in the VCAP_SERVICES environment variable.

```
 A sample VCAP_SERVICES for a multinode Redis plan appears as below:
"VCAP_SERVICES" : {
  "credentials": {
    "password": "97fde896c00c98a7cac542c5bafc6035",
    "redis_nodes": [
      {
        "hostname": "10.11.34.96",
        "port": "6379"
      },
      {
        "hostname": "10.11.34.97",
        "port": "6379"
      },
      {
        "hostname": "10.11.34.98",
        "port": "6379"
      }
    ],
```

```
    "sentinel_nodes": [
      {
        "hostname": "10.11.34.96",
        "port": "26379"
      },
      {
        "hostname": "10.11.34.97",
        "port": "26379"
      },
      {
        "hostname": "10.11.34.98",
        "port": "26379"
      }
    ]
  },
  "label": "redis",
  "name": "redisTest",
  "plan": "<service plan name>",
  "provider": null,
  "syslog_drain_url": null,
  "tags": [
    "redis",
    "keyvalue"
  ],
  "volume_mounts": []
}
```

1.3 Connect to the Multinode Service Instance

After binding the application, you need to configure the application to connect to the Redis cluster. The configuration enables parsing of the credentials available in the VCAP_SERVICES environment variable and establishes connection with the cluster.

Procedure

1. Parse the VCAP_SERVICES variable.

```
Example:
REDIS_PASSWORD
=arr.getJSONObject(0).getJSONObject("credentials").getString("password");
JSONArray rootobj =
arr.getJSONObject(0).getJSONObject("credentials").getJSONArray("sentinel_nodes
");
SENTINEL_HOST1 = rootobj.getJSONObject(0).getString("hostname");
SENTINEL_PORT1 = rootobj.getJSONObject(0).getString("port");
SENTINEL_HOST2 = rootobj.getJSONObject(1).getString("hostname");
SENTINEL_PORT2 = rootobj.getJSONObject(1).getString("port");
SENTINEL_HOST3 = rootobj.getJSONObject(2).getString("hostname");
SENTINEL_PORT3 = rootobj.getJSONObject(2).getString("port");
```

2. To establish a connection with the nodes of the Redis cluster, use a Java- based client library.

```
Example: The example uses a Jedis client.
Set<String> sentinels = new HashSet<String>();
String host1 = Configuration.getInstance().SENTINEL_HOST1 + ":" +
Configuration.getInstance().SENTINEL_PORT1;
```

```
String host2 = Configuration.getInstance().SENTINEL_HOST2 + ":" +
Configuration.getInstance().SENTINEL_PORT2;
String host3 = Configuration.getInstance().SENTINEL_HOST3 + ":" +
Configuration.getInstance().SENTINEL_PORT3;
sentinels.add(host1);
sentinels.add(host2);
sentinels.add(host3);
pool = new JedisSentinelPool("mymaster", sentinels);
Jedis jedis = pool.getResource();
jedis.auth(Configuration.getInstance().REDIS_PASSWORD);
```

1.4 Restore Data

Resotring data from one instance to another instance

Procedure

1. Log on to the Cloud Foundry instance using the command `cf login`.
2. Execute `cf list-backup <service_instance_name>` command to display a list of backups that are specific to the service instance within a space.
3. Execute `cf start-restore <service_instance_name> <backup_id>` command to restore data from the service instance from the specified instance name and backup ID.

> **i** Note
>
> To know more about the CF CLI commands, see CLI Plugin .

Restoring data from a deleted instance

Procedure

1. Log on to the Cloud Foundry instance using the command `cf login`.
2. Execute `cf instance-events --delete` command to list all deleted service instance events in the space.
3. Execute `cf list-backup<service_instance_name> --deleted` to display the list of all backups for a deleted service instance.
4. Execute `cf start-restore <service_instance_name> <backup_id>` command to restore data from the service instance from the specified instance name and backup ID.

> **i** Note
>
> Instance will be unavailable for connections during the restore operation; existing connections will be lost.

2 Schedule Maintenance for Redis Service Instance

SAP performs periodic maintenance updates to the Redis instance's operating system or processes in a predefined schedule by default. Updates to the operating system are often critical security patches and should be applied as soon as possible.

Context

Maintenance updates can potentially make the instance unavailable for a short period of time. To address this issue, SAP allows you to set a maintenance schedule for the instances from the Cloud Cockpit and perform updates. To schedule maintenance for an instance, follow the steps below:

Procedure

1. On the SAP Cloud Platform cockpit, choose *Login* and enter your credentials.
2. To navigate to the *Regions* overview page, choose *Home*.
3. Choose your *Global Account name* > *Subaccount* > *Spaces* > *Space Name* > *Service Instances* > *Service Instance name*
4. To set a custom maintenance schedule, click *Maintenance Schedules*.
5. Enter a maintenance schedule by selecting a day and time.
6. Choose *Save*.

> **i** Note
>
> SAP does not allow users to postpone maintenance updates indefinitely to avoid missing critical updates.

3 Data Protection and Privacy

Governments place legal requirements on industry to protect data and privacy. We provide features and functions to help you meet these requirements.

> **i Note**
>
> SAP does not provide legal advice in any form. SAP software supports data protection compliance by providing security features and data protection-relevant functions, such as blocking and deletion of personal data. In many cases, compliance with applicable data protection and privacy laws is not covered by a product feature. Furthermore, this information should not be taken as advice or a recommendation regarding additional features that would be required in specific IT environments. Decisions related to data protection must be made on a case-by-case basis, taking into consideration the given system landscape and the applicable legal requirements. Definitions and other terms used in this documentation are not taken from a specific legal source.

The following sections provide information about the Redis service. For the central data protection and privacy statement for SAP Cloud Platform, see Data Protection and Privacy

User Consent

We assume that software operators, such as SAP customers, collect and store the consent of data subjects, before collecting personal data from data subjects. A data privacy specialist can later determine whether data subjects have granted, withdrawn, or denied consent.

The MongoDB service does not provide any support for collecting and storing the consent of data subjects for applications built on SAP Cloud Platform. It is the responsibility of your applications to provide such support.

Read-Access Logging and Change Log

Read-access logging (RAL) is used to monitor and log read access to sensitive data. Data may be categorized as sensitive by law, by external company policy, or by internal company policy. Read-access logging enables you to answer questions about who accessed certain data within a specified time frame.

For auditing purposes or for legal requirements, changes made to data should be logged, enabling the monitoring of what changes were made and when. Auditing provides you with visibility on who did what in the Redis and when. The audit log entries are forwarded to the Audit Log service and stored there for a defined period of time.

For more information on retrieving your audit logs, see Audit Log Retrieval API Usage for the Cloud Foundry Environment.

For more information on retaining audit logs, see Audit Log Retention for the Cloud Foundry Environment.

For information on Audit log viewer, see Audit Log Viewer for the Cloud Foundry Environment

Erasure

When handling personal data, consider the legislation in the different countries where your organization operates. After the data has passed the end of purpose, regulations may require you to delete the data. However, additional regulations may require you to keep the data longer. During this period you must block access to the data by unauthorized persons until the end of the retention period, when the data is finally deleted.

You can delete your Redis instance, and therefore, all the data stored in your databases. To do so, navigate to your sub account using the procedure Navigate to Global Accounts, Subaccounts, Orgs, and Spaces in the Cockpit and delete the service from the *Overview* page.

After you delete your data, we may have this data in our backup system for the length of our backup retention period. SAP keeps backups of your databases for a retention period of 14 days. Backups are deleted afterwards.

Glossary

Term	Definition
Consent	The action of the data subject confirming that the usage of his or her personal data shall be allowed for a given purpose. A consent functionality allows the storage of a consent record in relation to a specific purpose and shows if a data subject has granted, withdrawn, or denied consent.
Deletion	Deletion of **personal data** so that the data is no longer available.
Personal data	Any information relating to an identified or identifiable natural person ("data subject"). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person
Retention period	The period of time between the end of the last business activity involving a specific object (for example, a business partner) and the deletion of the corresponding data, subject to applicable laws. The retention period is a combination of the residence period and the blocking period.

Term	Definition
Sensitive personal data	A category of personal data that usually includes the following type of information: • Special categories of personal data, such as data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, genetic data, biometric data, data concerning health or sex life or sexual orientation, or personal data concerning bank and credit accounts. • Personal data subject to professional secrecy • Personal data relating to criminal or administrative offenses • Personal data concerning insurances and bank or credit card accounts

4 Data Encryption Strategy

Your crucial information stored in the database is maintained in a highly secure manner as we at SAP use the encryption capabilities provided by the underlying IaaS providers AWS, Azure, GCP and SAP DC. Encryption details for each of the IaaS provides are described below:

Data Center	Data stored on persistent disk	Backup data	Reference
Amazon Web Service	Data is stored on encrypted Elastic Block Store (EBS) volumes. EBS uses Amazon Key Management Service (AWS KMS) customer master keys (CMKs) to encrypt volumes/disks. AWS manages the key per account and this key is used for all encryptions in that account.	Encrypted EBS volume snapshots stored on AWS S3. EBS uses Amazon Key Management Service (AWS KMS) customer master keys (CMKs) to encrypt snapshots.	EBS Encryption
Microsoft Azure	Data is stored on encrypted Managed Disks. Azure SSE (Storage Service Encryption) provides encryption-at-rest for managed disks.	Encrypted managed disk snapshots stored on Azure Zone Redundant Storage (ZRS).	Azure Manage Disks
GCP	Data is stored on encrypted Persistent Disks. Persistent disk is encrypted with system-defined keys (managed by GCP).	Encrypted persistent disk snapshots are taken.	Create Snapshots
SAP DC	Data is stored on persistent volume/disks.	Encrypted backups are uploaded to SAP Swift storage. SAP controls the keys for SAP Cloud Platform in SAP data centers.	

5 Tutorial

This 15-minute tutorial summarizes the basic steps for caching data using the Redis service.

Prerequisites

- You have downloaded Cloud Foundry CLI available at Github .
- You have installed a JDK 1.8 available on the Oracle Download page .
- You have downloaded Maven 3.0 available on the Apache Maven Project Download page .
- You have downloaded Git available on the Git Download page .

Procedure

1. Clone the project from github.

```
git clone https://github.com/SAP/hcp-cloud-foundry-tutorials.git
```

2. Log on to the Cloud Foundry and navigate to the org and space you have access to.

```
cf api api.cf.<host information>
cf login
```

> **i Note**
>
> For more information about the host specific to your region, see Regions and Hosts.

3. Create a service instance.

```
cf create-service redis v3.0-dev redis-service
```

> **→ Tip**
>
> The version of the Redis service changes on a regular basis. To check the version, use the `cf marketplace` command.

4. Build the sample project. For more information, see the section **More Details: Build Your Application** in this tutorial.

```
cd hcp-cloud-foundry-tutorials/hcp-cf-redis-tutorial
mvn package
```

5. Modify in the `manifest.yml`, provide unique host value, and then push the project.

```
cf push
```

The application push might fail with the following error: `Server error, status code: 400, error code: 210003, message: The host is taken:`

In such case, check the uniqueness of the host value in the manifest.yml .

6. Run the project in your Web browser using the URL that was created after you pushed the application.

Build Your Application

The procedure explains how to build your application.

1. Create the java maven project:

```
mvn archetype:generate -DgroupId=hcp-cf-redis-tutorial -DartifactId=hcp-cf-redis-tutorial -DinteractiveMode=false
```

2. Once the project has been created, edit your `pom.xml` file to put all the relevant dependencies. It should look similar to this pom.xml sample file.
3. Create a `manifest.yml` file, which contains the cf configuration details and looks similar to this manifest.xml sample file. To prevent an application push from failing, make sure that your host value is unique.
4. Setup a Spring Boot starter class, which starts the application and should look similar to this HCPWebApplication.java sample file. For more information about Spring Boot, see the Spring Boot page .
5. Also setup an important class is the controller class, which is responsible for HTTP request mapping. The class should look like this RootController.java sample file.
6. Check the important aspect of controller class as explained:
   The codeblock shows the class structure that includes an important part of the database connection from the bound service. The annotation `@Controller` tells you that it is a controller class. `@RequestMapping("/")` tells you that this class is called when the root URL of the app is hit. `onRootAccess()` is the method that handles the `GET` request, which is pointed to by the `@RequestMapping(method = RequestMethod.GET)` annotation. We have a `redisConnectionFactory` field that is `@Autowired`. Therefore, a Redis connection can be created when parsing the service instance bound to this application. The tutorial uses the `redisConnectionFactory` to access Redis directly.

```
package com.sap.hcp.cf.tutorials.redis;
import java.util.ArrayList;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.redis.connection.RedisConnection;
import org.springframework.data.redis.connection.RedisConnectionFactory;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import com.sap.hcp.cf.tutorials.redis.model.RedisObject;
import com.sap.hcp.cf.tutorials.redis.model.Result;
@Controller
@RequestMapping("/")
public class RootController {
    private static final Logger log =
LoggerFactory.getLogger(RootController.class);
```

```
    @Autowired
    RedisConnectionFactory redisConnectionFactory;
    @RequestMapping(method = RequestMethod.GET)
    public @ResponseBody Result onRootAccess() {
        RedisConnection redisConnection =
redisConnectionFactory.getConnection();
        log.info("Setting key/value pair 'hello'/'world'");
        redisConnection.set("hello".getBytes(), "world".getBytes());
        log.info("Retrieving value for key 'hello': ");
        final String value = new
String(redisConnection.get("hello".getBytes()));
        Result result = new Result();
        result.setStatus("Succesfully connected to Redis and retrieved the
key/value pair that was inserted");
        RedisObject redisObject = new RedisObject();
        redisObject.setKey("hello");
        redisObject.setValue(value);
        final ArrayList<RedisObject> redisObjects = new
ArrayList<RedisObject>();
        redisObjects.add(redisObject);
        result.setRedisObjects(redisObjects);
        return result;
    }
}
```

Now this application is ready to be deployed. To push this newly created application to Cloud Foundry and run it, repeat the steps 2-5 of this tutorial.

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.

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.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.sap.com/contactsap

THE BEST RUN SAP