



Distributed Cache Service

Data Migration Guide

Issue 01

Date 2020-03-17

Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://www.huawei.com>

Email: support@huawei.com

Contents

1 Overview.....	1
2 Migration Process.....	3
3 Migration Tools and Schemes.....	9
4 Migrating Data from Self-Hosted Redis to DCS.....	14
4.1 Online Migration of Self-Hosted Redis.....	14
4.2 Backup Migration of Self-Hosted Redis.....	17
4.3 Self-Hosted Redis Migration with redis-cli (AOF).....	20
4.4 Self-Hosted Redis Migration with redis-cli (RDB).....	22
4.5 Self-Hosted Redis Migration with redis-port (RDB).....	24
4.6 Self-Hosted Codis Migration with redis-cli and redis-port.....	27
4.7 Self-Hosted Redis Cluster Migration with redis-shake.....	29
5 Migrating Data Between DCS Instances.....	34
5.1 Online Migration Between DCS Redis Instances.....	34
5.2 Backup Migration Between Regions or Redis Versions.....	37
6 Migrating Redis Data from Another Cloud to DCS.....	41
6.1 Online Migration from Another Cloud.....	41
6.2 Backup Migration from Another Cloud.....	44
6.3 Online Migration with Rump.....	47
6.4 Codis Migration from Another Cloud with redis-cli and redis-port.....	49
6.5 Redis Cluster Migration from Another Cloud with redis-shake.....	51
7 Migrating Data from DCS to Self-Hosted Redis.....	54
8 FAQs.....	55
8.1 How Do I Migrate Memcached Data?.....	55
8.2 What Are Redis Version Requirements for Data Migration?.....	55
8.3 What Should I Consider When Transferring or Operating Data Between Different OSs?.....	55
8.4 Can I Migrate Data from a Multi-DB Source Redis Instance to a Cluster DCS Redis Instance?.....	56
8.5 How Can I Migrate Partial Data?.....	56
8.6 What Are the Constraints and Precautions for Migrating Redis Data to a Cluster Instance?.....	56
8.7 What Should I Consider for Online Migration?.....	57
8.8 Can I Perform Online Migration Without Any Service Interruption?.....	57

8.9 Can I Migrate Data Between DCS Memcached and Redis Instances?..... 58

1 Overview

This guide provides suggestions and instructions on how to migrate Redis data. For details on how to migrate Memcached data, see [8.1 How Do I Migrate Memcached Data?](#)

Due to variations of Redis application environments and scenarios, migration solutions must be detailed to address actual requirements. The time required for data migration is related to the data volume, the location of source Redis data, and the network bandwidth. Record and evaluate the duration during the rehearsal phase.

DCS Redis 3.0, 4.0, 5.0, and 6.0 (professional edition) instances can be used for data migration. Analyze Redis commands (reference: [Redis Command Compatibility](#)) used by your service systems and verify the commands one by one during the rehearsal phase. For technical support, contact Huawei customer service.

NOTICE

- Data migration is an important and stringent task requiring high accuracy and timeliness. It varies depending on specific services and operation environments.
- Cases provided in this document are for reference only. Consider your service scenarios and requirements during actual migration.
- Some commands in this document contain instance passwords, which will be recorded in the operating system (OS). Ensure that the passwords are not disclosed and clear operation records in a timely manner.

Table 1-1 DCS data migration modes

Migration Mode	Source	Target: DCS		
		Single-Node and Master/Standby	Proxy Cluster	Redis Cluster
Importing backup files	OBS bucket: AOF files	√	√	×

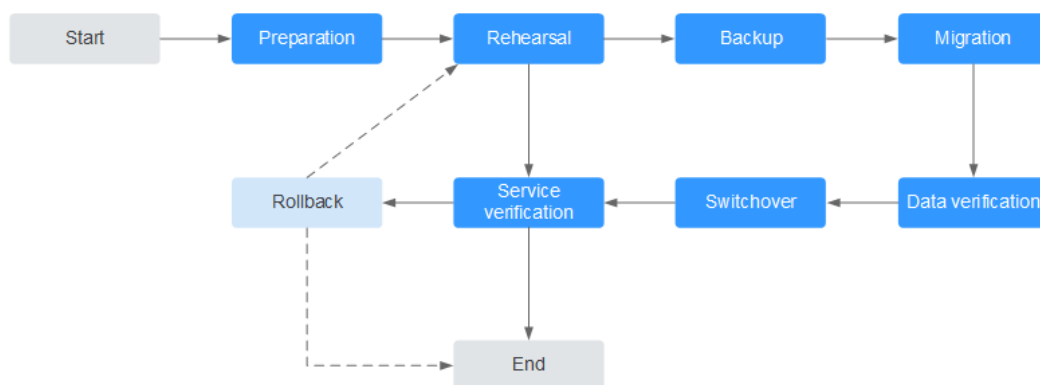
	OBS bucket: RDB files	√	√	√
Migrating data online	DCS for Redis: single-node or master/standby	√	√	√
	DCS for Redis: Proxy Cluster	√	√	√
	DCS for Redis: Redis Cluster	√	√	√
	Self-hosted single-node or master/standby Redis	√	√	√
	Self-hosted proxy-based cluster Redis	√	√	√
	Self-hosted Redis Cluster	√	√	√
	Other Redis: single-node or master/standby	×	×	×
	Other Redis: proxy-based cluster	×	×	×
	Other Redis: Redis Cluster	×	×	×

 **NOTE**

- **DCS for Redis** refers to Redis instances provided by DCS
- **Self-hosted Redis** refers to self-hosted Redis on the cloud, from other cloud vendors, or in on-premises data centers.
- **Other cloud Redis** refers to Redis services provided by other cloud vendors.
- √: Supported. ×: Not supported.

2 Migration Process

Figure 2-1 Migration flowchart



Evaluation

Collect the following information about the cached data to be migrated (based on [Information to be collected for the migration](#)):

- Number of instances
- Number of databases (DBs) configured for each instance
- Number of keys in each DB
- DBs used for your services
- Space occupied by each instance
- Redis version
- Redis instance configurations (single-node, master/standby, or cluster)
- Mapping relationships between your services and instances

Plan the following information about DCS instances based on the collected information:

- Number of instances to be applied for
- Specifications and type (single-node, master/standby, or cluster) of each instance
- Virtual Private Clouds (VPCs), subnets, and security groups, to which the instances and services belong

 NOTE

redis-cli -h \${redis_address} -p \${port}

- Run the following command to query the data distribution and obtain the IDs of DBs with data and the number of keys in each DB:

info keyspace

Query and record the number of keys in each DB for subsequent migration verification.

- Run the following command to query the space occupied by the instance data. Check whether the available disk space of Elastic Cloud Servers (ECSs) is sufficient for transition, and whether the instance specifications and remaining available memory are sufficient.

info memory

The occupied space can be obtained from the value of **used_memory_human**.

Preparation

After completing the evaluation, prepare the following items:

1. Mobile storage devices

These devices are used to copy and transfer data in case of network disconnection (in scenarios with data centers of enterprises).

2. Network resources

Apply for elastic IP addresses (EIPs) for ECSs, create VPCs and subnets based on the service planning, and set security group rules.

Apply for temporary high-specification bandwidth for the EIPs to improve data transmission efficiency.

3. Server resources

Buy ECSs to bear Redis clients. The ECSs are used to export or import cached data.

Recommended ECS specifications are 8 vCPUs | 16 GB or higher.

4. DCS instances

Buy DCS instances based on the migration planning. If the number of instances exceeds the default quota, submit a service ticket or contact customer service.

5. Related tools

Install the SSH tool, FTP tool, and Redis migration tools.

6. Information to be collected

Collect the contact information of people involved in the migration, server addresses, login credentials, cache instance information, and DB information.

7. Overall migration plan

Formulate the overall migration plan, including the personnel arrangement, rehearsal, migration, verification, service switchover, and rollback solutions.

Break down each solution into executable operations and set milestones to mark the end of tasks.

Rehearsal

The rehearsal phase aims to:

1. Verify the feasibility of the migration tools and migration process.
2. Discover problems that may occur during migration and make effective improvements.
3. Evaluate the time required for migration.
4. Optimize the migration steps and verify the feasibility of concurrent implementation of some tasks to improve migration efficiency.

Backup

Before migration, back up related data, including but not limited to cached data and Redis configuration files, in case of emergency.

Migration

After conducting one or two rounds of migration rehearsal and solving problems found in the rehearsal, start data migration.

Break down the migration process into executable steps with specific start and end confirmation actions.

Data Verification

Check the following items:

- The key distribution of each DB is consistent with the original or expected distribution.
- Main keys.
- Expiration time of keys.
- Whether instances can be normally backed up and restored.

Service Switchover

1. After the data migration and verification, use the new instances for your services.
2. If DB IDs are changed, modify the ID configurations for your services.
3. If your services are migrated from data centers or cloud platforms provided by other vendors to HUAWEI CLOUD as a whole, services and cached data can be migrated concurrently.

Service Verification

After the service switchover:

1. Verify the connectivity between your service applications and DCS instances.
2. Verify whether cached data can be normally added, deleted, modified, and queried.
3. If possible, perform pressure tests to ensure that the performance satisfies the peak service pressure.

Rollback

If your services are unavailable after the data migration because unexpected problems occur and cannot be solved in the short term, roll back your services.

Since source Redis data still exists, you only need to roll back your services and use the source Redis instances again.

After the rollback, you can continue to restart from the rehearsal or even preparation phase to solve the problems.

Information to be collected for the migration

The following table lists the information to be collected in the evaluation and preparation phases.

Table 2-1 Information to be collected for the migration

Migration Source	Item	Description
Source Redis (List the information about all instances to be migrated.)	Source Redis IP address	-
	Redis instance password (if any)	-
	Total data volume	Obtained from the value of used_memory_human by running the info memory command. Used to evaluate whether the migration solution, DCS instance specifications, and available disk space of ECSs meet requirements, and to estimate the time required for migration (service interruption duration).
	IDs of DBs with data	Obtained by running the info keyspaces command. Used to check whether the migration involves multiple DBs and non-AOF files. Some open-source tools can export and import data of only one DB at a time. For DCS instances, the single-node and master/standby types provide 256 DBs (DB 0 to DB 255), and the cluster type provides only one DB.
	Number of keys in each DB	Used to verify the data integrity after migration.

Migration Source	Item	Description
	Data type	The Cloud Data Migration (CDM) service supports two data formats: hash and string. If the source data contains data in other formats such as list and set, use a third-party migration tool.
HUAWEI CLOUD ECS If a large number of instances are to be migrated, prepare multiple ECSs for concurrent migration.	EIP	Select ECSs that can communicate with DCS instances for data import to ensure network stability. Configure high-specification bandwidth to improve data transmission efficiency.
	Login credentials (username and password)	-
	CPU and memory	Some migration tools support concurrent import through multiple threads. High-specification ECSs help improve import efficiency.
	Available disk space	Sufficient available disk space needs to be reserved on the ECSs to store compressed files and decompressed cached data files. Note: To improve data transmission efficiency, compress large-size data files before transmitting them to ECSs.
DCS instances (Select appropriate instance specifications and quantities based on the number of source Redis instances and data volume.)	Instance connection address	-
	Instance connection port	-
	Instance password	-
	Instance type	-
	Instance specifications and available memory	-
Network configurations	VPC	Plan VPCs in advance to ensure that your service applications and DCS instances are in same VPCs.
	Subnet	-

Migration Source	Item	Description
	Security group or whitelist	DCS helps you control access to your DCS instances by using security groups or whitelists. For details, see How Do I Configure a Security Group? or Managing IP Address Whitelist.
...	...	<i>Other configurations.</i>

3 Migration Tools and Schemes

Migration Tools

Table 3-1 Comparing Redis migration tools

Tool/ Command/ Service	Feature	Description
DCS console	Supports online migration (in full or incrementally) and backup migration (by importing backup files) with intuitive operations.	<ul style="list-style-type: none">• Backup migration is suitable when the source and target Redis instances are not connected, and the source Redis instance does not support the SYNC and PSYNC commands. To migrate data, import your backup files to OBS, and DCS will read data from OBS and migrate the data to the target DCS Redis instance.• Online migration is suitable when the source Redis instance supports the SYNC and PSYNC commands. Data in the source Redis instance can be migrated in full or incrementally to the target instance.

Tool/ Command/ Service	Feature	Description
redis-cli	<ul style="list-style-type: none"> The Redis command line interface (CLI), which can be used to export data as an RDB file or import the AOF file (that is, all DBs) of an instance. An AOF file is large file containing a full set of data change commands. 	-
redis-port	<ul style="list-style-type: none"> A third-party open-source tool, which processes RDB files in master/slave synchronization mode. An RDB file is a point-in-time memory snapshot, featuring a small size and short time for creation and restoration. Supports online data migration (in synchronous mode). 	Some cloud vendors do not support RDB file export by using redis-port. In this case, download backup files (in RDB format) by using the backup and restoration function provided by the management console of the relevant vendor, and import the files to HUAWEI CLOUD DCS.
Rump	Supports online migration between DBs of an instance or between DBs of different instances.	Rump does not support incremental migration. Stop services before migrating data. Otherwise, keys might be lost. For details, see 6.3 Online Migration with Rump .
redis-shake	An open-source tool that supports both online and offline migration.	redis-shake is suitable for migrating Redis Cluster data.
Self-developed migration script	Flexible and can be adjusted as required.	-

Migration Schemes

 NOTE

Self-hosted Redis refers to self-hosted Redis on HUAWEI CLOUD, in another cloud, or in on-premises data centers.

Table 3-2 Migration schemes

Scenario	Tool	Use Case	Description
From self-hosted Redis to DCS	DCS console	<ul style="list-style-type: none"> If the network between your self-hosted Redis instance and the DCS Redis instance is connected, follow to the instructions in 4.1 Online Migration of Self-Hosted Redis. If the network between your self-hosted Redis instance and the DCS Redis instance is not connected, follow to the instructions in 4.2 Backup Migration of Self-Hosted Redis. 	-
	redis-cli	4.3 Self-Hosted Redis Migration with redis-cli (AOF)	-
		4.4 Self-Hosted Redis Migration with redis-cli (RDB)	-
	redis-port	4.5 Self-Hosted Redis Migration with redis-port (RDB)	-
	redis-cli and redis-port	4.6 Self-Hosted Codis Migration with redis-cli and redis-port	-
	redis-shake	4.7 Self-Hosted Redis Cluster Migration with redis-shake	-

Scenario	Tool	Use Case	Description
Between DCS instances	DCS console	<p>Migrate data from an earlier-version DCS Redis instance to a later-version DCS Redis instance, for example, from a DCS Redis 3.0 instance to a DCS Redis 4.0 or 5.0 instance.</p> <ul style="list-style-type: none"> • If the network between the source and target DCS Redis instances is connected, follow to the instructions in 5.1 Online Migration Between DCS Redis Instances. • If the network between the source and target DCS Redis instances is not connected, follow to the instructions in 5.2 Backup Migration Between Regions or Redis Versions. 	<p>Attempts to migrate data from a later-version Redis instance to an earlier-version Redis instance are not recommended because they will fail due to data compatibility issues between different Redis versions.</p>
		<p>Migrate Redis data between regions, for example, from CN South-Guangzhou to CN North-Beijing1. For details, see 5.2 Backup Migration Between Regions or Redis Versions.</p>	<p>The SYNC and PSYNC commands are disabled by default for DCS Redis instances. These commands are enabled for online migration within a region, and remain disabled for online migration between regions. Therefore, you can only use backup migration when migrating DCS Redis instance data between regions.</p>

Scenario	Tool	Use Case	Description
		<p>Migrate Redis data from one HUAWEI CLOUD account to another.</p> <ul style="list-style-type: none"> For details, see 5.2 Backup Migration Between Regions or Redis Versions. If the DCS Redis instances of the two accounts are connected, you can also follow the instructions in 5.1 Online Migration Between DCS Redis Instances. 	-
From another cloud to DCS	DCS console	<ul style="list-style-type: none"> If the SYNC and PSYNC commands are not disabled for the Redis service provided by another cloud, follow the instructions in 6.1 Online Migration from Another Cloud. If the SYNC and PSYNC commands are disabled for the Redis service provided by another cloud, follow the instructions in 6.2 Backup Migration from Another Cloud. 	If online migration is required, contact the O&M personnel of another cloud to enable the SYNC and PSYNC commands.
	Rump	6.3 Online Migration with Rump	-
	redis-cli and redis-port	6.4 Codis Migration from Another Cloud with redis-cli and redis-port	-
	redis-shake	6.5 Redis Cluster Migration from Another Cloud with redis-shake	-
From DCS to self-hosted Redis	DCS console	7 Migrating Data from DCS to Self-Hosted Redis	You can migrate data online from a DCS Redis instance to your self-hosted Redis by using the DCS console.

4 Migrating Data from Self-Hosted Redis to DCS

4.1 Online Migration of Self-Hosted Redis

Application Scenarios

If the source and target instances are interconnected and the **SYNC** and **PSYNC** commands are supported by the source instance, data can be migrated online in full or incrementally from the source to the target.

CAUTION

- If the **SYNC** and **PSYNC** commands are disabled on the source Redis instance, enable them before performing online migration. Otherwise, the migration fails. If you use a HUAWEI CLOUD DCS Redis instance for online migration, the **SYNC** command is automatically enabled.
 - You cannot use public networks for online migration.
-

Impacts on Services

During online migration, data is essentially synchronized in full to a new replica. Therefore, perform online migration during low-demand hours.

Prerequisites

- Before migrating data, read through [Migration Tools and Schemes](#) to learn about the DCS data migration function and select an appropriate target instance.
- To migrate data from a single-node or master/standby instance to a Redis Cluster instance, check if any data exists in DBs other than DB0 in the source instance. If yes, move the data to DB0. Otherwise, the migration will fail because a Redis Cluster instance has only one DB.

Step 1: Obtain the Source Redis Address

Obtain the IP address/domain name and port number of the source Redis instance.

Step 2: Prepare the Target DCS Redis Instance

- If a target DCS Redis instance is not available, create one first. For details, see [Buying a DCS Redis Instance](#).
- If you already have a DCS Redis instance, you do not need to create one again, but you need to clear the instance data before the migration. For details, see [Clearing DCS Instance Data](#).

If the target instance data is not cleared before the migration and the source and target instances contain the same key, the key in the target instance will be overwritten by the key in the source instance after the migration.

Step 3: Check the Network

Step 1 Check whether the source Redis instance, the target Redis instance, and the migration task are configured with the same VPC.

If yes, go to [Step 5: Configure the Online Migration Task](#). If no, go to [Step 2](#).

Step 2 Check whether the VPCs configured for the source Redis instance, the target Redis instance, and the migration task are connected to ensure that the VM resource of the migration task can access the source and target Redis instances.

If yes, go to [Step 1](#). If no, go to [Step 3](#).

Step 3 Perform the following operations to establish the network.

- If the source and target Redis instances are in the same region, create a VPC peering connection by referring to [VPC Peering Connection](#).
- If the source and target Redis instances are in different regions, create a cloud connection by referring to [Cloud Connect Getting Started](#).
- If the source and target Redis instances are on different clouds, create a connection by referring to [Direct Connect documentation](#).

----End

Step 4: Create an Online Migration Task

Step 1 Log in to the DCS console.

Step 2 In the navigation pane, choose **Data Migration**.

Step 3 Click **Create Online Migration Task**.

Step 4 Enter the task name and description.

Step 5 Configure the VPC, subnet, and security group for the migration task.

The VPC, subnet, and security group facilitate the migration. Ensure that the migration resources can access the source and target Redis instances.

----End

Step 5: Configure the Online Migration Task

Step 1 On the **Online Migration** tab page, click **Configure** in the row containing the online migration task you just created.

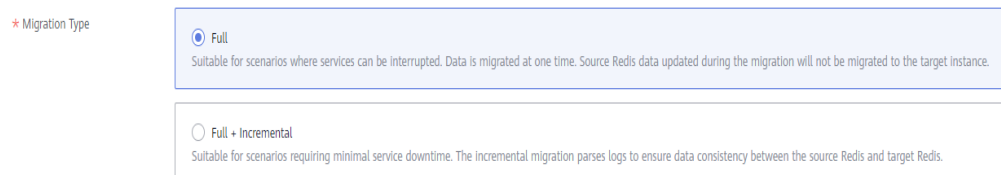
Step 2 Select a migration type.

Supported migration types are **Full** and **Full + Incremental**, which are described in [Table 4-1](#).

Table 4-1 Migration type description

Migration Type	Description
Full	Suitable for scenarios where services can be interrupted. Data is migrated at one time. Source instance data updated during the migration will not be migrated to the target instance.
Full + incremental	Suitable for scenarios requiring minimal service downtime. The incremental migration parses logs to ensure data consistency between the source and target instances. Once the migration starts, it remains Migrating until you click Stop in the Operation column. After the migration is stopped, data in the source instance will not be lost, but data will not be written to the target instance.

Figure 4-1 Selecting the migration type



Step 3 Configure source Redis and target Redis.

- Source Redis Type:** Select **Redis in the cloud** or **Self-hosted Redis** as required.
 - Redis in the cloud:** a HUAWEI CLOUD DCS Redis instance that is in the same VPC as the migration task
 - Self-hosted Redis:** self-hosted Redis on HUAWEI CLOUD, in another cloud, or in on-premises data centers. If you select this option, enter Redis addresses.
- If the instance is password-protected, you can click **Test Connection** to check whether the instance password is correct and whether the network is connected.

Step 4 For **Target Redis Instance**, select the DCS Redis Instance prepared in [Step 2: Prepare the Target DCS Redis Instance](#).

If the instance is password-protected, you can click **Test Connection** to check whether the instance password meets the requirements.

Step 5 Confirm the migration task details and click **Submit**.

Go back to the data migration task list. After the migration is successful, the task status changes to **Successful**.

 **NOTE**

Once incremental migration starts, it remains **Migrating** until you click **Stop**.

If the migration fails, click the migration task and check the log on the **Migration Logs** page.

----End

Verifying the Migration

After the migration is complete, use redis-cli to connect the source and target Redis instances to check data integrity.

1. Connect to the source Redis and the target Redis.
2. Run the **info keyspace** command to check the values of **keys** and **expires**.

```
192.168.1.217:6379> info keyspace
# Keyspace
db0:keys=81869,expires=0,avg_ttl=0
192.168.1.217:6379>
```

3. Calculate the difference between the values of **keys** and **expires** of the source Redis and the target Redis.

If the differences are the same, the data is complete and the migration is successful.

During full migration, source Redis data updated during the migration will not be migrated to the target instance.

4.2 Backup Migration of Self-Hosted Redis

Application Scenarios

Use the DCS console to migrate Redis data from Redis of another cloud or self-hosted Redis to HUAWEI CLOUD DCS for Redis.

Simply download the source Redis data and then upload the data to an OBS bucket in the same region as the target DCS Redis instance. After you have created a migration task on the DCS console, DCS will read data from the OBS bucket and data will be migrated to the target instance.

.aof, .rbb, .zip, and .tar.gz files can be uploaded to OBS buckets. You can directly upload .aof and .rdb files or compress them into .zip or .tar.gz files before uploading.

Prerequisites

- The OBS bucket must be in the same region as the target DCS Redis instance. For example, they both reside in the CN North-Beijing1 region.
- The data files to be uploaded must be in the .aof, .rdb, .zip, or .tar.gz format.
- To migrate data from a single-node or master/standby Redis instance of another cloud, create a backup task and download the backup file.
- To migrate data from a cluster Redis instance of another cloud, download all backup files, upload all of them to the OBS bucket, and select all of them for the migration. Each backup file contains data for a shard of the instance.
- .rdb backup files of self-hosted Redis 5.0 cannot be imported. .rdb backup files of self-hosted Redis 3.0 or 4.0 can be exported using redis-cli. .rdb files of other cloud Redis can be exported only by creating backup tasks, and cannot be exported by running commands in redis-cli.
- Redis Cluster instances only support .rdb files.

Step 1: Prepare the Target DCS Redis Instance

- If a target DCS Redis instance is not available, create one first. For details, see [Buying a DCS Redis Instance](#).
- If you already have a DCS Redis instance, you do not need to create one again, but you need to clear the instance data before the migration. For details, see [Clearing DCS Instance Data](#).

You can use a DCS Redis 3.0, 4.0, or 5.0 instance as the target instance.

Step 2: Create an OBS Bucket and Upload Backup Files

Step 1 Upload the backup data files to the OBS bucket by using OBS Browser+.

If the backup file to be uploaded is smaller than 5 GB, go to step [Step 2](#) to upload the file using the OBS console.

If the backup file to be uploaded is larger than 5 GB, follow the [instructions](#) provided by OBS.

Step 2 On the OBS console, upload the backup data files to the OBS bucket.

Perform the following steps if the backup files are smaller than 5 GB:

1. Create an OBS bucket.

When creating an OBS bucket, pay attention to the configuration of the following parameters. For details on how to set other parameters, see [Creating a Bucket](#) in *OBS User Guide*.

- a. **Region:**

The OBS bucket must be in the same region as the target DCS Redis instance.

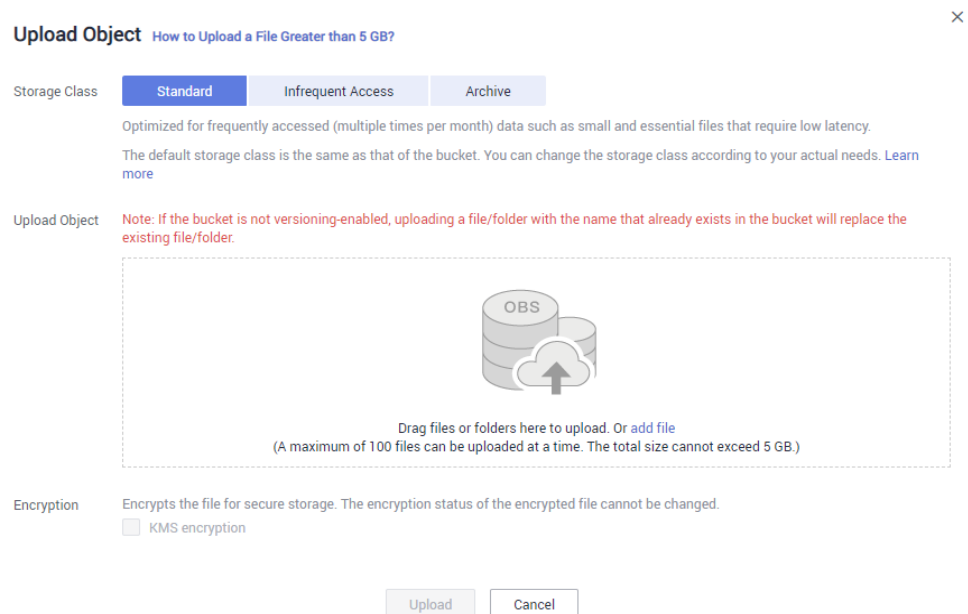
- b. **Storage Class:** Available options are **Standard**, **Infrequent Access**, and **Archive**.

Do not select **Archive**. Otherwise, the migration will fail.

- c. Click **Create Now**.

2. In the bucket list, click the bucket created in **Step 2.1**.
3. In the navigation pane, choose **Objects**.
4. On the **Objects** tab page, click **Upload Object**.
5. Specify **Storage Class**.
Do not select **Archive**. Otherwise, the migration will fail.
6. Upload the objects.
Drag files or folders to the **Upload Object** area or click **add file**.
A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.

Figure 4-2 Uploading objects in batches



7. (Optional) Select **KMS encryption** to encrypt the uploaded files.
8. Click **Upload**.

----End

Step 3: Create a Migration Task

- Step 1** Log in to the DCS console.
- Step 2** In the navigation pane, choose **Data Migration**.
- Step 3** Click **Create Backup Import Task**.
- Step 4** Enter the task name and description.
- Step 5** In the **Source Redis** area, select **OBS Bucket** for **Data Source** and then select the OBS bucket to which you have uploaded backup files.

In the **Backup Files** table, the files you have uploaded are displayed.

NOTE

You can upload files in the .aof, .rdb, .zip, or .tar.gz format.

Figure 4-3 Specifying the backup file information

Source Redis

* Data Source: OBS bucket | Redis

* OBS Bucket: 05041fffa40025702f6dc009cc6f8f33-hilens-s... View Bucket

* Backup Files: To import data from multiple backup files, you can create multiple migration tasks and run them at a time.

Name	Size
<input type="checkbox"/> tes...ns.zip	9.46 KB

Step 6 Select the backup files whose data is to be migrated.

Step 7 Select the target Redis instance prepared in **Step 1: Prepare the Target DCS Redis Instance**. If the target Redis instance has a password, enter the password and test the connection to check whether the password is correct.

Step 8 Click **Next**.

Step 9 Confirm the migration task details and click **Submit**.

Go back to the data migration task list. After the migration is successful, the task status changes to **Successful**.

----End

4.3 Self-Hosted Redis Migration with redis-cli (AOF)

Introduction

redis-cli is the command line tool of Redis, which can be used after you install the Redis server.

Run the following command to download Redis:

```
wget http://download.redis.io/releases/redis-5.0.8.tar.gz
```

This section describes how to use redis-cli to migrate a data from a self-hosted Redis instance to a HUAWEI CLOUD DCS instance.

Step 1: Generating an AOF File

NOTICE

- Before data migration, suspend your services so that data changes newly generated will not be lost during the migration.
- Migrate data during off-peak hours.

Run the following command to enable cache persistence and obtain an AOF persistence file:

```
redis-cli -h {source_redis_address} -p 6379 -a {password} config set  
appendonly yes
```

If the size of the AOF file does not change after you have enabled persistence, the AOF file contains full cached data.

 NOTE

- To find out the path for storing the AOF file, use redis-cli to access the Redis instance, and run the **config get dir** command. Unless otherwise specified, the file is named as **appendonly.aof** by default.
- To disable synchronization after the AOF file is generated, use redis-cli to log in to the Redis instance and run the **config set appendonly no** command.

Step 2: Uploading the AOF file to HUAWEI CLOUD ECS

1. To save the transmission time, compress the AOF file before transmission.
2. Upload the compressed file to HUAWEI CLOUD ECS using an appropriate mode (for example, SFTP mode).

 NOTE

Ensure that the ECS has sufficient disk space for data file decompression, and can communicate with the DCS instance. Generally, the ECS and DCS instance are configured to belong to the same VPC and subnet, and the configured security group rules do not restrict access ports. For details on how to configure a security group, see [How Do I Configure a Security Group?](#)

Step 3: Importing Data

```
redis-cli -h {dcs_instance_address} -p 6379 -a {password} --pipe <  
appendonly.aof
```

NOTICE

If SSL is enabled, replace the instance address and port number with the actual values.

Step 4: Verifying Migration

After the data is imported successfully, access the DCS instance and run the **info** command to check whether the data has been successfully imported as required.

If the data import fails, analyze the cause, modify the data import statement, run the **flushall** or **flushdb** command to clear the cached data in the instance, and import the data again.

Efficiency of Data Export and Import

An AOF file can be generated quickly. It applies to scenarios where you can access the Redis server and modify the configurations, such as scenarios with self-built Redis servers.

It takes 4s to 10s to import 1 million data records (20 bytes per data record) in a VPC.

4.4 Self-Hosted Redis Migration with redis-cli (RDB)

Introduction

redis-cli is the command line tool of Redis, which can be used after you install the Redis server.

redis-cli supports data export as an RDB file. If your Redis service does not support AOF file export, use redis-cli to obtain an RDB file. Then, use another tool (such as redis-port) to import the file to a DCS instance.

Operations described in this section are performed on the Linux OS.

Run the following command to download Redis. redis-cli can be used after installation and compilation.

```
wget http://download.redis.io/releases/redis-5.0.8.tar.gz
```

NOTICE

The source Redis instance must support the **SYNC** command, which is required when exporting the RDB file using redis-cli.

The **SYNC** command is not supported by DCS Reds 4.0 or 5.0 instances and cannot be used to export RDB files. To back up master/standby instance data, use the backup and restoration function provided by the DCS console.

Step 1: Preparation for Data Export

For master/standby or cluster DCS instances, there is a delay in writing data into an RDB file based on the delay policies configured in the **redis.conf** file. Therefore, before data export, learn the RDB policy configurations of the Redis instance to be migrated, suspend your service systems, and then write the required number of test keys into the Redis instance. This ensures that the RDB file is newly generated.

For the Redis service provided by a third-party cloud platform, you can contact its technical support to learn data writing policy configurations of an RDB file.

For example, the default RDB policy configurations in the **redis.conf** file are as follows:

```
save 900 1 //Writes changed data into an RDB file if there is any data change within 900s.  
save 300 10 //Writes changed data into an RDB file if there are more than 10 data changes within 300s.  
save 60 10000 //Writes changed data into an RDB file if there are more than 10,000 data changes within 60s.
```

Based on the preceding policy configurations, after stopping your service systems from writing data into the Redis instances, you can manually write test data to trigger the policies, so that all service data can be synchronized to the RDB file.

You can delete the test data after data import.

 NOTE

If there is any DB not used by your service systems, you can write test data into the DB, and run the **flushdb** command to clear the DB after importing data into DCS.

Step 2: Exporting an RDB File

NOTICE

1. Migrate data during off-peak hours.
2. When exporting Redis Cluster data, individually export the data of each node in the cluster, and then import the data node by node.

Run the following command to export the RDB file:

```
redis-cli -h {source_redis_address} -p 6379 -a {password} --rdb {output.rdb}
```

If "Transfer finished with success." is displayed after the command is executed, the file is exported successfully.

Step 3: Uploading the RDB File to HUAWEI CLOUD ECS

1. To save the transmission time, compress the RDB file before transmission.
2. Upload the compressed file to HUAWEI CLOUD ECS using an appropriate mode (for example, SFTP mode).

 NOTE

Ensure that the ECS has sufficient disk space for data file decompression, and can communicate with the DCS instance. Generally, the ECS and DCS instance are configured to belong to the same VPC and subnet, and the configured security group rules do not restrict access ports. For details on how to configure a security group, see [How Do I Configure a Security Group?](#)

Step 4: Importing Data

Use redis-port to import data.

Step 5: Verifying Migration

After the data is imported successfully, access the DCS instance and run the **info** command to check whether the data has been successfully imported as required.

If the data import fails, analyze the cause, modify the data import statement, run the **flushall** or **flushdb** command to clear the cached data in the instance, and import the data again.

Efficiency of Data Export and Import

Compared with master/standby instances, single-node instances without data persistence configured require a longer time for export of an RDB file, because the RDB file is temporarily generated.

It takes 4s to 10s to import 1 million data records (20 bytes per data record) in a VPC.

4.5 Self-Hosted Redis Migration with redis-port (RDB)

Introduction

redis-port is an open-source batch data transmission tool used for database synchronization between Redis nodes. redis-port provides the following functions:

- dump
Generates a cache snapshot and exports the cached data as an RDB file.
- decode
Parses the RDB file to check data distribution.
- restore
Imports the RDB file to a Redis instance.
- sync
Synchronizes the data in a Redis instance to another instance.

In this section, [redis-port V2.0-beta \(Linux\)](#) is used to describe how to migrate Redis data from HUAWEI CLOUD ECS to DCS.

Step 1: Installing redis-port

Download and decompress the tool package. redis-port can be used directly without compilation.

Install redis-port on both the ECS for data export and the ECS for data import.

```
wget https://github.com/CodisLabs/redis-port/releases/download/v2.0-beta/redis-port-v2.0-beta-go1.10.1-linux.tar.gz
```

```
tar -xvf redis-port-v2.0-beta-go1.10.1-linux.tar.gz
```

Step 2: Exporting Data

```
redis-dump -n 3 -m {password}@{source-redis-host}:{port} -o {outputfile.rdb}
```

Note: -n indicates that multiple CPUs process the export task concurrently.

 NOTE

1. The command format may vary depending on the version of installed redis-port. For details, see the related help document.
2. During data export and import, redis-port does not support certain special characters in passwords, such as the at sign (@) and number sign (#). If any connection information (such as passwords, instance addresses, and port numbers) fails to be parsed, remove the special characters from passwords temporarily.
3. When exporting Redis Cluster data, individually export the data of each node in the cluster, and then import the data node by node. For details, see [8.6 What Are the Constraints and Precautions for Migrating Redis Data to a Cluster Instance?](#)
4. redis-port-v2.0-beta-Go1.9.5 and earlier versions do not support migration of sorted sets. Sorted sets will change to sets after being imported using redis-port of such versions.
5. The **SYNC** command is not supported by DCS Reds 4.0 or 5.0 instances and cannot be used to export RDB files. To back up master/standby instance data, use the backup and restoration function provided by the DCS console.

Parameter description:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-dump --help
Usage:
  redis-dump [--ncpu=N] (--master=MASTER|MASTER) [--output=OUTPUT] [--aof=FILE]
  redis-dump --version

Options:
  -n N, --ncpu=N           Set runtime.GOMAXPROCS to N.
  -m MASTER, --master=MASTER The master redis instance ([auth@]host:port).
  -o OUTPUT, --output=OUTPUT Set output file. [default: /dev/stdout].
  -a FILE, --aof=FILE      Also dump the replication backlog.

Examples:
  $ redis-dump 127.0.0.1:6379 -o dump.rdb
  $ redis-dump 127.0.0.1:6379 -o dump.rdb -a
  $ redis-dump -m passwd@192.168.0.1:6380 -o dump.rdb -a dump.aof
```

Example:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-dump -n 3 -m Heru
+123@192.168.0.196:6379 -o save196.rdb
2018/03/26 09:10:28 dump.go:68: [INFO] dump: master = "Heru+123@192.168.0.196:6379", output =
"save196.rdb", aoflog = ""
2018/03/26 09:10:29 dump.go:111: [INFO] dump: runid = "a62dda896a855aef4a5429fd36fc4268882bc715",
offset = 204541
2018/03/26 09:10:29 dump.go:112: [INFO] dump: rdb file = 46721058 (44.56mb)
2018/03/26 09:10:29 dump.go:151: [INFO] dump: (w,a) = (rdb,aof)
2018/03/26 09:10:29 dump.go:181: [INFO] dump: rdb = 46721058 - [100.00%] (w,a)=(46721058,0) ~
(44.56mb,0)
2018/03/26 09:10:29 dump.go:185: [INFO] dump: done
root@redis-nodelete:~/port/redis-port-v2.0-alpha-go1.9.2-linux#
```

Step 3: Transmitting Data to HUAWEI CLOUD ECS

1. To save the transmission time, compress the RDB file before transmission.
2. Upload the compressed file to HUAWEI CLOUD ECS using an appropriate mode (for example, SFTP mode).

 NOTE

Ensure that the ECS has sufficient disk space for data file decompression, and can communicate with the DCS instance. Generally, the ECS and DCS instance are configured to belong to the same VPC and subnet, and the configured security group rules do not restrict access ports. For details on how to configure a security group, see [How Do I Configure a Security Group?](#)

Step 4: Importing Data

```
redis-restore -n {N} -i {outputfile.rdb} -t {password}@{dcs_instance_address}:  
{port} [--unixtime-in-milliseconds="yyyy-MM-dd hh:mm:ss"] [--  
db={DB_number}]
```

By specifying the **db** parameter, you can import the cached data of the specified DB in the file. **-n** indicates that multiple CPUs process the import task concurrently.

Parameter description:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-restore --help
Usage:
  redis-restore [--ncpu=N] [--input=INPUT|INPUT] --target=TARGET [--aof=FILE] [--db=DB] [--unixtime-  
in-milliseconds=EXPR]
  redis-restore --version

Options:
  -n N, --ncpu=N          Set runtime.GOMAXPROCS to N.
  -i INPUT, --input=INPUT Set input rdb encoded file.
  -t TARGET, --target=TARGET The target redis instance ([auth@]host:port).
  -a FILE, --aof=FILE     Also restore the replication backlog.
  --db=DB                 Accept db = DB, default is *.
  --unixtime-in-milliseconds=EXPR Update expire time when restoring objects from RDB.

Examples:
  $ redis-restore dump.rdb -t 127.0.0.1:6379
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --aof dump.aof --db=1
  $ redis-restore -t 127.0.0.1:6379 --aof dump.aof
  $ redis-restore -t 127.0.0.1:6379 --db=0
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="@209059200000" //
  ttlms += (now - '1976-08-17')
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="+1000" // ttlms +=
  1s
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="-1000" // ttlms -= 1s
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="1976-08-17 00:00:00" // ttlms
  += (now - '1976-08-17')
```

Example:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-restore -i save196.rdb -t Heru
+123@192.168.0.171:6379
2018/03/26 09:15:33 restore.go:70: [INFO] restore: input = "save196.rdb", aoflog = "" target = "Heru
+123@192.168.0.171:6379"
2018/03/26 09:15:33 restore.go:126: [INFO] restore: (r,f,s/a,f,s) = (rdb,rdb.forward,rdb.skip/
aof,rdb.forward,rdb.skip)
2018/03/26 09:15:34 restore.go:155: [INFO] restore: size = 46721058 - [ 49.94%, 0.00%] (r,f,s/
a,f,s)=(23330816,0,599496/0,0,0) ~ (22.25mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:155: [INFO] restore: size = 46721058 - [ 99.31%, 0.00%] (r,f,s/
a,f,s)=(46399488,12558,1179884/0,0,0) ~ (44.25mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:155: [INFO] restore: size = 46721058 - [100.00%, 0.00%] (r,f,s/
a,f,s)=(46721058,20000,1179884/0,0,0) ~ (44.56mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:159: [INFO] restore: done
root@redis-nodelete:~/port/redis-port-v2.0-alpha-go1.9.2-linux#
```

Step 5: Verifying Migration

After the data is imported successfully, access the DCS instance and run the **info** command to check whether the data has been successfully imported as required.

If the data import fails, analyze the cause, modify the data import statement, run the **flushall** or **flushdb** command to clear the cached data in the instance, and import the data again.

4.6 Self-Hosted Codis Migration with redis-cli and redis-port

Codis data can be smoothly migrated to DCS Proxy Cluster instances because DCS Proxy Clusters are based on the open-source Codis. The following describes the procedure of migrating Codis data using redis-cli and redis-port.

- From self-hosted Codis to DCS Proxy Cluster:
Self-hosted Codis deployed in a local data center, HUAWEI CLOUD, or another public cloud is supported. There are no restrictions on exporting data from self-hosted Codis.
- From Codis of another cloud to DCS Proxy Cluster:
Some cloud vendors have restrictions on data export commands. You can export data by downloading backup files on their consoles.

Preparation

- Create a DCS Proxy Cluster instance.
The available memory of the instance must be larger than the size of the data to be imported.
- Prepare a server which will be used for transmitting and importing data.
The server must be connected to the DCS instance and have sufficient disk space for storing migration data.
You can use HUAWEI CLOUD ECS and configure the same VPC, subnet, and security group for the ECS and the DCS instance.

Procedure

Step 1 Export the RDB file.

Download and install Redis: <http://download.redis.io/releases/redis-5.0.8.tar.gz>.

Run the following command to export the RDB file for each shard in the cluster:

```
redis-cli -h {source_redis_address} -p {source_redis_port} -a {source_redis_password} --rdb {output.rdb}
```

If "Transfer finished with success." is displayed after the command is executed, the file is exported successfully.

 NOTE

1. Migrate data during off-peak hours.
2. When exporting Codis data, individually export the data of each shard in the cluster, and then import the data file by file.

Step 2 Upload the RDB file to the server. HUAWEI CLOUD ECS is used as an example.

If the files are larger, compress them before transmission.

Step 3 Download redis-port.

Log in to the ECS. Run the following commands to download and decompress the tool package. redis-port can be used directly without compilation.

```
wget https://github.com/CodisLabs/redis-port/releases/download/v2.0-beta/redis-port-v2.0-beta-go1.10.1-linux.tar.gz
```

```
tar -xvf redis-port-v2.0-beta-go1.10.1-linux.tar.gz
```

Step 4 Import data.

Run the following commands in redis-port to import all RDB files to the target DCS instance. If the ECS and the DCS instance are in the same VPC, it takes 4 to 10 seconds to export 1 million data records, assuming that each record contains 20 byte data.

```
redis-restore -n {N} -i {outputfile.rdb} -t {password}@{dcs_instance_address}:{port} [--unixtime-in-milliseconds="yyyy-MM-dd hh:mm:ss"]
```

By specifying the **db** parameter, you can import the cached data of the specified DB in the file. **-n** indicates that multiple CPUs process the import task concurrently.

Parameter description:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-restore --help
Usage:
  redis-restore [--ncpu=N] [--input=INPUT|INPUT] --target=TARGET [--aof=FILE] [--db=DB] [--unixtime-in-milliseconds=EXPR]
  redis-restore --version

Options:
  -n N, --ncpu=N          Set runtime.GOMAXPROCS to N.
  -i INPUT, --input=INPUT Set input rdb encoded file.
  -t TARGET, --target=TARGET The target redis instance ([auth@]host:port).
  -a FILE, --aof=FILE     Also restore the replication backlog.
  --db=DB                 Accept db = DB, default is *.
  --unixtime-in-milliseconds=EXPR Update expire time when restoring objects from RDB.

Examples:
$ redis-restore dump.rdb -t 127.0.0.1:6379
$ redis-restore -i dump.rdb -t 127.0.0.1:6379 --aof dump.aof --db=1
$ redis-restore -t 127.0.0.1:6379 --aof dump.aof
$ redis-restore -t 127.0.0.1:6379 --db=0
$ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="@209059200000" //
ttlms += (now - '1976-08-17')
$ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="+1000" // ttlms +=
1s
$ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="-1000" // ttlms -= 1s
$ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="1976-08-17 00:00:00" // ttlms
+= (now - '1976-08-17')
```

Example:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-restore -i save196.rdb -t
*****@192.168.0.171:6379
2018/03/26 09:15:33 restore.go:70: [INFO] restore: input = "save196.rdb", aoflog = "" target =
*****@192.168.0.171:6379"
2018/03/26 09:15:33 restore.go:126: [INFO] restore: (r,f,s/a,f,s) = (rdb,rdb.forward,rdb.skip/
aof,rdb.forward,rdb.skip)
2018/03/26 09:15:34 restore.go:155: [INFO] restore: size = 46721058 - [ 49.94%, 0.00%] (r,f,s/
a,f,s)=(23330816,0,599496/0,0,0) ~ (22.25mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:155: [INFO] restore: size = 46721058 - [ 99.31%, 0.00%] (r,f,s/
a,f,s)=(46399488,12558,1179884/0,0,0) ~ (44.25mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:155: [INFO] restore: size = 46721058 - [100.00%, 0.00%] (r,f,s/
a,f,s)=(46721058,20000,1179884/0,0,0) ~ (44.56mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:159: [INFO] restore: done
root@redis-nodelete:~/port/redis-port-v2.0-alpha-go1.9.2-linux#
```

Step 5 Verify the migration.

After the migration is completed, access the DCS instance and run the **info** command to check whether the data has been successfully imported as required.

If the data import fails, run the **flushall** or **flushdb** command to clear the cached data in the instance, and import the data again.

----End

4.7 Self-Hosted Redis Cluster Migration with redis-shake

redis-shake is an open-source tool for migrating data online or offline (by importing backup files) between Redis Clusters. Data can be migrated to DCS Redis Cluster instances seamlessly because DCS Redis Cluster inherits the native Redis Cluster design.

The following describes how to use [redis-shake](#) to migrate data to a DCS Redis Cluster instance.

Migrating Data Online

You can migrate data online from a self-hosted Redis Cluster to a DCS Redis Cluster instance as long as the two clusters are directly connected or connected through a transit server.

Data in Redis Clusters of another cloud cannot be migrated online because the **SYNC** and **PSYNC** commands are disabled by some vendors.

1. Create a Redis Cluster instance on the DCS console.
The memory of this instance cannot be smaller than that of the source Redis.
2. Prepare a cloud server and install redis-shake.

redis-shake must be able to access both the source and target Redis. Bound an EIP to the cloud server.

You can use HUAWEI CLOUD ECS and configure the same VPC, subnet, and security group for the ECS and the DCS instance. If the source Redis is deployed on cloud servers of another cloud, allow public access to the servers.

[Download](#) and decompress the release version of redis-shake.

```
[root@ecs-p-4-centos redisshake]# ll
total 16972
-rw-r--r-- 1 1320024 users      2749 Jun 24 16:15 ChangeLog
-rwxr-xr-x 1 1320024 users     14225 Jun 24 16:14 hypervisor
-rwxr-xr-x 1 1320024 users  13000971 Jun 24 16:14 redis-shake
-rw-r--r-- 1 1320024 users      8875 Jun 24 16:15 redis-shake.conf
-rw-r--r-- 1 root      root    4326892 Jun 24 16:17 redis-shake.tar.g
-rwxr-xr-x 1 1320024 users       458 Jun 24 16:14 start.sh
-rwxr-xr-x 1 1320024 users       374 Jun 24 16:14 stop.sh
```

3. Locate the masters of the source and target Redis Clusters and obtain the IP addresses of the masters.

Online data migration must be performed node by node. Run the following command to query the IP addresses and port numbers of all nodes in both the source and target Redis Clusters.

redis-cli -h {redis_address} -p {redis_port} -a {redis_password} cluster nodes

In the command output similar to the following, obtain the IP addresses and ports of all masters.

```
[root@ecs-p-4-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a 23 cluster nodes
fb75f0743af4695a3d241ff7790b2f508e4985ff 192.168.0.140:6379@16379 myself,master - 0 1562144170000 3 connected
d112bae791b2bbd9602fe32963536b8a0db9eb79 192.168.0.61:6379@16379 master - 0 1562144171524 1 connected 0-5460
73e2f8fe196166f9ad1283361867d24c136413f0 192.168.0.194:6379@16379 master - 0 1562144170000 2 connected 5461-14
40d72299fde6045de0f79ee4b97910b505acbc6a 192.168.0.231:6379@16379 slave 73e2f8fe196166f9ad1283361867d24c136413
be6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985f
c16b9acaee7dd0721f129596cd43bd499c0e396 192.168.0.169:6379@16379 slave d112bae791b2bbd9602fe32963536b8a0db9eb
```

 **NOTE**

After Redis is installed, it runs with redis-cli. To install Redis on CentOS, run the **yum install redis** command.

4. Edit the redis-shake configuration file.

Edit the **redis-shake.conf** file, providing the following information about all the masters of both the source and the target:

```
source.type = cluster
# If there is no password, skip the following parameter.
source.password_raw = {source_redis_password}
# IP addresses and port numbers of all masters of the source Redis Cluster, which are separated by
semicolons (;).
source.address = {master1_ip}:{master1_port};{master2_ip}:{master2_port}...{masterN_ip}:
{masterN_port}
target.type = cluster
# If there is no password, skip the following parameter.
target.password_raw = {target_redis_password}
# IP addresses and port numbers of all masters of the target instance, which are separated by
semicolons (;).
target.address = {master1_ip}:{master1_port};{master2_ip}:{master2_port}...{masterN_ip}:
{masterN_port}
```

Save and exit.

5. Migrate data online.

Run the following command to synchronize data between the source and the target Redis:

./redis-shake -type sync -conf redis-shake.conf

If the following information is displayed, the full synchronization has been completed and incremental synchronization begins.

```
sync rdb done.
```

If the following information is displayed, no new data is incremented. You can stop the incremental synchronization by pressing **Ctrl+C**.

```
sync: +forwardCommands=0 +filterCommands=0 +writeBytes=0
```

Figure 4-4 Online migration using redis-shake

```

root@ecs-p-54-centos redisshake# ./redis-shake -type sync -conf redis-shake.conf
2019/07/03 20:22:53 [INFO] the target redis type is cluster, only pass db0
2019/07/03 20:22:53 [WARN]
redis-shake, here we go !!
if you have any problem, please visit https://github.com/alibaba/RedisShake/wiki/FAQ
2019/07/03 20:22:53 [INFO] redis-shake configuration: {"Id":"redis-shake","LogFile":"","LogLevel":"info","SystemProfile":9310,"HttpProfile":9320,"Ncpu":0,"Parallel":32,"SourceType":"standalone","SourceAddress":"127.0.0.1:6379","SourcePasswordRaw":"","SourcePasswordEncoding":"","SourceVersion":0,"SourceAuthType":"auth","SourceParallel":1,"SourceTlsEnable":false,"TargetAddress":"192.168.0.140:6379;192.168.0.61:6379;192.168.0.194:6379","TargetPasswordRaw":"","TargetPasswordEncoding":"","TargetVersion":0,"TargetDBString":"-1","TargetAuthType":"auth","TargetType":"cluster","TargetTlsEnable":false,"RdbInput":["local"],"RdbOutput":["local_dump"],"RdbParallel":1,"RdbSpecialCloud":"","FakeTime":"","Rewrite":true,"FilterDB":"","FilterKey":"","FilterSlot":"","FilterLua":false,"BigKeyThreshold":524288000,"Psync":false,"Metric":true,"MetricPrintLog":false,"HeartbeatUrl":"","HeartbeatInterval":3,"HeartbeatExternal":"test external","HeartbeatNetworkInterface":"","SenderSize":104857600,"SenderCount":5000,"SenderDelayChannelSize":65535,"KeepAlive":0,"PidPath":"","ScanKeyNumber":50,"ScanSpecialCloud":"","ScanKeyFile":"","Ops":200000,"ReplaceHashTag":false,"ExtraInfo":false,"SockFileName":"","SockFileSize":0,"SourceAddressList":["127.0.0.1:6379"],"TargetAddressList":["192.168.0.140:6379","192.168.0.61:6379","192.168.0.194:6379"],"HeartbeatIp":"127.0.0.1","ShiftTime":0,"TargetRedisVersion":"5.0.5","TargetReplace":true,"TargetDB":-1,"Version":"improve-1.6.9","737b9045681eb923830add743606075c71b748,go1.10.3,2019-06-24_16:14:53"}
2019/07/03 20:22:53 [INFO] routine[0] starts syncing data from 127.0.0.1:6379 to [192.168.0.140:6379 192.168.0.61:6379 192.168.0.194:6379] with http[9321]
2019/07/03 20:22:53 [INFO] input password is empty, skip auth address[127.0.0.1:6379] with type[auth].
2019/07/03 20:22:54 [INFO] dbSyncer[0] rdb file size = 18968654
2019/07/03 20:22:54 [INFO] Aux information key:redis-ver value:3.2.12
2019/07/03 20:22:54 [INFO] Aux information key:redis-bits value:64
2019/07/03 20:22:54 [INFO] Aux information key:ctime value:1562156573
2019/07/03 20:22:54 [INFO] Aux information key:used-mem value:90132832
2019/07/03 20:22:54 [INFO] db_size:632286 expire_size:0
2019/07/03 20:22:55 [INFO] dbSyncer[0] total=18968654 - 1456787 [ 7%] entry=47532
2019/07/03 20:22:56 [INFO] dbSyncer[0] total=18968654 - 2983937 [ 15%] entry=98437

```

6. Verify the migration.

After data synchronization, access the DCS Redis Cluster instance using redis-cli. Run the **info** command to query the number of keys in the **Keyspace** section to confirm that data has been fully imported.

If the data has not been fully imported, run the **flushall** or **flushdb** command to clear the cached data in the instance, and synchronize data again.

7. Clear the redis-shake configuration file.

Importing Backup Files

If the source Redis and the destination Redis cannot be connected, or the source Redis is deployed on other clouds, you can migrate data by importing backup files.

1. Create a Redis Cluster instance on the DCS console.

The memory of this instance cannot be smaller than that of the source Redis.

2. Run the following command to obtain the IP addresses and port numbers of all masters of the source Redis and target Redis:

```
redis-cli -h {redis_address} -p {redis_port} -a {redis_password} cluster nodes
```

In the command output similar to the following, obtain the IP addresses and ports of all masters.

```

[root@ecs-p-54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a 23 cluster nodes
fb75f0743af4695a3d241ff7790b2f508e4985ff 192.168.0.140:6379@16379 myself,master - 0 1562144170000 3 connected
d112bae791b2bbd9602fe32963536b8a0db9eb79 192.168.0.61:6379@16379 master - 0 1562144171524 1 connected 0-5460
73e2f8fe196166f9ad1283361867d24c136413f0 192.168.0.194:6379@16379 master - 0 1562144170000 2 connected 5461-1040d72299fd6e045de0f79ee4b97910b505acbc6a 192.168.0.231:6379@16379 slave 73e2f8fe196166f9ad1283361867d24c136413f0
e6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985ff
c16b9acaeeed7dd0721f129596cd43bd499c0e396 192.168.0.169:6379@16379 slave d112bae791b2bbd9602fe32963536b8a0db9eb79

```

NOTE

After Redis is installed, it runs with redis-cli. To install Redis on CentOS, run the **yum install redis** command.

3. Prepare a cloud server and install redis-shake.

redis-shake must be able to access the target Redis and bound to an EIP.

You can use HUAWEI CLOUD ECS and configure the same VPC, subnet, and security group for the ECS and the DCS instance.

Download and decompress the release version of redis-shake.

```
[root@ecs-p-4-centos redisshake]# ll
total 16972
-rw-r--r-- 1 1320024 users 2749 Jun 24 16:15 ChangeLog
-rwxr-xr-x 1 1320024 users 14225 Jun 24 16:14 hypervisor
-rwxr-xr-x 1 1320024 users 13000971 Jun 24 16:14 redis-shake
-rw-r--r-- 1 1320024 users 8875 Jun 24 16:15 redis-shake.conf
-rw-r--r-- 1 root root 4326892 Jun 24 16:17 redis-shake.tar.g
-rwxr-xr-x 1 1320024 users 458 Jun 24 16:14 start.sh
-rwxr-xr-x 1 1320024 users 374 Jun 24 16:14 stop.sh
```

 **NOTE**

If the source Redis is deployed in the data center intranet, install redis-shake on the intranet server. Export data and then upload the data to the cloud server as instructed by the following steps

4. Export the RDB file.

- Edit the **redis-shake.conf** file by providing the following information about all the masters of both the source and the target:

```
source.type = cluster
# If there is no password, skip the following parameter.
source.password_raw = {source_redis_password}
# IP addresses and port numbers of all masters of the source Redis Cluster, which are separated by semicolons (;).
source.address = {master1_ip}:{master1_port};{master2_ip}:{master2_port}...{masterN_ip}:{masterN_port}
```

- Run the following command to export the RDB file:

```
./redis-shake -type dump -conf redis-shake.conf
```

If the following information is displayed in the execution log, the backup file is exported successfully:

```
execute runner[*run.CmdDump] finished!
```

5. Import the RDB file.

- Import the RDB file (or files) to the cloud server. The cloud server must be connected to the target DCS instance.
- Edit the redis-shake configuration file.

Edit the **redis-shake.conf** file by providing the following information about all the masters of both the source and the target:

```
target.type = cluster
# If there is no password, skip the following parameter.
target.password_raw = {target_redis_password}
# IP addresses and port numbers of all masters of the target instance, which are separated by semicolons (;).
target.address = {master1_ip}:{master1_port};{master2_ip}:{master2_port}...{masterN_ip}:{masterN_port}
# List the RDB files to be imported, separated by semicolons (;).
rdb.input = local_dump.0;local_dump.1;local_dump.2;local_dump.3
```

Save and exit.

- Run the following command to import the RDB file to the target instance:

```
./redis-shake -type restore -conf redis-shake.conf
```

If the following information is displayed in the execution log, the backup file is imported successfully:

```
Enabled http stats, set status (incr), and wait forever.
```

6. Verify the migration.

After data synchronization, access the DCS Redis Cluster instance using `redis-cli`. Run the **info** command to query the number of keys in the **Keyspace** section to confirm that data has been fully imported.

If the data has not been fully imported, run the **flushall** or **flushdb** command to clear the cached data in the instance, and synchronize data again.

5 Migrating Data Between DCS Instances

5.1 Online Migration Between DCS Redis Instances

Application Scenarios

If the source and target instances are interconnected and the **SYNC** and **PSYNC** commands are supported by the source instance, data can be migrated online in full or incrementally from the source to the target.

⚠ CAUTION

- If the **SYNC** and **PSYNC** commands are disabled on the source Redis instance, enable them before performing online migration. Otherwise, the migration fails. If you use a HUAWEI CLOUD DCS Redis instance for online migration, the **SYNC** command is automatically enabled.
 - You cannot use public networks for online migration.
-

Impacts on Services

During online migration, data is essentially synchronized in full to a new replica. Therefore, perform online migration during low-demand hours.

Prerequisites

- Before migrating data, read through [Migration Tools and Schemes](#) to learn about the DCS data migration function and select an appropriate target instance.
- To migrate data from a single-node or master/standby instance to a Redis Cluster instance, check if any data exists in DBs other than DB0 in the source instance. If yes, move the data to DB0. Otherwise, the migration will fail because a Redis Cluster instance has only one DB.

Step 1: Obtain the Source Redis Address

Obtain the IP address/domain name and port number of the source Redis instance.

Step 2: Prepare the Target DCS Redis Instance

- If a target DCS Redis instance is not available, create one first. For details, see [Buying a DCS Redis Instance](#).
- If you already have a DCS Redis instance, you do not need to create one again, but you need to clear the instance data before the migration. For details, see [Clearing DCS Instance Data](#).

If the target instance data is not cleared before the migration and the source and target instances contain the same key, the key in the target instance will be overwritten by the key in the source instance after the migration.

Step 3: Check the Network

Step 1 Check whether the source Redis instance, the target Redis instance, and the migration task are configured with the same VPC.

If yes, go to [Step 5: Configure the Online Migration Task](#). If no, go to [Step 2](#).

Step 2 Check whether the VPCs configured for the source Redis instance, the target Redis instance, and the migration task are connected to ensure that the VM resource of the migration task can access the source and target Redis instances.

If yes, go to [Step 1](#). If no, go to [Step 3](#).

Step 3 Perform the following operations to establish the network.

- If the source and target Redis instances are in the same region, create a VPC peering connection by referring to [VPC Peering Connection](#).
- If the source and target Redis instances are in different regions, create a cloud connection by referring to [Cloud Connect Getting Started](#).
- If the source and target Redis instances are on different clouds, create a connection by referring to [Direct Connect documentation](#).

----End

Step 4: Create an Online Migration Task

Step 1 Log in to the DCS console.

Step 2 In the navigation pane, choose **Data Migration**.

Step 3 Click **Create Online Migration Task**.

Step 4 Enter the task name and description.

Step 5 Configure the VPC, subnet, and security group for the migration task.

The VPC, subnet, and security group facilitate the migration. Ensure that the migration resources can access the source and target Redis instances.

----End

Step 5: Configure the Online Migration Task

Step 1 On the **Online Migration** tab page, click **Configure** in the row containing the online migration task you just created.

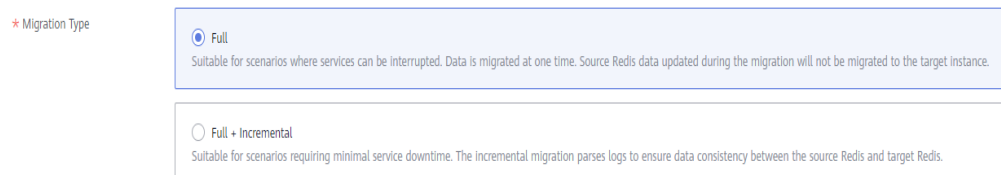
Step 2 Select a migration type.

Supported migration types are **Full** and **Full + Incremental**, which are described in [Table 5-1](#).

Table 5-1 Migration type description

Migration Type	Description
Full	Suitable for scenarios where services can be interrupted. Data is migrated at one time. Source instance data updated during the migration will not be migrated to the target instance.
Full + incremental	Suitable for scenarios requiring minimal service downtime. The incremental migration parses logs to ensure data consistency between the source and target instances. Once the migration starts, it remains Migrating until you click Stop in the Operation column. After the migration is stopped, data in the source instance will not be lost, but data will not be written to the target instance.

Figure 5-1 Selecting the migration type



Step 3 Configure source Redis and target Redis.

- Source Redis Type:** Select **Redis in the cloud** or **Self-hosted Redis** as required.
 - Redis in the cloud:** a HUAWEI CLOUD DCS Redis instance that is in the same VPC as the migration task
 - Self-hosted Redis:** self-hosted Redis on HUAWEI CLOUD, in another cloud, or in on-premises data centers. If you select this option, enter Redis addresses.
- If the instance is password-protected, you can click **Test Connection** to check whether the instance password is correct and whether the network is connected.

Step 4 For **Target Redis Instance**, select the DCS Redis Instance prepared in [Step 2: Prepare the Target DCS Redis Instance](#).

If the instance is password-protected, you can click **Test Connection** to check whether the instance password meets the requirements.

Step 5 Confirm the migration task details and click **Submit**.

Go back to the data migration task list. After the migration is successful, the task status changes to **Successful**.

 **NOTE**

Once incremental migration starts, it remains **Migrating** until you click **Stop**.

If the migration fails, click the migration task and check the log on the **Migration Logs** page.

----End

Verifying the Migration

After the migration is complete, use redis-cli to connect the source and target Redis instances to check data integrity.

1. Connect to the source Redis and the target Redis.
2. Run the **info keyspace** command to check the values of **keys** and **expires**.

```
192.168.1.217:6379> info keyspace
# Keyspace
db0:keys=81869,expires=0,avg_ttl=0
192.168.1.217:6379>
```

3. Calculate the difference between the values of **keys** and **expires** of the source Redis and the target Redis.

If the differences are the same, the data is complete and the migration is successful.

During full migration, source Redis data updated during the migration will not be migrated to the target instance.

5.2 Backup Migration Between Regions or Redis Versions

Application Scenarios

Use the DCS console to migrate Redis data from Redis of another cloud or self-hosted Redis to HUAWEI CLOUD DCS for Redis.

Simply download the source Redis data and then upload the data to an OBS bucket in the same region as the target DCS Redis instance. After you have created a migration task on the DCS console, DCS will read data from the OBS bucket and data will be migrated to the target instance.

.aof, .rbb, .zip, and .tar.gz files can be uploaded to OBS buckets. You can directly upload .aof and .rdb files or compress them into .zip or .tar.gz files before uploading.

Prerequisites

- The OBS bucket must be in the same region as the target DCS Redis instance. For example, they both reside in the CN North-Beijing1 region.
- The data files to be uploaded must be in the .aof, .rdb, .zip, or .tar.gz format.
- To migrate data from a single-node or master/standby Redis instance of another cloud, create a backup task and download the backup file.
- To migrate data from a cluster Redis instance of another cloud, download all backup files, upload all of them to the OBS bucket, and select all of them for the migration. Each backup file contains data for a shard of the instance.
- .rdb backup files of self-hosted Redis 5.0 cannot be imported. .rdb backup files of self-hosted Redis 3.0 or 4.0 can be exported using redis-cli. .rdb files of other cloud Redis can be exported only by creating backup tasks, and cannot be exported by running commands in redis-cli.
- Redis Cluster instances only support .rdb files.

Step 1: Prepare the Target DCS Redis Instance

- If a target DCS Redis instance is not available, create one first. For details, see [Buying a DCS Redis Instance](#).
- If you already have a DCS Redis instance, you do not need to create one again, but you need to clear the instance data before the migration. For details, see [Clearing DCS Instance Data](#).

You can use a DCS Redis 3.0, 4.0, or 5.0 instance as the target instance.

Step 2: Create an OBS Bucket and Upload Backup Files

Step 1 Upload the backup data files to the OBS bucket by using OBS Browser+.

If the backup file to be uploaded is smaller than 5 GB, go to step [Step 2](#) to upload the file using the OBS console.

If the backup file to be uploaded is larger than 5 GB, follow the [instructions](#) provided by OBS.

Step 2 On the OBS console, upload the backup data files to the OBS bucket.

Perform the following steps if the backup files are smaller than 5 GB:

1. Create an OBS bucket.

When creating an OBS bucket, pay attention to the configuration of the following parameters. For details on how to set other parameters, see [Creating a Bucket](#) in *OBS User Guide*.

a. **Region:**

The OBS bucket must be in the same region as the target DCS Redis instance.

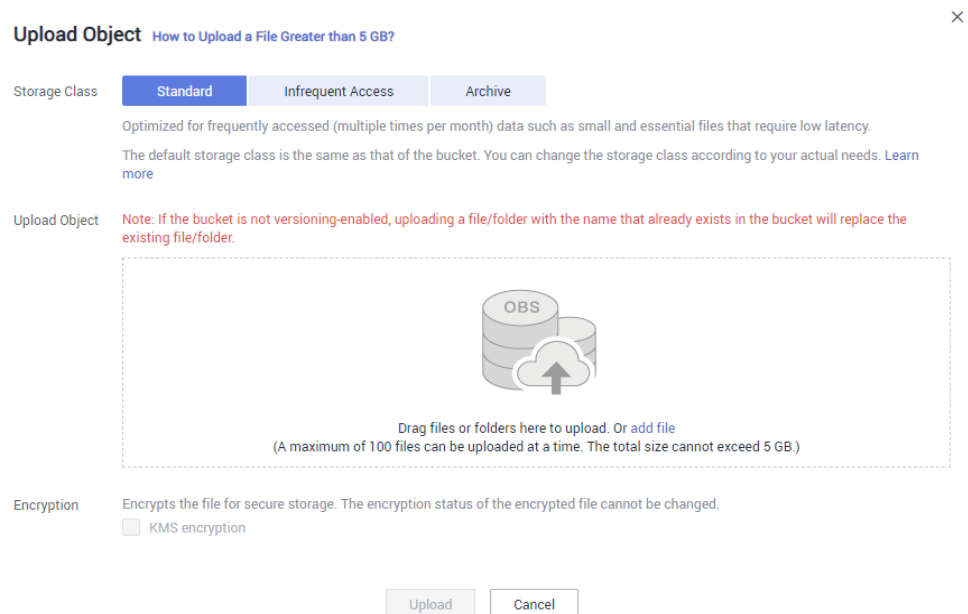
b. **Storage Class:** Available options are **Standard**, **Infrequent Access**, and **Archive**.

Do not select **Archive**. Otherwise, the migration will fail.

c. Click **Create Now**.

2. In the bucket list, click the bucket created in **Step 2.1**.
3. In the navigation pane, choose **Objects**.
4. On the **Objects** tab page, click **Upload Object**.
5. Specify **Storage Class**.
Do not select **Archive**. Otherwise, the migration will fail.
6. Upload the objects.
Drag files or folders to the **Upload Object** area or click **add file**.
A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.

Figure 5-2 Uploading objects in batches



7. (Optional) Select **KMS encryption** to encrypt the uploaded files.
8. Click **Upload**.

----End

Step 3: Create a Migration Task

- Step 1** Log in to the DCS console.
- Step 2** In the navigation pane, choose **Data Migration**.
- Step 3** Click **Create Backup Import Task**.
- Step 4** Enter the task name and description.
- Step 5** In the **Source Redis** area, select **OBS Bucket** for **Data Source** and then select the OBS bucket to which you have uploaded backup files.

In the **Backup Files** table, the files you have uploaded are displayed.

NOTE

You can upload files in the .aof, .rdb, .zip, or .tar.gz format.

Figure 5-3 Specifying the backup file information

Source Redis

* Data Source OBS bucket Redis

* OBS Bucket [View Bucket](#)

* Backup Files

To import data from multiple backup files, you can create multiple migration tasks and run them at a time.

Name	Size
<input type="checkbox"/> test-ns.zip	9.46 KB

Step 6 Select the backup files whose data is to be migrated.

Step 7 Select the target Redis instance prepared in **Step 1: Prepare the Target DCS Redis Instance**. If the target Redis instance has a password, enter the password and test the connection to check whether the password is correct.

Step 8 Click **Next**.

Step 9 Confirm the migration task details and click **Submit**.

Go back to the data migration task list. After the migration is successful, the task status changes to **Successful**.

----End

6 Migrating Redis Data from Another Cloud to DCS

6.1 Online Migration from Another Cloud

Application Scenarios

If the source and target instances are interconnected and the **SYNC** and **PSYNC** commands are supported by the source instance, data can be migrated online in full or incrementally from the source to the target.

CAUTION

- If the **SYNC** and **PSYNC** commands are disabled on the source Redis instance, enable them before performing online migration. Otherwise, the migration fails. If you use a HUAWEI CLOUD DCS Redis instance for online migration, the **SYNC** command is automatically enabled.
 - You cannot use public networks for online migration.
-

Impacts on Services

During online migration, data is essentially synchronized in full to a new replica. Therefore, perform online migration during low-demand hours.

Prerequisites

- Before migrating data, read through [Migration Tools and Schemes](#) to learn about the DCS data migration function and select an appropriate target instance.
- To migrate data from a single-node or master/standby instance to a Redis Cluster instance, check if any data exists in DBs other than DB0 in the source instance. If yes, move the data to DB0. Otherwise, the migration will fail because a Redis Cluster instance has only one DB.

Step 1: Obtain the Source Redis Address

Obtain the IP address/domain name and port number of the source Redis instance.

Step 2: Prepare the Target DCS Redis Instance

- If a target DCS Redis instance is not available, create one first. For details, see [Buying a DCS Redis Instance](#).
- If you already have a DCS Redis instance, you do not need to create one again, but you need to clear the instance data before the migration. For details, see [Clearing DCS Instance Data](#).

If the target instance data is not cleared before the migration and the source and target instances contain the same key, the key in the target instance will be overwritten by the key in the source instance after the migration.

Step 3: Check the Network

Step 1 Check whether the source Redis instance, the target Redis instance, and the migration task are configured with the same VPC.

If yes, go to [Step 5: Configure the Online Migration Task](#). If no, go to [Step 2](#).

Step 2 Check whether the VPCs configured for the source Redis instance, the target Redis instance, and the migration task are connected to ensure that the VM resource of the migration task can access the source and target Redis instances.

If yes, go to [Step 1](#). If no, go to [Step 3](#).

Step 3 Perform the following operations to establish the network.

- If the source and target Redis instances are in the same region, create a VPC peering connection by referring to [VPC Peering Connection](#).
- If the source and target Redis instances are in different regions, create a cloud connection by referring to [Cloud Connect Getting Started](#).
- If the source and target Redis instances are on different clouds, create a connection by referring to [Direct Connect documentation](#).

----End

Step 4: Create an Online Migration Task

Step 1 Log in to the DCS console.

Step 2 In the navigation pane, choose **Data Migration**.

Step 3 Click **Create Online Migration Task**.

Step 4 Enter the task name and description.

Step 5 Configure the VPC, subnet, and security group for the migration task.

The VPC, subnet, and security group facilitate the migration. Ensure that the migration resources can access the source and target Redis instances.

----End

Step 5: Configure the Online Migration Task

Step 1 On the **Online Migration** tab page, click **Configure** in the row containing the online migration task you just created.

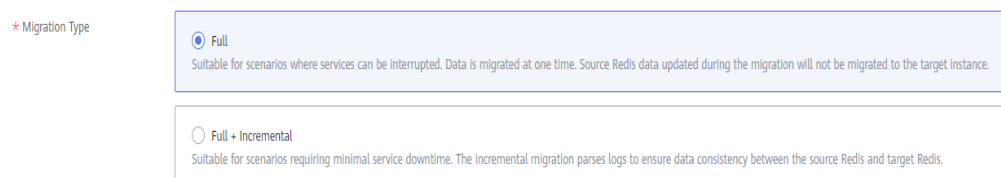
Step 2 Select a migration type.

Supported migration types are **Full** and **Full + Incremental**, which are described in [Table 6-1](#).

Table 6-1 Migration type description

Migration Type	Description
Full	Suitable for scenarios where services can be interrupted. Data is migrated at one time. Source instance data updated during the migration will not be migrated to the target instance.
Full + incremental	Suitable for scenarios requiring minimal service downtime. The incremental migration parses logs to ensure data consistency between the source and target instances. Once the migration starts, it remains Migrating until you click Stop in the Operation column. After the migration is stopped, data in the source instance will not be lost, but data will not be written to the target instance.

Figure 6-1 Selecting the migration type



Step 3 Configure source Redis and target Redis.

- Source Redis Type:** Select **Redis in the cloud** or **Self-hosted Redis** as required.
 - Redis in the cloud:** a HUAWEI CLOUD DCS Redis instance that is in the same VPC as the migration task
 - Self-hosted Redis:** self-hosted Redis on HUAWEI CLOUD, in another cloud, or in on-premises data centers. If you select this option, enter Redis addresses.
- If the instance is password-protected, you can click **Test Connection** to check whether the instance password is correct and whether the network is connected.

Step 4 For **Target Redis Instance**, select the DCS Redis Instance prepared in [Step 2: Prepare the Target DCS Redis Instance](#).

If the instance is password-protected, you can click **Test Connection** to check whether the instance password meets the requirements.

Step 5 Confirm the migration task details and click **Submit**.

Go back to the data migration task list. After the migration is successful, the task status changes to **Successful**.

 **NOTE**

Once incremental migration starts, it remains **Migrating** until you click **Stop**.

If the migration fails, click the migration task and check the log on the **Migration Logs** page.

----End

Verifying the Migration

After the migration is complete, use redis-cli to connect the source and target Redis instances to check data integrity.

1. Connect to the source Redis and the target Redis.
2. Run the **info keyspace** command to check the values of **keys** and **expires**.

```
192.168.1.217:6379> info keyspace
# Keyspace
db0:keys=81869,expires=0,avg_ttl=0
192.168.1.217:6379>
```

3. Calculate the difference between the values of **keys** and **expires** of the source Redis and the target Redis.

If the differences are the same, the data is complete and the migration is successful.

During full migration, source Redis data updated during the migration will not be migrated to the target instance.

6.2 Backup Migration from Another Cloud

Application Scenarios

Use the DCS console to migrate Redis data from Redis of another cloud or self-hosted Redis to HUAWEI CLOUD DCS for Redis.

Simply download the source Redis data and then upload the data to an OBS bucket in the same region as the target DCS Redis instance. After you have created a migration task on the DCS console, DCS will read data from the OBS bucket and data will be migrated to the target instance.

.aof, .rbb, .zip, and .tar.gz files can be uploaded to OBS buckets. You can directly upload .aof and .rdb files or compress them into .zip or .tar.gz files before uploading.

Prerequisites

- The OBS bucket must be in the same region as the target DCS Redis instance. For example, they both reside in the CN North-Beijing1 region.
- The data files to be uploaded must be in the .aof, .rdb, .zip, or .tar.gz format.
- To migrate data from a single-node or master/standby Redis instance of another cloud, create a backup task and download the backup file.
- To migrate data from a cluster Redis instance of another cloud, download all backup files, upload all of them to the OBS bucket, and select all of them for the migration. Each backup file contains data for a shard of the instance.
- .rdb backup files of self-hosted Redis 5.0 cannot be imported. .rdb backup files of self-hosted Redis 3.0 or 4.0 can be exported using redis-cli. .rdb files of other cloud Redis can be exported only by creating backup tasks, and cannot be exported by running commands in redis-cli.
- Redis Cluster instances only support .rdb files.

Step 1: Prepare the Target DCS Redis Instance

- If a target DCS Redis instance is not available, create one first. For details, see [Buying a DCS Redis Instance](#).
- If you already have a DCS Redis instance, you do not need to create one again, but you need to clear the instance data before the migration. For details, see [Clearing DCS Instance Data](#).

You can use a DCS Redis 3.0, 4.0, or 5.0 instance as the target instance.

Step 2: Create an OBS Bucket and Upload Backup Files

Step 1 Upload the backup data files to the OBS bucket by using OBS Browser+.

If the backup file to be uploaded is smaller than 5 GB, go to step [Step 2](#) to upload the file using the OBS console.

If the backup file to be uploaded is larger than 5 GB, follow the [instructions](#) provided by OBS.

Step 2 On the OBS console, upload the backup data files to the OBS bucket.

Perform the following steps if the backup files are smaller than 5 GB:

1. Create an OBS bucket.

When creating an OBS bucket, pay attention to the configuration of the following parameters. For details on how to set other parameters, see [Creating a Bucket](#) in *OBS User Guide*.

a. **Region:**

The OBS bucket must be in the same region as the target DCS Redis instance.

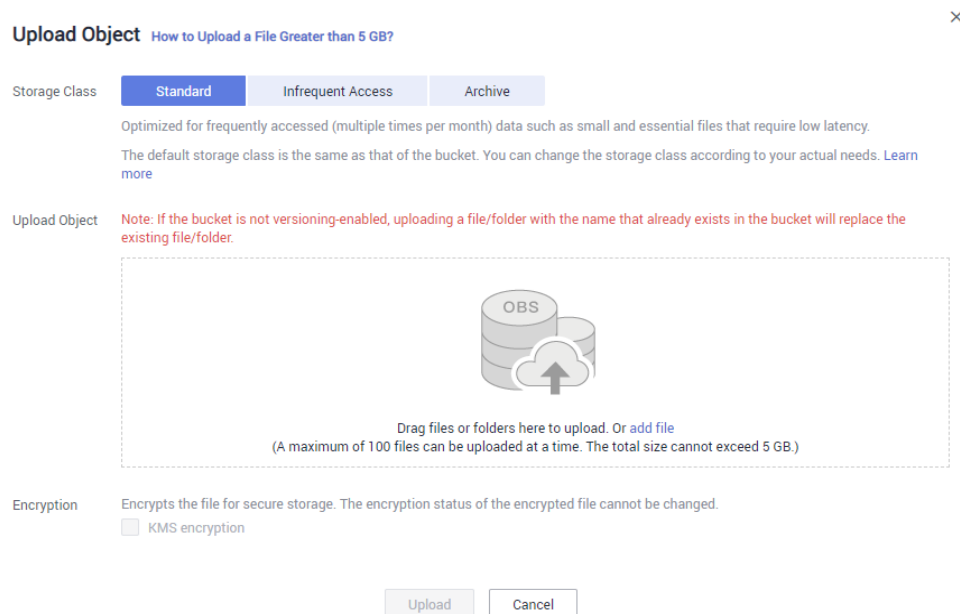
b. **Storage Class:** Available options are **Standard**, **Infrequent Access**, and **Archive**.

Do not select **Archive**. Otherwise, the migration will fail.

c. Click **Create Now**.

2. In the bucket list, click the bucket created in **Step 2.1**.
3. In the navigation pane, choose **Objects**.
4. On the **Objects** tab page, click **Upload Object**.
5. Specify **Storage Class**.
Do not select **Archive**. Otherwise, the migration will fail.
6. Upload the objects.
Drag files or folders to the **Upload Object** area or click **add file**.
A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.

Figure 6-2 Uploading objects in batches



7. (Optional) Select **KMS encryption** to encrypt the uploaded files.
8. Click **Upload**.

----End

Step 3: Create a Migration Task

- Step 1** Log in to the DCS console.
- Step 2** In the navigation pane, choose **Data Migration**.
- Step 3** Click **Create Backup Import Task**.
- Step 4** Enter the task name and description.
- Step 5** In the **Source Redis** area, select **OBS Bucket** for **Data Source** and then select the OBS bucket to which you have uploaded backup files.

In the **Backup Files** table, the files you have uploaded are displayed.

NOTE

You can upload files in the .aof, .rdb, .zip, or .tar.gz format.

Figure 6-3 Specifying the backup file information

Source Redis

* Data Source OBS bucket Redis

* OBS Bucket 05041ffa40025702f6dc009cc6f8f33-hilens-s... ▾ View Bucket

* Backup Files To import data from multiple backup files, you can create multiple migration tasks and run them at a time.

Name	Size
<input type="checkbox"/> testns.zip	9.46 KB

Step 6 Select the backup files whose data is to be migrated.

Step 7 Select the target Redis instance prepared in **Step 1: Prepare the Target DCS Redis Instance**. If the target Redis instance has a password, enter the password and test the connection to check whether the password is correct.

Step 8 Click **Next**.

Step 9 Confirm the migration task details and click **Submit**.

Go back to the data migration task list. After the migration is successful, the task status changes to **Successful**.

----End

6.3 Online Migration with Rump

Background

- Redis instances provided by some cloud service vendors do not allow **SLAVEOF**, **BGSAVE**, and **PSYNC** commands to be issued from Redis clients. As a result, redis-cli, redis-port, and other tools cannot be used to export data.
- Using the **KEYS** command may block Redis.
- Cloud service vendors usually only support downloading backup files. This method is suitable only for offline migration, featuring longer service interruption.

Rump is an open-source tool designed for migrating Redis data online. It supports migration between DBs of the same instance and between DBs of different instances.

Migration Principles

Rump uses the **SCAN** command to acquire keys and the **DUMP/RESTORE** command to get or set values.

Featuring time complexity $O(1)$, **SCAN** is capable of quickly getting all keys. **DUMP/RESTORE** is used to read/write values independent from the key type.

Rump brings the following benefits:

- The **SCAN** command replaces the **KEYS** command to avoid blocking Redis.

- Any type of data can be migrated.
- **SCAN** and **DUMP/RESTORE** operations are pipelined, improving the network efficiency during data migration.
- No temporary file is involved, saving disk space.
- Buffered channels are used to optimize performance of the source server.

NOTICE

1. To cluster DCS instances, you cannot use Rump. Instead, use `redis-port` or `redis-cli`.
 2. To prevent migration command resolution errors, do not include special characters (`#@:`) in the instance password.
 3. Stop the service before migrating data. If data is kept being written in during the migration, some keys might be lost.
-

Step 1: Installing Rump

1. Download [Rump \(release version\)](#).

On 64-bit Linux, run the following command:

```
wget https://github.com/stickermule/rump/releases/download/0.0.3/rump-0.0.3-linux-amd64;
```

2. After decompression, run the following commands to add the execution permission:

```
mv rump-0.0.3-linux-amd64 rump;  
chmod +x rump;
```

Step 2: Migrating Data

```
rump -from {source_redis_address} -to {target_redis_address}
```

Parameter/Option description:

- *{source_redis_address}*
Source Redis instance address, in the format of `redis://[user:password@]host:port/db`. **[user:password@]** is optional. If the instance is accessed in password-protected mode, you must specify the password in the RFC 3986 format. **user** can be omitted, but the colon (`:`) cannot be omitted. For example, the address may be `redis://:mypassword@192.168.0.45:6379/1`.
db is the sequence number of the database. If it is not specified, the default value is 0.
- *{target_redis_address}*
Address of the target Redis instance, in the same format as *{source_redis_address}*.
In the following example, data in DB0 of the source Redis is migrated to the target Redis whose connection address is 192.168.0.153. ********* stands for the password.

```
[root@ecs ~]# ./rump -from redis://127.0.0.1:6379/0 -to redis://*****@192.168.0.153:6379/0  
.Sync done.  
[root@ecs ~]#
```

6.4 Codis Migration from Another Cloud with redis-cli and redis-port

Codis data can be smoothly migrated to DCS Proxy Cluster instances because DCS Proxy Clusters are based on the open-source Codis. The following describes the procedure of migrating Codis data using redis-cli and redis-port.

- From self-hosted Codis to DCS Proxy Cluster:
Self-hosted Codis deployed in a local data center, HUAWEI CLOUD, or another public cloud is supported. There are no restrictions on exporting data from self-hosted Codis.
- From Codis of another cloud to DCS Proxy Cluster:
Some cloud vendors have restrictions on data export commands. You can export data by downloading backup files on their consoles.

Preparation

- Create a DCS Proxy Cluster instance.
The available memory of the instance must be larger than the size of the data to be imported.
- Prepare a server which will be used for transmitting and importing data.
The server must be connected to the DCS instance and have sufficient disk space for storing migration data.
You can use HUAWEI CLOUD ECS and configure the same VPC, subnet, and security group for the ECS and the DCS instance.

Procedure

Step 1 Export the RDB file.

Download and install Redis: <http://download.redis.io/releases/redis-5.0.8.tar.gz>.

Run the following command to export the RDB file for each shard in the cluster:

```
redis-cli -h {source_redis_address} -p {source_redis_port} -a {source_redis_password} --rdb {output.rdb}
```

If "Transfer finished with success." is displayed after the command is executed, the file is exported successfully.

NOTE

1. Migrate data during off-peak hours.
2. When exporting Codis data, individually export the data of each shard in the cluster, and then import the data file by file.

Step 2 Upload the RDB file to the server. HUAWEI CLOUD ECS is used as an example.

If the files are larger, compress them before transmission.

Step 3 Download redis-port.

Log in to the ECS. Run the following commands to download and decompress the tool package. redis-port can be used directly without compilation.

```
wget https://github.com/CodisLabs/redis-port/releases/download/v2.0-beta/redis-port-v2.0-beta-go1.10.1-linux.tar.gz
```

```
tar -xvf redis-port-v2.0-beta-go1.10.1-linux.tar.gz
```

Step 4 Import data.

Run the following commands in redis-port to import all RDB files to the target DCS instance. If the ECS and the DCS instance are in the same VPC, it takes 4 to 10 seconds to export 1 million data records, assuming that each record contains 20 byte data.

```
redis-restore -n {N} -i {outputfile.rdb} -t {password}@{dcs_instance_address}:{port} [--unixtime-in-milliseconds="yyyy-MM-dd hh:mm:ss"]
```

By specifying the **db** parameter, you can import the cached data of the specified DB in the file. **-n** indicates that multiple CPUs process the import task concurrently.

Parameter description:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-restore --help
Usage:
  redis-restore [--ncpu=N] [--input=INPUT|INPUT] --target=TARGET [--aof=FILE] [--db=DB] [--unixtime-in-milliseconds=EXPR]
  redis-restore --version

Options:
  -n N, --ncpu=N           Set runtime.GOMAXPROCS to N.
  -i INPUT, --input=INPUT  Set input rdb encoded file.
  -t TARGET, --target=TARGET The target redis instance ([auth@]host:port).
  -a FILE, --aof=FILE      Also restore the replication backlog.
  --db=DB                  Accept db = DB, default is *.
  --unixtime-in-milliseconds=EXPR Update expire time when restoring objects from RDB.

Examples:
  $ redis-restore dump.rdb -t 127.0.0.1:6379
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --aof dump.aof --db=1
  $ redis-restore -t 127.0.0.1:6379 --aof dump.aof
  $ redis-restore -t 127.0.0.1:6379 --db=0
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="@209059200000" //
  ttlms += (now - '1976-08-17')
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="+1000" // ttlms +=
  1s
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="-1000" // ttlms -= 1s
  $ redis-restore -i dump.rdb -t 127.0.0.1:6379 --unixtime-in-milliseconds="1976-08-17 00:00:00" // ttlms
  += (now - '1976-08-17')
```

Example:

```
root@redis-nodelete:~/port/redis-port-v2.0-beta-go1.10.1-linux# ./redis-restore -i save196.rdb -t
*****@192.168.0.171:6379
2018/03/26 09:15:33 restore.go:70: [INFO] restore: input = "save196.rdb", aoflog = "" target =
*****@192.168.0.171:6379"
2018/03/26 09:15:33 restore.go:126: [INFO] restore: (r,f,s/a,f,s) = (rdb,rdb.forward,rdb.skip/
aof,rdb.forward,rdb.skip)
2018/03/26 09:15:34 restore.go:155: [INFO] restore: size = 46721058 - [ 49.94%, 0.00%] (r,f,s/
a,f,s)=(23330816,0,599496/0,0,0) ~ (22.25mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:155: [INFO] restore: size = 46721058 - [ 99.31%, 0.00%] (r,f,s/
a,f,s)=(46399488,12558,1179884/0,0,0) ~ (44.25mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:155: [INFO] restore: size = 46721058 - [100.00%, 0.00%] (r,f,s/
```

```
a,f,s)=(46721058,20000,1179884/0,0,0) ~ (44.56mb,-,-/0,-,-)
2018/03/26 09:15:35 restore.go:159: [INFO] restore: done
root@redis-nodelete:~/port/redis-port-v2.0-alpha-go1.9.2-linux#
```

Step 5 Verify the migration.

After the migration is completed, access the DCS instance and run the **info** command to check whether the data has been successfully imported as required.

If the data import fails, run the **flushall** or **flushdb** command to clear the cached data in the instance, and import the data again.

----End

6.5 Redis Cluster Migration from Another Cloud with redis-shake

redis-shake is an open-source Redis migration tool that allows you to migrate Redis Cluster data online or by importing backup files. However, if the source Redis Cluster is deployed in another cloud, the **SYNC** and **PSYNC** commands may have been disabled by the cloud vendors. In this case, online migration is not supported and you can only migrate data by importing backup files.

The following describes how to use **redis-shake** for backup migration to a DCS Redis Cluster instance.

Importing Backup Files

If the source Redis and the destination Redis cannot be connected, or the source Redis is deployed on other clouds, you can migrate data by importing backup files.

1. Create a Redis Cluster instance on the DCS console.
The memory of this instance cannot be smaller than that of the source Redis.
2. Run the following command to obtain the IP addresses and port numbers of all masters of the source Redis and target Redis:

```
redis-cli -h {redis_address} -p {redis_port} -a {redis_password} cluster nodes
```

In the command output similar to the following, obtain the IP addresses and ports of all masters.

```
[root@ecs-54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a 23 cluster nodes
fb75f0743af4695a3d241ff7790b2f508e4985ff 192.168.0.140:6379@16379 myself,master - 0 1562144170000 3 connected
d112bae791b2bbd9602fe32963536b8a0db9eb79 192.168.0.61:6379@16379 master - 0 1562144171524 1 connected 0-5460
73e2f8fe196166f9ad1283361867d24c136413f0 192.168.0.194:6379@16379 master - 0 1562144170000 2 connected 5461-10
40d72299fde6045de0f79ee4b97910b505acbc6a 192.168.0.231:6379@16379 slave 73e2f8fe196166f9ad1283361867d24c136413
b6e6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985f
c16b9acaee7dd0721f129596cd43bd499c0e396 192.168.0.169:6379@16379 slave d112bae791b2bbd9602fe32963536b8a0db9eb
```

 **NOTE**

After Redis is installed, it runs with redis-cli. To install Redis on CentOS, run the **yum install redis** command.

3. Prepare a cloud server and install redis-shake.
redis-shake must be able to access the target Redis and bound to an EIP.
You can use HUAWEI CLOUD ECS and configure the same VPC, subnet, and security group for the ECS and the DCS instance.

Download and decompress the release version of redis-shake.

```
[root@ecs-p-4-centos redisshake]# ll
total 16972
-rw-r--r-- 1 1320024 users 2749 Jun 24 16:15 ChangeLog
-rwxr-xr-x 1 1320024 users 14225 Jun 24 16:14 hypervisor
-rwxr-xr-x 1 1320024 users 13000971 Jun 24 16:14 redis-shake
-rw-r--r-- 1 1320024 users 8875 Jun 24 16:15 redis-shake.conf
-rw-r--r-- 1 root root 4326892 Jun 24 16:17 redis-shake.tar.gz
-rwxr-xr-x 1 1320024 users 458 Jun 24 16:14 start.sh
-rwxr-xr-x 1 1320024 users 374 Jun 24 16:14 stop.sh
```

 **NOTE**

If the source Redis is deployed in the data center intranet, install redis-shake on the intranet server. Export data and then upload the data to the cloud server as instructed by the following steps

4. Export the RDB file.

- Edit the **redis-shake.conf** file by providing the following information about all the masters of both the source and the target:

```
source.type = cluster
# If there is no password, skip the following parameter.
source.password_raw = {source_redis_password}
# IP addresses and port numbers of all masters of the source Redis Cluster, which are separated by semicolons (;).
source.address = {master1_ip}:{master1_port};{master2_ip}:{master2_port}...{masterN_ip}:{masterN_port}
```

- Run the following command to export the RDB file:

```
./redis-shake -type dump -conf redis-shake.conf
```

If the following information is displayed in the execution log, the backup file is exported successfully:

```
execute runner[*run.CmdDump] finished!
```

5. Import the RDB file.

- Import the RDB file (or files) to the cloud server. The cloud server must be connected to the target DCS instance.
- Edit the redis-shake configuration file.

Edit the **redis-shake.conf** file by providing the following information about all the masters of both the source and the target:

```
target.type = cluster
# If there is no password, skip the following parameter.
target.password_raw = {target_redis_password}
# IP addresses and port numbers of all masters of the target instance, which are separated by semicolons (;).
target.address = {master1_ip}:{master1_port};{master2_ip}:{master2_port}...{masterN_ip}:{masterN_port}
# List the RDB files to be imported, separated by semicolons (;).
rdb.input = local_dump.0;local_dump.1;local_dump.2;local_dump.3
```

Save and exit.

- Run the following command to import the RDB file to the target instance:

```
./redis-shake -type restore -conf redis-shake.conf
```

If the following information is displayed in the execution log, the backup file is imported successfully:

```
Enabled http stats, set status (incr), and wait forever.
```

6. Verify the migration.

After data synchronization, access the DCS Redis Cluster instance using `redis-cli`. Run the **info** command to query the number of keys in the **Keyspace** section to confirm that data has been fully imported.

If the data has not been fully imported, run the **flushall** or **flushdb** command to clear the cached data in the instance, and synchronize data again.

7 Migrating Data from DCS to Self-Hosted Redis

Scenario

You can use the online migration function of the DCS console to migrate HUAWEI CLOUD DCS Redis instances to your self-hosted Redis. You can also export the DCS instance data to an RDB file and import it to local or self-hosted Redis.

Recommended Solutions

- Online migration on the DCS console
For details, see [4.1 Online Migration of Self-Hosted Redis](#). Select **Self-hosted Redis** and enter the target Redis address when configuring the target Redis.
- Use redis-cli or the DCS console to export the DCS instance data to an RDB file, and then use redis-port to import the file to the target.
For details on how to install and use redis-port, see [4.5 Self-Hosted Redis Migration with redis-port \(RDB\)](#).
- Rump
This tool is recommended for online migration if possible. For details, see [6.3 Online Migration with Rump](#).

8 FAQs

8.1 How Do I Migrate Memcached Data?

Memcached does not provide commands for traversal data query. Therefore, you cannot directly export data from your Memcached and migrate the data to a DCS Memcached instance.

Record cache keys through logging of your application systems, extract key-value data, and write the data to a DCS Memcached instance, achieving gradual data migration.

NOTE

By using some open-source tools, you can run the **stats cachedump** command of Memcached and perform get operations to query partial key-value data stored in your Memcached. However, you can only query key data not greater than 2 MB (including the size of all keys queried and additional information with a size of more than 20 bytes for each key) by using this command. Therefore, you cannot use such tools or similar methods for data migration.

8.2 What Are Redis Version Requirements for Data Migration?

DCS provides Redis 3.0, 4.0, and 5.0 instances.

If the source Redis instance version is later than the target Redis instance version, analyze the cache commands used by your service systems and verify the commands one by one during the rehearsal phase.

8.3 What Should I Consider When Transferring or Operating Data Between Different OSs?

Convert the format of a data file before importing the file.

Run the following command to convert the format of a file in the Windows OS to that in the Unix-like OS:

dos2unix *{filename}*

Run the following command to convert the format of a file in the Unix-like OS to that in the Windows OS:

unix2dos *{filename}*

8.4 Can I Migrate Data from a Multi-DB Source Redis Instance to a Cluster DCS Redis Instance?

A total of 256 DBs (DB 0 to DB 255) can be configured for a single-node or master/standby DCS instance.

Only one DB is configured for a cluster DCS instance.

Solutions:

1. Combine different DBs in the source Redis instance into one DB.
2. Apply for multiple DCS instances.

After the migration, the instance connection address and DB IDs change. In this case, modify the configurations for your services.

8.5 How Can I Migrate Partial Data?

The online migration function provided on the console does not support migration of specified DBs. If you want to migrate **specified Redis DBs**, use `redis-port` to export or import the specified DBs.

By using `redis-port`, you can run the **restore** command to import a specified DB in an RDB file.

```
redis-restore -n {N} -i {dump.rdb} -t {password}@{dcs_instance_address}:{port} [--unixtime-in-milliseconds="yyyy-MM-dd hh:mm:ss"] [--db={DB_number}]
```

For details, see [Migrating Data with redis-port](#).

To migrate **specified data**, develop a script to obtain the specified keys and data and then import the data to a DCS instance.

8.6 What Are the Constraints and Precautions for Migrating Redis Data to a Cluster Instance?

- Proxy Cluster instances

Proxy Cluster instances are used in the same way that you use single-node or master/standby instances. However, only one DB is configured for a Proxy Cluster instance, and the **SELECT** command is not supported. When data files are imported in batches, an error message will be displayed and ignored if the **SELECT** command exists. Then, the remaining data will continue to be imported.

Example:

DB 0 and DB 2 in the source Redis instance contain data, and the generated AOF or RDB file contains these two DBs.

When the source Redis data is imported into a Proxy Cluster DCS instance, the **SELECT 2** command will be ignored, and then data in DB 2 in the source Redis instance will be imported.

Note that:

- If different DBs in the source Redis instance contain the same keys, values of keys in the DB with the largest ID will overwrite those in the other DBs.
- If the source Redis instance contains multiple DBs, data is stored in the same DB after being migrated to a cluster DCS instance, and the **SELECT** command is not supported. In this case, modify the configurations for your services.

- Redis Cluster instances

Only one DB is configured for a Redis Cluster instance. Data is migrated to a Redis Cluster instance in a different way from other types of instances. Nodes in the shards of a Redis Cluster must be connected separately through clients. Data is imported to the nodes separately. Run the following command to query the IP addresses of the cluster nodes:

```
redis-cli -h {Redis Cluster IP} -p 6379 -a {password} cluster nodes
```

In the returned list of IP addresses, record the ones marked by "master".

8.7 What Should I Consider for Online Migration?

- Network
Before online migration, ensure that the network configured for the migration task is connected to source and target Redis instances.
- Tool
Use the online migration function provided on the DCS console.
- Data integrity
If you suspend your services for data migration, check the data volume and main keys after the migration.
If you do not suspend your services, migrate data incrementally.

8.8 Can I Perform Online Migration Without Any Service Interruption?

Yes. You can use the application dual-write mode. In this mode, during data migration, data is still read from the source Redis instance, and operations such as adding, deleting, and modifying data are also performed on the DCS Redis instance.

After maintaining the preceding mode for a period of time (waiting for a large amount of data to be deleted after expiration), migrate the cached data from your service systems to DCS. If service system migration to HUAWEI CLOUD is also involved, deploy your service systems before migrating your cached data to HUAWEI CLOUD.

This mode is not recommended for the following reasons:

1. Stable and quick network access cannot be ensured. If the source Redis instance is not deployed on HUAWEI CLOUD, access DCS over a public network, which is inefficient.
2. Modify the code to implement concurrent writing of two sets of data.
3. The data eviction policy varies depending on the source Redis instance. It may take a long time to complete data migration and it is difficult to ensure data integrity.

8.9 Can I Migrate Data Between DCS Memcached and Redis Instances?

No. Memcached and Redis are different cache databases and do not support data migration between each other.