Distributed Cache Service

Data Migration Guide

Issue 01

Date 2024-04-01

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property 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 Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Overview

This guide provides suggestions and instructions on how to migrate Redis data. For details on how to migrate Memcached data, see **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.

When migrating data, analyze the cache commands (reference: **Command Compatibility**) used by your service systems and verify the commands one by one during the rehearsal phase. If necessary, contact technical support.

NOTICE

- Currently, the data migration function is free of charge in the OBT. You will be notified when data migration starts to be charged.
- 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.
- DCS for Redis 3.0 is no longer provided. You can use DCS for Redis 4.0 or later.

DCS Data Migration Modes

NOTE

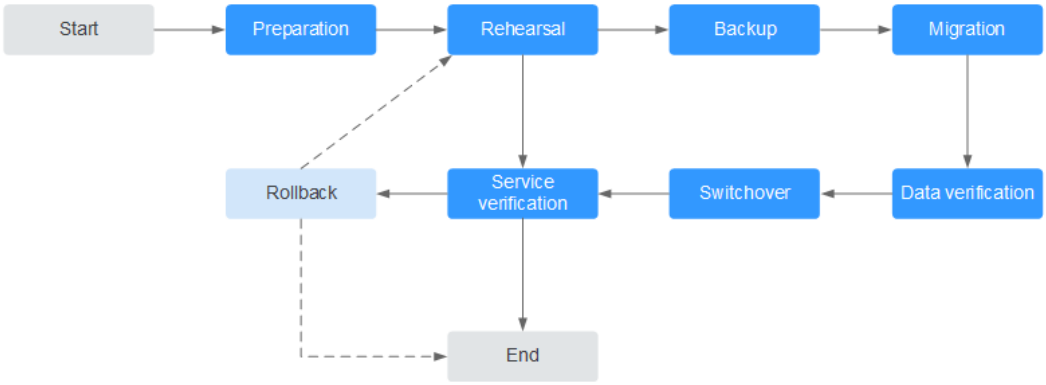
- **DCS for Redis** refers to Redis instances provided by DCS.
- **Self-hosted Redis** refers to self-hosted Redis on the cloud, from other cloud vendors, or in on-premises data centers.
- **Other cloud Redis** refers to Redis services provided by other cloud vendors.
- √: Supported. ×: Not supported.

Table 1-1 DCS data migration modes

Migration Mode	Source	Target: DCS		
		Single-Node, Read/Write Splitting, or Master/Standby	Proxy Cluster	Redis Cluster
Importing backup files	AOF file	√	√	√
	RDB file	√	√	√
Migrating data online	DCS for Redis: Single-node, read/write splitting, or master/standby	√	√	√
	DCS for Redis: Proxy Cluster **NOTE** Proxy Cluster DCS Redis 3.0 instances cannot be used as the source, while Proxy Cluster DCS Redis 4.0 or 5.0 instances can.	√	√	√
	DCS for Redis: Redis Cluster	√	√	√
	Self-hosted Redis	√	√	√
	Other cloud Redis	√	√	√
	NOTE You can migrate data online in full or incrementally from **other cloud Redis** to **DCS for Redis** if they are connected and the **SYNC** and **PSYNC** commands can be run on the source Redis. However, some instances provided by other cloud vendors may fail to be migrated online. In this case, migrate data through backup import or use other migration schemes. For details, see **Migration Tools and Schemes**.			

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), security groups, and 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

   Create VPCs and subnets based on service planning.

3. Server resources

   Apply for 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

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

5. Related tools

   Install the FTP tool, SSH 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 keyspace** 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 by default.
	Number of keys in each DB	Used to verify the data integrity after migration.
	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.

Migration Source	Item	Description
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
	Whitelist or security group	DCS Redis 3.0, 4.0, 5.0, and 6.0 professional edition instances are deployed in different modes. Therefore, the access control methods vary. You can control access to your DCS instances by setting security groups or whitelists. For details, see **How Do I Configure a Security Group?** or **Managing IP Address Whitelist**.
...	...	*Other configurations*.

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.	• 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	• 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.	-
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 **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	• If the network between your self-hosted Redis instance and the DCS Redis instance is connected, follow to the instructions in **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 **Backup Migration of Self-Hosted Redis**.	-
	redis-cli	**Self-Hosted Redis Migration with redis-cli (AOF)**	-
		Self-Hosted Redis Migration with redis-cli (RDB)	-
	redis-shake	**Self-Hosted Redis Cluster Migration with redis-shake**	-
Between DCS instances	DCS console	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. • If the network between the source and target DCS Redis instances is connected, follow to the instructions in **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 **Backup Migration Between Regions or Redis Versions**.	**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.

Scenario	Tool	Use Case	Description
		Migrate Redis data between regions. For details, see **Backup Migration Between Regions or Redis Versions**.	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.
		Migrate Redis data from one account to another. • For details, see **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 **Online Migration Between DCS Redis Instances**.	-
From another cloud to DCS	DCS console	• If the **SYNC** and **PSYNC** commands are not disabled for the Redis service provided by another cloud, follow the instructions in **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 **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.

Scenario	Tool	Use Case	Description
	Rump	**Online Migration with Rump**	-
	redis-shake	**Offline Migration of Redis Cluster from Another Cloud with redis-shake**	-
		Online Full Migration of Redis from Another Cloud with redis-shake	-
From DCS to self-hosted Redis	DCS console	**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 DCS Redis instance for online migration, the **SYNC** command is automatically enabled.
- You cannot use public networks for online migration.
- During online migration, you are advised to set **repl-timeout** on the source instance to 300s and **client-output-buffer-limit** to 20% of the maximum memory of the instance.
- The source must be Redis 3.0 or later.

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.

- By default, a Proxy Cluster instance has only one database (DB0). Before you migrate data from a multi-DB single-node or master/standby instance to a Proxy Cluster instance, check whether any data exists on databases other than DB0. If yes, enable multi-DB for the Proxy Cluster instance by referring to **Enabling Multi-DB**.
- By default, a Redis Cluster instance has only one DB (DB0). Before you migrate data from a multi-DB single-node or master/standby instance to a Redis Cluster instance, check whether any data exists on databases other than DB0. To ensure that the migration succeeds, move all data to DB0 by referring to **Online Migration with Rump**.

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 4: Create an 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 4: Create an Online Migration Task**. 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 DCS 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.

NOTE

- The online migration task uses a tenant IP address (**Migration ECS** displayed on the **Basic Information** page of the task.) If a whitelist is configured for the source or target instance, add the migration IP address to the whitelist or disable the whitelist.
- To allow the VM used by the migration task to access the source and target instances, set an outbound rule for the task's security group to allow traffic through the IP addresses and ports of the source and target instances. By default, all outbound traffic is allowed.

----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.

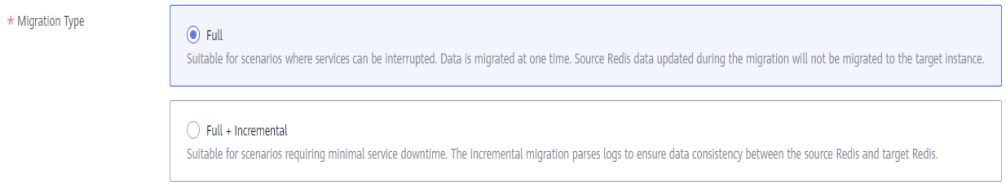
Step 2 Select a migration type.

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. When the transmission network is stable, the delay of incremental migration is within seconds. The actual delay depends on the transmission quality of the network link.

Figure 4-1 Selecting the migration type



Step 3 Configure source Redis and target Redis.

1. The Redis type can be **Redis in the cloud** or **Self-hosted Redis** as required.
   - **Redis in the cloud**: a DCS Redis instance (source or target) that is in the same VPC as the migration task. If you select this option, specify a DCS Redis instance.
   - **Self-hosted Redis**: a DCS Redis instance, Redis in another cloud, or self-hosted Redis. If you select this option, enter Redis addresses.

   NOTE

   If the source and target Redis instances in different regions of Huawei Cloud are connected, simply select **Self-hosted Redis** for **Target Redis Type** and enter the instance addresses, regardless of whether the target Redis instance is self-hosted or in the cloud.
2. If the instance is password-protected, click **Test Connection** to check whether the instance password is correct and whether the network is connected. If the instance is not password-protected, click **Test Connection** directly.
3. You can specify the source DB and target DB. For example, if you enter **5** for source DB and **6** for target DB, data in DB5 of the source Redis will be migrated to DB6 of the target Redis. If the source DB is not specified but the target DB is specified, all source data will be migrated to the specified target DB by default. If the target DB is not specified, data will be migrated to the corresponding target DB.

   NOTE

   - If the source Redis is multi-DB and the target is single-DB (DB0), either ensure that all source data is in DB0, or specify a source DB and set the target DB to **0**. Otherwise, migration will fail.
   - For details about DB in DCS for Redis, see **Does DCS for Redis Support Multi-DB?**

Step 4 Click **Next**.

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**.
- To stop a migration task, select the check box on the left of the migration task and click **Stop** above the instance list.
- After data migration, duplicate keys will be overwritten.

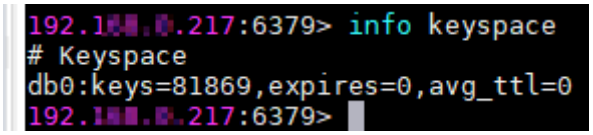
If the migration fails, click the migration task and check the log on the **Migration Logs** page.

----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**.



3. Calculate the differences 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 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, .rdb, .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.
- 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.

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**.

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.

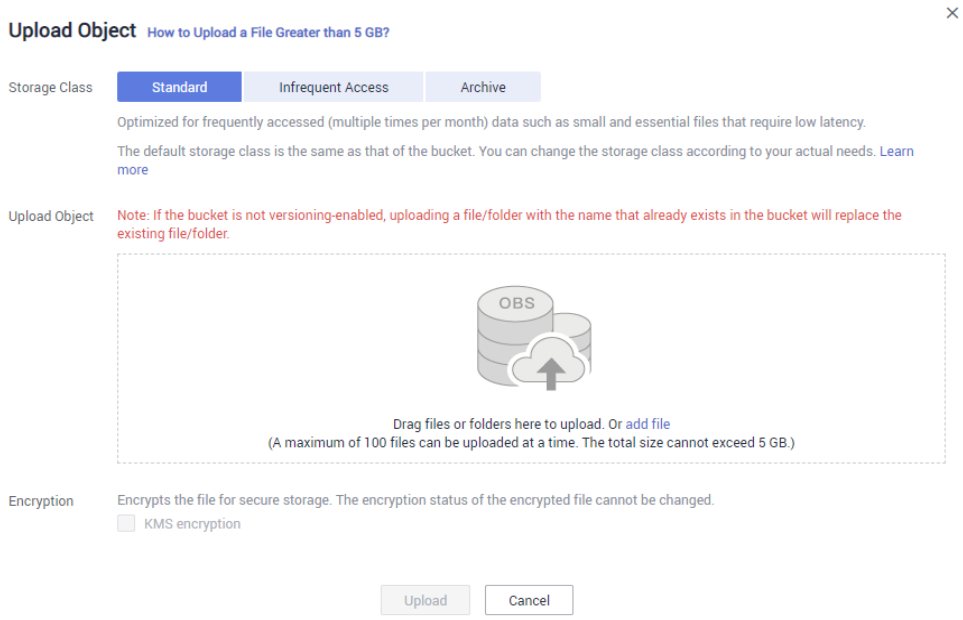
   c. Click **Create Now**.
2. In the bucket list, click the bucket created in **Step 2.1**.
3. In the navigation pane, choose **Objects**.
4. On the **Objects** tab page, click **Upload Object**.
5. Specify **Storage Class**.

   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.

Step 6 You can specify **Source DB** to migrate data from the specified DB in the source backup file. For example, if you enter **5**, only data in DB5 will be migrated. To migrate all databases, do not specify **Source DB**.

Step 7 Enable **Multi-DB Proxy Cluster** if the source Redis is a multi-DB (**multi-db** set to **yes**) Proxy Cluster DCS Redis instance.

Step 8 Click **Add Backup** and select the backup files to be migrated.

Step 9 In the **Target Redis** area, select the **Target Redis Instance** prepared in **Step 1: Prepare the Target DCS Redis Instance**.

Step 10 If the target Redis instance has a password, enter the password and click **Test Connection** to check whether the password is correct. If the instance is not password-protected, click **Test Connection** directly.

Step 11 For **Target DB**, you can specify a DB in the target Redis to migrate data to. For example, if you enter **5**, data will be migrated to DB5 of the target Redis. If you do not specify a DB, data will be migrated to a DB corresponding to the source DB.

NOTE

- If the source Redis is multi-DB and the target is single-DB (DB0), either ensure that all source data is in DB0, or specify a source DB and set the target DB to **0**. Otherwise, migration will fail.
- For details about DB in DCS for Redis, see **Does DCS for Redis Support Multi-DB?**

Step 12 Click **Next**.

Step 13 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 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 about how to configure a security group, see **Security Group Configurations**.

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-shake) 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 4.0/5.0/6.0 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 about how to configure a security group, see **Security Group Configurations**.

Step 4: Importing Data

Use redis-shake 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 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. (The following uses v2.1.2 as an example. You can also use **other redis-shake versions**.)

   ```
   [root@ecs-p[redacted]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 [redacted].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-[redacted]54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a [redacted]3 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 73e2f8fe196166f9ad1283361867d24c13641
   be6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985f
   c16b9acaeed7dd0721f129596cd43bd499c0e396 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 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}
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

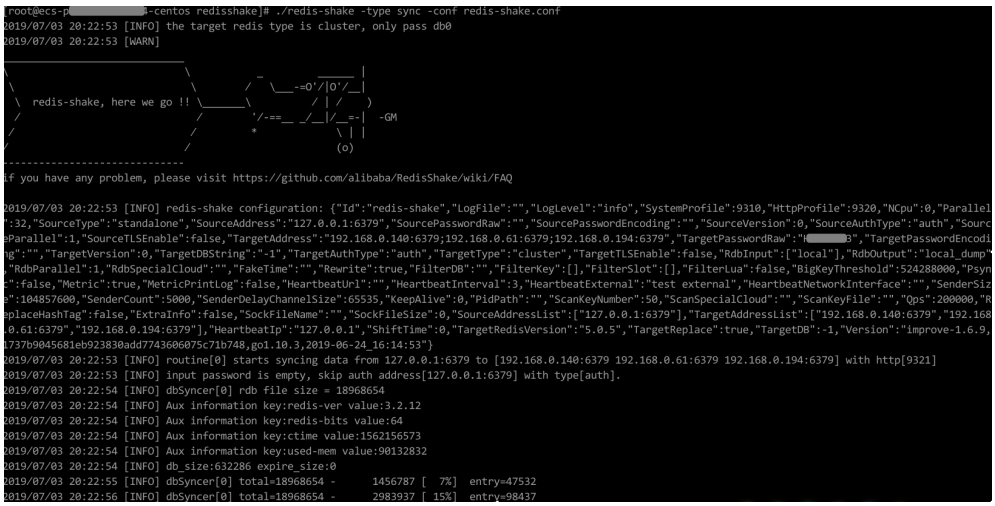
   If the following information is displayed, the full synchronization has been completed and incremental synchronization begins.

   ```
   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 4-3 Online migration using redis-shake

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

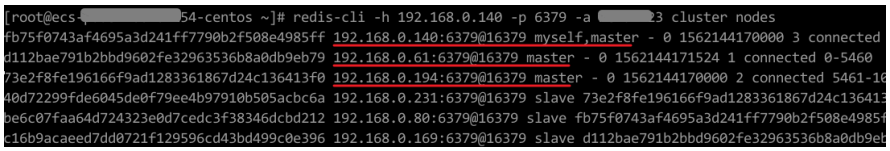
1. Create a Redis Cluster instance on the DCS console.

   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.

   NOTE

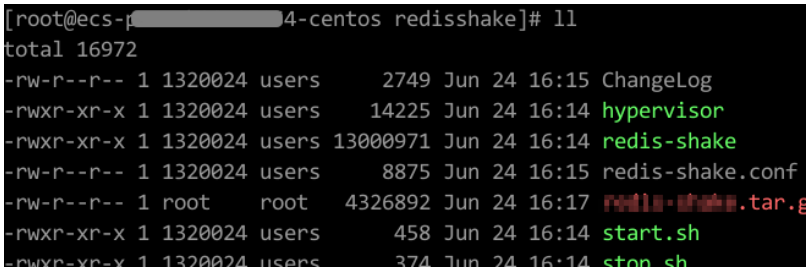
   After Redis is installed, it runs with redis-cli. To install Redis on CentOS, run the **yum install redis** command.

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. (The following uses v2.1.2 as an example.)

   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.
   a. Import the RDB file (or files) to the cloud server. The cloud server must be connected to the target DCS instance.
   b. 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.
   c. 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 DCS Redis instance for online migration, the **SYNC** command is automatically enabled.
- You cannot use public networks for online migration.
- During online migration, you are advised to set **repl-timeout** on the source instance to 300s and **client-output-buffer-limit** to 20% of the maximum memory of the instance.
- The source must be Redis 3.0 or later.

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.
- By default, a Proxy Cluster instance has only one database (DB0). Before you migrate data from a multi-DB single-node or master/standby instance to a

Proxy Cluster instance, check whether any data exists on databases other than DB0. If yes, enable multi-DB for the Proxy Cluster instance by referring to **Enabling Multi-DB**.

- By default, a Redis Cluster instance has only one DB (DB0). Before you migrate data from a multi-DB single-node or master/standby instance to a Redis Cluster instance, check whether any data exists on databases other than DB0. To ensure that the migration succeeds, move all data to DB0 by referring to **Online Migration with Rump**.

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 4: Create an 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 4: Create an Online Migration Task**. 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 DCS 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.

NOTE

- The online migration task uses a tenant IP address (**Migration ECS** displayed on the **Basic Information** page of the task.) If a whitelist is configured for the source or target instance, add the migration IP address to the whitelist or disable the whitelist.
- To allow the VM used by the migration task to access the source and target instances, set an outbound rule for the task's security group to allow traffic through the IP addresses and ports of the source and target instances. By default, all outbound traffic is allowed.

----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.

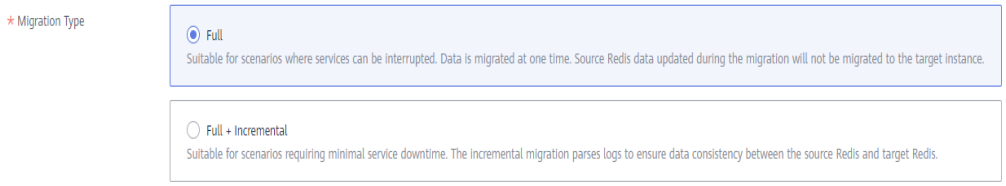
Step 2 Select a migration type.

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. When the transmission network is stable, the delay of incremental migration is within seconds. The actual delay depends on the transmission quality of the network link.

Figure 5-1 Selecting the migration type



Step 3 Configure source Redis and target Redis.

1. The Redis type can be **Redis in the cloud** or **Self-hosted Redis** as required.
   - **Redis in the cloud**: a DCS Redis instance (source or target) that is in the same VPC as the migration task. If you select this option, specify a DCS Redis instance.
   - **Self-hosted Redis**: a DCS Redis instance, Redis in another cloud, or self-hosted Redis. If you select this option, enter Redis addresses.

   NOTE

   If the source and target Redis instances in different regions of Huawei Cloud are connected, simply select **Self-hosted Redis** for **Target Redis Type** and enter the instance addresses, regardless of whether the target Redis instance is self-hosted or in the cloud.
2. If the instance is password-protected, click **Test Connection** to check whether the instance password is correct and whether the network is connected. If the instance is not password-protected, click **Test Connection** directly.
3. You can specify the source DB and target DB. For example, if you enter **5** for source DB and **6** for target DB, data in DB5 of the source Redis will be migrated to DB6 of the target Redis. If the source DB is not specified but the target DB is specified, all source data will be migrated to the specified target DB by default. If the target DB is not specified, data will be migrated to the corresponding target DB.

   NOTE

   - If the source Redis is multi-DB and the target is single-DB (DB0), either ensure that all source data is in DB0, or specify a source DB and set the target DB to **0**. Otherwise, migration will fail.
   - For details about DB in DCS for Redis, see **Does DCS for Redis Support Multi-DB?**

Step 4 Click **Next**.

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**.
- To stop a migration task, select the check box on the left of the migration task and click **Stop** above the instance list.
- After data migration, duplicate keys will be overwritten.

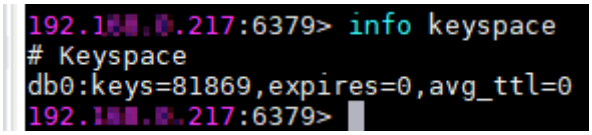
If the migration fails, click the migration task and check the log on the **Migration Logs** page.

----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**.



3. Calculate the differences 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 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, .rdb, .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.
- 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.

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**.

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**.

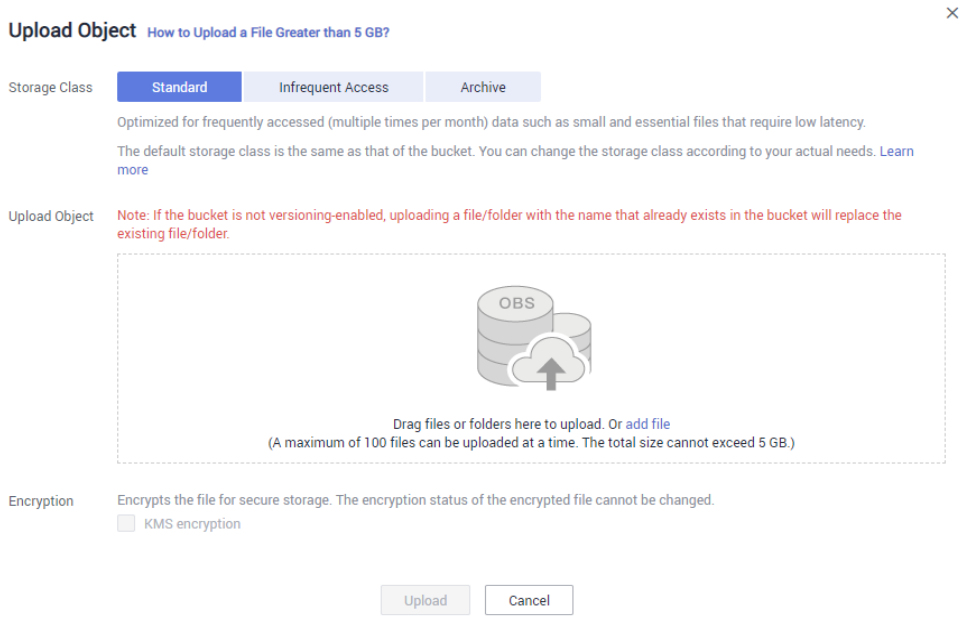
   Do not select **Archive**. Otherwise, the migration will fail.

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.

Step 6 You can specify **Source DB** to migrate data from the specified DB in the source backup file. For example, if you enter **5**, only data in DB5 will be migrated. To migrate all databases, do not specify **Source DB**.

Step 7 Enable **Multi-DB Proxy Cluster** if the source Redis is a multi-DB (**multi-db** set to **yes**) Proxy Cluster DCS Redis instance.

Step 8 Click **Add Backup** and select the backup files to be migrated.

Step 9 In the **Target Redis** area, select the **Target Redis Instance** prepared in **Step 1: Prepare the Target DCS Redis Instance**.

Step 10 If the target Redis instance has a password, enter the password and click **Test Connection** to check whether the password is correct. If the instance is not password-protected, click **Test Connection** directly.

Step 11 For **Target DB**, you can specify a DB in the target Redis to migrate data to. For example, if you enter **5**, data will be migrated to DB5 of the target Redis. If you do not specify a DB, data will be migrated to a DB corresponding to the source DB.

NOTE

- If the source Redis is multi-DB and the target is single-DB (DB0), either ensure that all source data is in DB0, or specify a source DB and set the target DB to **0**. Otherwise, migration will fail.
- For details about DB in DCS for Redis, see **Does DCS for Redis Support Multi-DB?**

Step 12 Click **Next**.

Step 13 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 DCS Redis instance for online migration, the **SYNC** command is automatically enabled.
- You cannot use public networks for online migration.
- During online migration, you are advised to set **repl-timeout** on the source instance to 300s and **client-output-buffer-limit** to 20% of the maximum memory of the instance.
- The source must be Redis 3.0 or later.

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.

- By default, a Proxy Cluster instance has only one database (DB0). Before you migrate data from a multi-DB single-node or master/standby instance to a Proxy Cluster instance, check whether any data exists on databases other than DB0. If yes, enable multi-DB for the Proxy Cluster instance by referring to **Enabling Multi-DB**.
- By default, a Redis Cluster instance has only one DB (DB0). Before you migrate data from a multi-DB single-node or master/standby instance to a Redis Cluster instance, check whether any data exists on databases other than DB0. To ensure that the migration succeeds, move all data to DB0 by referring to **Online Migration with Rump**.

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 4: Create an 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 4: Create an Online Migration Task**. 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 DCS 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.

NOTE

- The online migration task uses a tenant IP address (**Migration ECS** displayed on the **Basic Information** page of the task.) If a whitelist is configured for the source or target instance, add the migration IP address to the whitelist or disable the whitelist.
- To allow the VM used by the migration task to access the source and target instances, set an outbound rule for the task's security group to allow traffic through the IP addresses and ports of the source and target instances. By default, all outbound traffic is allowed.

----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.

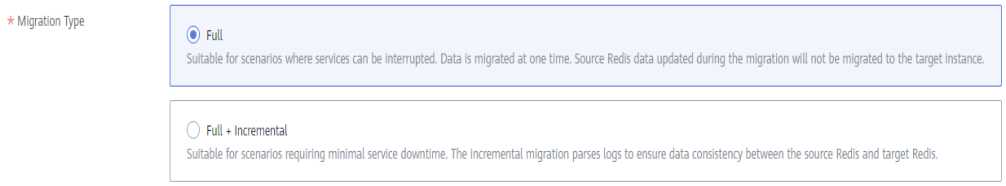
Step 2 Select a migration type.

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. When the transmission network is stable, the delay of incremental migration is within seconds. The actual delay depends on the transmission quality of the network link.

Figure 6-1 Selecting the migration type



Step 3 Configure source Redis and target Redis.

1. The Redis type can be **Redis in the cloud** or **Self-hosted Redis** as required.
   - **Redis in the cloud**: a DCS Redis instance (source or target) that is in the same VPC as the migration task. If you select this option, specify a DCS Redis instance.
   - **Self-hosted Redis**: a DCS Redis instance, Redis in another cloud, or self-hosted Redis. If you select this option, enter Redis addresses.

   NOTE

   If the source and target Redis instances in different regions of Huawei Cloud are connected, simply select **Self-hosted Redis** for **Target Redis Type** and enter the instance addresses, regardless of whether the target Redis instance is self-hosted or in the cloud.
2. If the instance is password-protected, click **Test Connection** to check whether the instance password is correct and whether the network is connected. If the instance is not password-protected, click **Test Connection** directly.
3. You can specify the source DB and target DB. For example, if you enter **5** for source DB and **6** for target DB, data in DB5 of the source Redis will be migrated to DB6 of the target Redis. If the source DB is not specified but the target DB is specified, all source data will be migrated to the specified target DB by default. If the target DB is not specified, data will be migrated to the corresponding target DB.

   NOTE

   - If the source Redis is multi-DB and the target is single-DB (DB0), either ensure that all source data is in DB0, or specify a source DB and set the target DB to **0**. Otherwise, migration will fail.
   - For details about DB in DCS for Redis, see **Does DCS for Redis Support Multi-DB?**

Step 4 Click **Next**.

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**.
- To stop a migration task, select the check box on the left of the migration task and click **Stop** above the instance list.
- After data migration, duplicate keys will be overwritten.

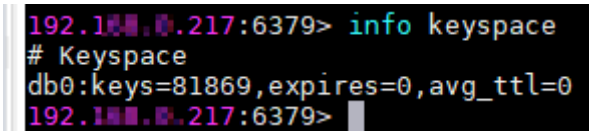
If the migration fails, click the migration task and check the log on the **Migration Logs** page.

----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**.



3. Calculate the differences 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 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, .rdb, .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.
- 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.

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**.

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**.

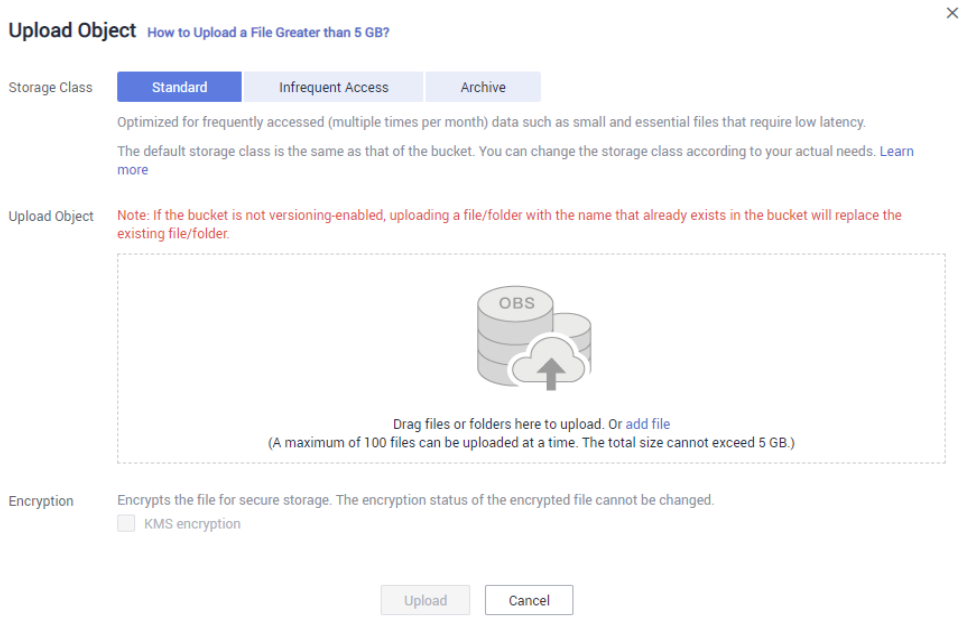
   Do not select **Archive**. Otherwise, the migration will fail.

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.

Step 6 You can specify **Source DB** to migrate data from the specified DB in the source backup file. For example, if you enter **5**, only data in DB5 will be migrated. To migrate all databases, do not specify **Source DB**.

Step 7 Enable **Multi-DB Proxy Cluster** if the source Redis is a multi-DB (**multi-db** set to **yes**) Proxy Cluster DCS Redis instance.

Step 8 Click **Add Backup** and select the backup files to be migrated.

Step 9 In the **Target Redis** area, select the **Target Redis Instance** prepared in **Step 1: Prepare the Target DCS Redis Instance**.

Step 10 If the target Redis instance has a password, enter the password and click **Test Connection** to check whether the password is correct. If the instance is not password-protected, click **Test Connection** directly.

Step 11 For **Target DB**, you can specify a DB in the target Redis to migrate data to. For example, if you enter **5**, data will be migrated to DB5 of the target Redis. If you do not specify a DB, data will be migrated to a DB corresponding to the source DB.

NOTE

- If the source Redis is multi-DB and the target is single-DB (DB0), either ensure that all source data is in DB0, or specify a source DB and set the target DB to **0**. Otherwise, migration will fail.
- For details about DB in DCS for Redis, see **Does DCS for Redis Support Multi-DB?**

Step 12 Click **Next**.

Step 13 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-shake, 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-shake 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 the source.

  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 Offline Migration of Redis Cluster from Another Cloud with redis-shake

redis-shake is an open-source tool for migrating data online or offline (by importing backup files) between Redis Clusters. If the source Redis Cluster is

deployed in another cloud, and online migration is not supported, you can 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 destitution 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-[redacted]54-centos ~]# redis-cli -h 192.168.0.140 -p 6379 -a [redacted]3 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 73e2f8fe196166f9ad1283361867d24c13641
   be6c07faa64d724323e0d7cedc3f38346dcbd212 192.168.0.80:6379@16379 slave fb75f0743af4695a3d241ff7790b2f508e4985f
   c16b9acaeed7dd0721f129596cd43bd499c0e396 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. (The following uses v2.1.2 as an example. You can also use **other redis-shake versions**.)

   ```
   [root@ecs-p[redacted]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 [redacted].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!
  ```

  NOTE

  If you cannot export the source Redis backup files using this method due to cloud vendors' restrictions on the **SYNC** and **PSYNC** commands, export the files on the source console or contact the source technical support.

5. Import the RDB file.

   a. Import the RDB file (or files) to the cloud server. The cloud server must be connected to the target DCS instance.

   b. 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.

   c. 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.

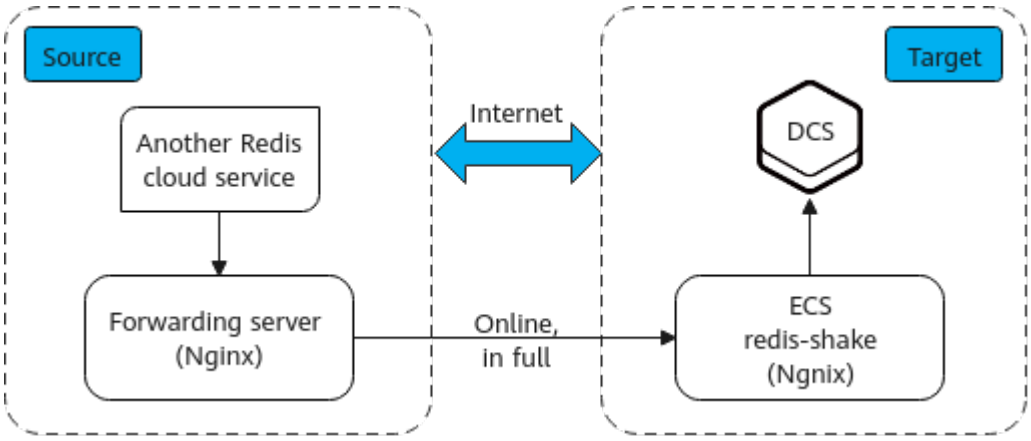
6.5 Online Full Migration of Redis from Another Cloud with redis-shake

redis-shake is an open-source Redis migration tool. Its **rump** mode allows you to obtain the full data of a source Redis using the **SCAN** command and write the data to a target Redis. This migration solution does not involve the **SYNC** or

PSYNC command and can be widely used for migration between self-built Redis and cloud Redis.

This section describes how to use the **rump** mode of redis-shake to migrate the full Redis data of another cloud service vendor at a time online to Huawei Cloud DCS.

Figure 6-3 Data flow in this solution



Prerequisites

- A **DCS Redis instance** has been created on Huawei Cloud.
- An **ECS** has been created on Huawei Cloud for running redis-shake.
- The ECS is in the same VPC as the DCS Redis instance and bound with an EIP.
- The **rump** mode does not support incremental data migration. To keep data consistency, stop writing data to the source Redis before migration.
- This solution applies only to same-database mapping and does not apply to inter-database mapping.
- If the source Redis has multiple databases (there are databases other than DB0), and your DCS instance is Proxy Cluster, multi-DB must be enabled for the DCS instance. Otherwise, the migration will fail. (Single-DB Proxy Cluster instances do not support the **SELECT** command.)
- If the source Redis has multiple databases (there are databases other than DB0), and your DCS instance is Redis Cluster, this solution cannot be used. (Redis Cluster DCS instances support only DB0.)

Procedure

Step 1 Install Nginx on the ECS and the source forwarding server. The following describes how to install Nginx on an ECS running CentOS 7.x. The commands vary depending on the OS.

1. Add Nginx to the Yum repository.
   ```
   sudo rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm
   ```
2. Check whether Nginx has been added successfully.
   ```
   yum search nginx
   ```
3. Install Nginx.
   ```
   sudo yum install -y nginx
   ```


4. Install the stream module.
```
yum install nginx-mod-stream --skip-broken
```
5. Start Nginx and set it to run automatically upon system startup.
```
sudo systemctl start nginx.service
sudo systemctl enable nginx.service
```
6. In the address box of a browser, enter the server address (the EIP of the ECS) to check whether Nginx is installed successfully.

   If the following page is displayed, Nginx has been installed successfully.

Step 2 Add the source forwarding server to the whitelist of the source Redis.

Step 3 Configure a security group for the source forwarding server.

1. Obtain the EIP of the Huawei Cloud ECS.
2. In the inbound rule of the security group of the source forwarding server, add the EIP of the Huawei Cloud ECS, and open the port that Huawei Cloud ECS's requests come through. The following takes port 6379 as an example.

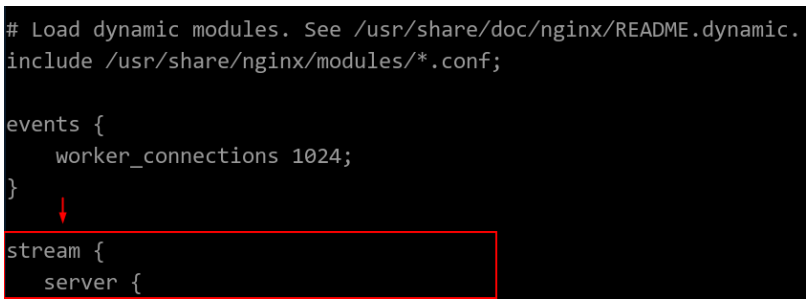
Step 4 Configure Nginx forwarding for the source forwarding server.

1. Log in to the Linux source forwarding server and run the following commands to open and modify the configuration file:
```
cd /etc/nginx
vi nginx.conf
```
2. Example forwarding configuration:
```
stream {
   server {
      listen 6379;
      proxy_pass {source_instance_address}:{port};
   }
  }
```
   6379 is the listening port of the source forwarding server. *{source_instance_address}* and *{port}* are the connection address and port of the source Redis instance.

   This configuration allows you to access the source Redis through the local listening port 6379 of the source forwarding server.

   This configuration must be added exactly where it is shown in the following figure.

Figure 6-4 Configuration location



3. Restart Nginx.
```
service nginx restart
```
4. Verify whether Nginx has been started.
```
netstat -an|grep 6379
```
If the port is being listened, Nginx has been started successfully.

Figure 6-5 Verification result

Step 5 Configure Nginx forwarding for the Huawei Cloud ECS.

1. Log in to the Linux ECS on Huawei Cloud and run the following commands to open and modify the configuration file:

cd /etc/nginx

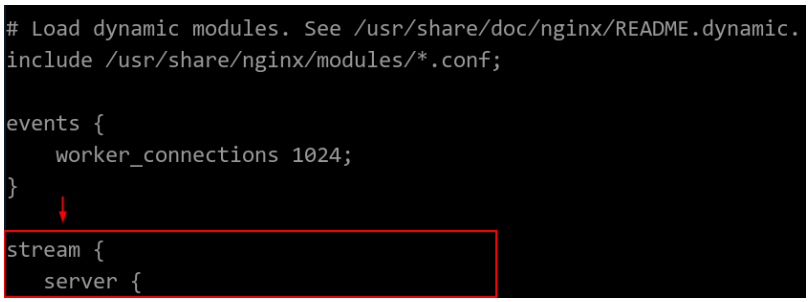
vi nginx.conf

2. Configuration example:
```
stream {
   server {
      listen 6666;
      proxy_pass {source_ecs_address}:6379;
   }
  }
```
6666 is Huawei Cloud ECS's listening port, *{source_ecs_address}* is the public IP address of the source forwarding server, and **6379** is the listening port of the source forwarding server Nginx.

This configuration allows you to access the source forwarding server through the local listening port 6666 of the Huawei Cloud ECS.

This configuration must be added exactly where it is shown in the following figure.

Figure 6-6 Configuration location



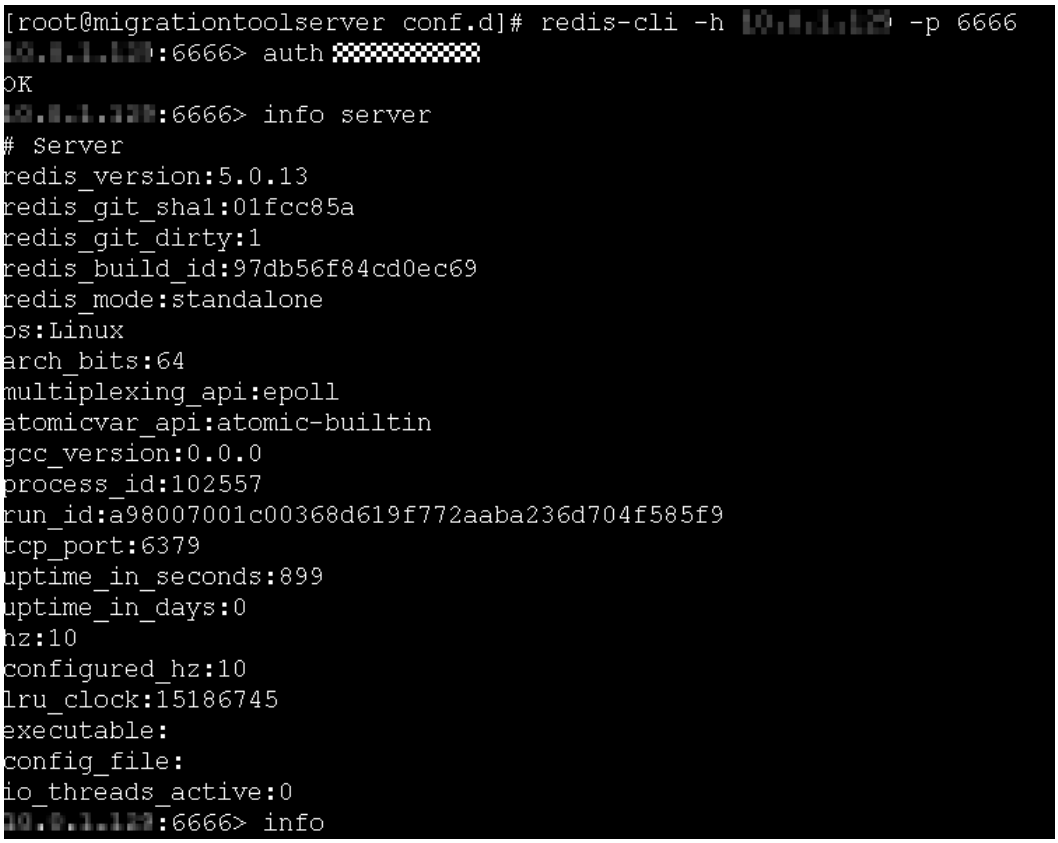
3. Restart Nginx.
```
service nginx restart
```
4. Verify whether Nginx has been started.
```
netstat -an|grep 6666
```
If the port is being listened, Nginx has been started successfully.

Figure 6-7 Verification result

Step 6 Run the following command on the Huawei Cloud ECS to test the network connection of port 6666:
```
redis-cli -h {target_ecs_address} -p 6666 -a {password}
```


{target_ecs_address} is the EIP of the Huawei Cloud ECS, **6666** is the listening port of the Huawei Cloud ECS, and *{password}* is the source Redis password. If there is no password, leave it blank.

Figure 6-8 Connection example



Step 7 Prepare the migration tool redis-shake.

1. Log in to the Huawei Cloud ECS.
2. Download redis-shake. Version 2.0.3 is used as an example. You can use **other redis-shake versions** as required.
   ```
   wget https://github.com/tair-opensource/RedisShake/releases/download/release-v2.0.3-20200724/redis-shake-v2.0.3.tar.gz
   ```
3. Decompress the redis-shake file.
   ```
   tar -xvf redis-shake-v2.0.3.tar.gz
   ```

Step 8 Configure the redis-shake configuration file.

1. Go to the directory generated after the decompression.
   ```
   cd redis-shake-v2.0.3
   ```
2. Modify the **redis-shake.conf** configuration file.
   ```
   vim redis-shake.conf
   ```

   Modify the source Redis configuration.

   - source.type

     Type of the source Redis instance. Use **standalone** for single-node, master/standby, and Proxy Cluster, and **cluster** for cluster instances.

   - source.address

EIP of the Huawei Cloud ECS and the mapped port of the source forwarding server (Huawei Cloud ECS's listening port 6666). Separate the EIP and port number with a colon (:).

- source.password_raw

  Password of the source Redis instance. If no password is set, you do not need to set this parameter.

Modify the target DCS configuration.

- target.type

  Type of the DCS Redis instance. Use **standalone** for single-node, master/standby, and Proxy Cluster, and **cluster** for cluster instances.

- target.address

  Colon (:) separated connection address and port of the DCS Redis instance.

- target.password_raw

  Password of the DCS Redis instance. If no password is set, you do not need to set this parameter.

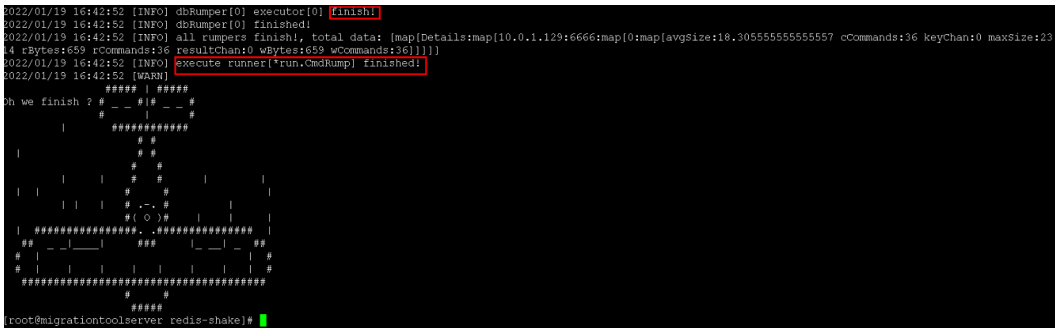
3. Press **Esc** to exit the editing mode and enter **:wq!**. Press **Enter** to save the configuration and exit the editing interface.

Step 9 Run the following command to start redis-shake and migrate data in the **rump** (online in full) mode:

```
./redis-shake.linux -conf redis-shake.conf -type rump
```

Figure 6-9 Migration process

Figure 6-10 Migration result



Step 10 After the migration is complete, use redis-cli to connect to the source and target Redis instances to check whether the data is complete.

1. Connect to the source and target Redis instances, respectively.

For details, see **Access Using redis-cli**.

2. Run the **info keyspace** command to check the values of **keys** and **expires**.
3. Calculate the differences 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.

Step 11 Delete the redis-shake configuration file.

----End

7 Migrating Data from DCS to Self-Hosted Redis

Scenario

You can use the online migration function of the DCS console to migrate 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 **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-shake to import the file to the target.

  For details about how to install and use redis-shake, see **Self-Hosted Redis Cluster Migration with redis-shake** and **redis-shake configuration instructions**.
- Rump

  This tool is recommended for online migration if possible. For details, see **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 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}*

8.3 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, read/write splitting, or master/standby DCS instance.

- If the target is a Redis Cluster instance (a Redis Cluster instance has only one DB):

  Solutions:

  a. Combine different DBs in the source Redis instance into one DB.

  b. 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.

- If the target is a Proxy Cluster instance:

  By default, the multi-DB function is disabled for Proxy Cluster instances and only one DB (DB0) is available. To use more than one DB, enable multi-DB by referring to **Enabling Multi-DB** before migration.

8.4 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 by default, 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".

8.5 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.

- Impact of capacity expansion of the source instance

  During online migration, expanding the source instance's capacity may affect the migration or customer data. If the memory of the source instance becomes insufficient during the migration, stop the migration task and then expand the capacity.

- Timing

  Migration should take place during off-peak hours.

- Version restrictions

  You can migrate data from an earlier version to a later version, and vice versa, but you need to check whether the target instance supports the commands used in your service systems.

- Multi-DB

  If both the target and source are Proxy Cluster DCS instances, ensure that the **multi-db** parameter settings of the two are the same. Otherwise, the migration will fail.

8.6 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 the cloud service is also involved, deploy your service systems before migrating your cached data.

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 DCS, 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.

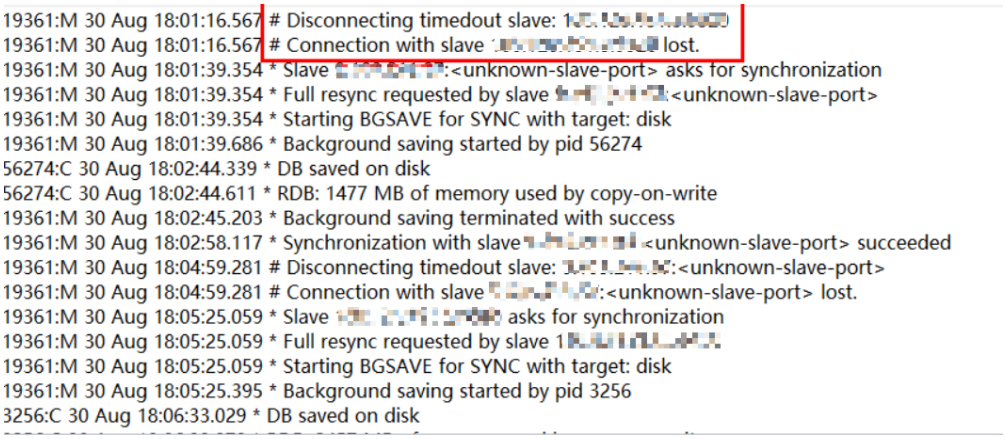
8.7 Can I Migrate Data Between DCS Memcached and Redis Instances?

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

8.8 What If "Disconnecting timedout slave" and "overcoming of output buffer limits" Are Reported on the Source Instance During Online Migration?

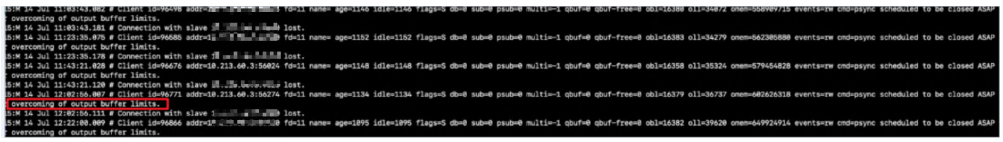
The following error messages may be displayed during online migration:

- "Disconnecting timedout slave" is reported on the source instance, as shown in the following figure:



  Solution: Set the **repl-timeout** parameter of the source Redis instance to 300s.

- "overcoming of output buffer limits" is reported on the source instance, as shown in the following figure:



  Solution: Set the **client-output-buffer-limit** parameter of the source Redis instance to 20% of the maximum memory of the instance.

8.9 Why Is Memory of a DCS Redis Instance Unchanged After Data Migration Using Rump, Even If No Error Message Is Returned?

For details on how to use Rump, see the **Data Migration Guide**.

Possible causes:

- Rump does not support migration to cluster DCS instances.
- Commands are incorrectly run in Rump.

8.10 Why Are Processes Frequently Killed During Data Migration?

Possible cause: The memory is insufficient.

Solution: Expand the memory of the server on which the migration command is executed.

8.11 Is All Data in a DCS Redis Instance Migrated During Online Migration?

Migration between single-node and master/standby instances involves the full set of data. All DBs will be migrated, and you cannot migrate specified DBs. After the migration, a given key will remain in the same DB as it was before the migration.

By contrast, a cluster instance only has one DB, which is DB0. During the migration, data in all slots of DB0 is migrated.

8.12 Can I Migrate Data to Multiple Target Instances in One Migration Task?

No. A migration task allows data to be migrated to only one target instance. To migrate data to multiple target instances, create multiple migration tasks.

8.13 Why Does Migration Task Creation Fail?

Possible causes:

1. The underlying resources are insufficient.
2. The specifications of the ECS used for the migration are insufficient.
3. The memory of the target Redis created before the migration is less than that of the source Redis.

8.14 How Do I Enable the SYNC and PSYNC Commands?

- Migration within DCS:
  - By default, the **SYNC** and **PSYNC** commands can be used when self-hosted Redis is migrated to DCS.
  - During online migration between DCS Redis instances in the same region under the same account, the **SYNC** and **PSYNC** commands are automatically enabled.
  - During online migration between DCS Redis instances in different regions or under different accounts within a region, the **SYNC** and **PSYNC** commands are not automatically enabled, and online migration cannot be used. You can migrate data using backup files.
- Migration from other cloud vendors to DCS:
  - Generally, cloud vendors disable the **SYNC** and **PSYNC** commands. If you want to use the online migration function on the DCS console, contact the O&M personnel of the source cloud vendor to enable the commands. For offline migration, you can import backup files.
  - If incremental migration is not required, you can perform full migration by referring to **Online Full Migration of Redis from Another Cloud with redis-shake**. This method does not depend on **SYNC** and **PSYNC**.

8.15 Why Does Redis Cluster Migration Fail If It Uses Built-in Keys and Cross-Slot Lua Scripts?

If your source Redis Cluster uses a cross-slot Lua script with built-in keys and it fails to be migrated to a cluster DCS instance, you can use a master/standby or read/write splitting instance as the target.

In scenarios where slot distribution changes, such as cluster scaling and slot migration, errors may incur in running a cross-slot Lua script with built-in keys. Therefore, **cross-slot Lua scripts with built-in keys are not recommended for Redis Cluster instances**.

NOTE

Redis Cluster instances support cross-slot Lua scripts with built-in keys:

- Built-in keys: Keys are written in the Lua script instead of being input through a function.
- Cross-slot: All slots in a Lua script are on one shard.

Symptom

Online migration or backup import may fail when the source instance is a Redis Cluster that uses a cross-slot Lua script with built-in keys.

Solution

Select a master/standby or read/write splitting instance as the target.

Suggestion

Do not use cross-slot Lua scripts with built-in keys for Redis Cluster instances.

NOTE

- Redis Cluster instances support cross-slot Lua scripts with built-in keys:
  - Built-in keys: Keys are written in the Lua script instead of being input through a function.
  - Cross-slot: All slots in a Lua script are on one shard.
- In scenarios where slot distribution changes, such as cluster scaling and slot migration, errors may incur in running a cross-slot Lua script with built-in keys.

8.16 Handling Migration Errors

This section provides solutions to common migration errors.

Restart data sync failed.

Solution:

1. **Check whether the source Redis has big keys**. If it does, split the big keys into small keys before migration.
2. Check the specifications of the target Redis instance and whether other tasks are being performed on the instance.
   - If the memory of the target Redis instance is smaller than the size of the data to be migrated, the memory will be used up during the migration and the migration will fail.
   - If a master/standby switchover is being performed on the target Redis instance, contact technical support to stop the master/standby switchover task and start it only after the data migration is completed.
3. Send the error information to technical support.

The Redis service address is unreachable.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**
- **Bandwidth**
- **Redis Performance**

Redis authentication failed.

Solution:

Ensure that the passwords of the source and target Redis databases are correct and are not changed during the migration.

If you forget the password, reconfigure the migration task after **resetting the password**.

RDB parsing failed.

Analysis:

Check the source Redis logs. Generally, this error is resulted from full output buffer when full synchronization takes too long or the size of incremental data is too large. You can use the following methods to solve this error:

- Increase the output buffer limit by **modifying the output-buffer-limit parameter**. This method is recommended.
- Increase the concurrency of redis-shake full synchronization by modifying the **parallel** parameter.
- Synchronize data during off-peak hours.

Failed to resume from the break point.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**
- **Bandwidth**
- **Redis Performance**

IP address and port number of Redis are invalid.

Send the error information to technical support.

Job failed.

Send the error information to technical support.

Failed to download the file.

Solution:

See **Why Am I Unable to Download an Object?**

The cluster does not support import of AOF files.

Analysis:

Redis Cluster instances only support .rdb files.

Failed to migrate the AOF file to the target Redis.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**
- **Bandwidth**
- **Redis Performance**

Failed to migrate the RDB file to the target Redis.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**
- **Bandwidth**
- **Redis Performance**

Failed to decompress the file.

Solution:

1. Ensure that the file is not damaged and the file format is correct.
2. Check whether the specifications of the migration ECS are too small and the disk space is full. In this case, expand the specifications of the migration ECS.

The file format is not supported.

Analysis:

Only .rdb, .aof, .zip, and .tar.gz files are supported.

Failed to migrate files.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**

- **Bandwidth**
- **Redis Performance**

No such file or directory.

Solution:

1. Check whether the specifications of the migration ECS are too small and the disk space is full. In this case, expand the specifications of the migration ECS.
2. Send the error information to technical support.

Failed to connect to the source Redis.

Solution:

1. See **Troubleshooting Redis Connection Failures**.
2. Check the specifications of the source Redis and the memory size of the migration ECS. If the memory of the migration server is small and the data volume of the source Redis is large, migration will be slow and data will be stacked on the migration ECS. To solve this error, you can expand the specifications of the migration ECS.
3. Run the **route -n** command on the migration ECS to check whether its route is normal.
4. Send the error information to technical support.

Failed to export the backup file from the source Redis.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**
- **Bandwidth**
- **Redis Performance**

Failed to import the backup file to the target Redis.

Check items:

- **Connection Between the Redis Instance and the ECS**
- **Public Access**
- **Password**
- **Instance Configuration**
- **Client Connections**
- **Bandwidth**
- **Redis Performance**

Failed to modify the redis-shake-conf configuration file due to incorrect parameters.

Solution:

1. Check whether the specifications of the migration ECS are too small and the disk space is full. In this case, expand the specifications of the migration ECS.
2. Send the error information to technical support.

Data synchronization failed. Source node: {0}; target node: {1}.

Solution:

1. **Check whether the source Redis has big keys**. If it does, split the big keys into small keys before migration.
2. Ensure that the specifications of the target Redis instance are not smaller than those of the source Redis. For details about how to view specifications, see **Viewing Instance Details**.
3. See **Troubleshooting Redis Connection Failures**.
4. Check the specifications of the source Redis and the memory size of the migration ECS. If the memory of the migration server is small and the data volume of the source Redis is large, migration will be slow and data will be stacked on the migration ECS. To solve this error, you can expand the specifications of the migration ECS.

Failed to deploy the migration tool.

Solution:

1. Check whether the network between the data plane and OBS is normal.
2. Send the error information to technical support.

Online migration failed.

Send the error information to technical support.

Failed to bind the port to the ECS.

Solution:

The underlying resources are insufficient to support the migration task. Contact technical support.

Failed to create the migration ECS.

Send the error information to technical support.

File operation exception.

Solution:

1. Check the specifications of the source Redis and the memory size of the migration ECS. If the memory of the migration server is small and the data

volume of the source Redis is large, migration will be slow and data will be stacked on the migration ECS. To solve this error, you can expand the specifications of the migration ECS.

2. Send the error information to technical support.

Command execution exception.

Solution:

- If the error information contains "listening-port" or "REPLCONF", check whether the **SYNC** and **PSYNC** commands are enabled on the source Redis instance and whether the underlying resources of the migration task are connected to the source and target Redis instances.

  For online migration, the source and target Redis instances must be connected, and the **SYNC** and **PSYNC** commands must be enabled on the source Redis instance. Otherwise, the migration will fail.

  – Check the network.

    Check whether the source Redis instance, the target Redis instance, and the migration VM are configured with the same VPC. If they are in the same VPC, check the security group rules (for DCS Redis 3.0 instances) or whitelists (for DCS Redis 4.0 or 5.0 instances) to ensure that the IP addresses and ports of the Redis instances are accessible. If they are in different VPCs, **create a VPC peering connection**.

    The source and target Redis instances must be accessible to the underlying VMs used for the migration task. For details about how to configure a security group or whitelist, see **How Do I Configure a Security Group?** or **Managing IP Address Whitelist**.

    If the source and target Redis instances are on different clouds, create a connection by referring to **Direct Connect documentation**.

  – Check the commands.

    By default, the **SYNC** and **PSYNC** commands are disabled by cloud vendors. To enable the commands, contact the O&M personnel of the cloud vendors.

    - Migration within Huawei Cloud:
      - By default, the **SYNC** and **PSYNC** commands can be used when self-hosted Redis is migrated to DCS.
      - During online migration between Huawei Cloud DCS instances in the same region under the same account, the **SYNC** and **PSYNC** commands are automatically enabled.
      - During online migration between Huawei Cloud DCS instances in different regions or under different accounts within a region, the **SYNC** and **PSYNC** commands are not automatically enabled, and online migration cannot be used. You can migrate data using backup files instead.
    - Migration from other cloud vendors to Huawei Cloud:

      Generally, cloud vendors disable the **SYNC** and **PSYNC** commands. If you want to use online migration, contact the O&M personnel of the source cloud vendor to enable the commands. For offline migration, you can import backup files.

- If the error information contains "read error" and the migration failed during full migration due to the large data size, disable auto-reconnect before starting the migration and enable it after incremental migration starts. In addition, **increase the value of repl-timeout** and modify the output buffer of the source Redis based on the source memory size. For example, if the memory size of the source Redis is 24 GB, you can run the **client-output-buffer-limit slave 2gb 2gb 600** command to change the buffer size to 2 GB.
- If the error information contains "write: connection reset by peer", the target Redis memory may be too small. As a result, data cannot be synchronized when the target memory is full. In this case, set this parameter to **increase the specifications of the target Redis** to at least the same as the specifications of the source.
- If the error information contains "read: connection reset by peer", the source Redis is deployed in master/standby mode, and master/standby switchover occurs frequently during the migration. In this case, **check whether the source Redis has big keys**. If it does, split the big keys into small keys before migration. You can also run the **config set slave-priority 0** command to forcibly disable master/standby switchover and enable it after the migration is complete. If the target Redis instance is Proxy Cluster, check the size of the pipeline. Run the **proxy.config set client-max-pipeline 50000** command to change this limit to 50,000 for proxies.
- Send the error information to technical support.

Decoding or parsing failed.

Solution:

1. Check whether the specifications of the migration ECS are too small and the disk space is full. In this case, expand the specifications of the migration ECS.
2. Send the error information to technical support.

Unknown or unsupported command.

Solution:

Check whether the **SYNC** and **PSYNC** commands are enabled on the source Redis instance. If they are not enabled, contact technical support.

For online migration, the source and target Redis instances must be connected, and the **SYNC** and **PSYNC** commands must be enabled on the source Redis instance. Otherwise, the migration will fail.

- Check the network.

  Check whether the source Redis instance, the target Redis instance, and the migration VM are configured with the same VPC. If they are in the same VPC, check the security group rules (for DCS Redis 3.0 instances) or whitelists (for DCS Redis 4.0 or 5.0 instances) to ensure that the IP addresses and ports of the Redis instances are accessible. If they are in different VPCs, **create a VPC peering connection**.

  The source and target Redis instances must be accessible to the underlying VMs used for the migration task. For details about how to configure a security group or whitelist, see **How Do I Configure a Security Group?** or **Managing IP Address Whitelist**.

If the source and target Redis instances are on different clouds, create a connection by referring to **Direct Connect documentation**.

- Check the commands.

  By default, the **SYNC** and **PSYNC** commands are disabled by cloud vendors. To enable the commands, contact the O&M personnel of the cloud vendors.

  - Migration within Huawei Cloud:
    - By default, the **SYNC** and **PSYNC** commands can be used when self-hosted Redis is migrated to DCS.
    - During online migration between Huawei Cloud DCS instances in the same region under the same account, the **SYNC** and **PSYNC** commands are automatically enabled.
    - During online migration between Huawei Cloud DCS instances in different regions or under different accounts within a region, the **SYNC** and **PSYNC** commands are not automatically enabled, and online migration cannot be used. You can migrate data using backup files instead.
  - Migration from other cloud vendors to Huawei Cloud:

    Generally, cloud vendors disable the **SYNC** and **PSYNC** commands. If you want to use online migration, contact the O&M personnel of the source cloud vendor to enable the commands. For offline migration, you can import backup files.

Data synchronization failed.

Solution:

1. If the error information contains "key name is busy", a key with the same name already exists in the target Redis. In this case, delete the key.
2. If the error information contains "not in the same slot", reconstruct your service. Do not use cross-slot keys in a multi-key command. You can also use a master/standby instance instead of a Proxy Cluster instance as the target.
3. If the error information contains "read: connection reset by peer", the source Redis is deployed in master/standby mode, and master/standby switchover occurs frequently during the migration. In this case, **check whether the source Redis has big keys**. If it does, split the big keys into small keys before migration. You can also run the **config set slave-priority 0** command to forcibly disable master/standby switchover and enable it after the migration is complete. If the target Redis instance is Proxy Cluster, check the size of the pipeline. Run the **proxy.config set client-max-pipeline 50000** command to change this limit to 50,000 for proxies.

Failed to import the backup file.

Send the error information to technical support.

8.17 Troubleshooting Data Migration Failures

When you use the console to migrate data, the migration may fail if you select an inappropriate migration scheme, the **SYNC** and **PSYNC** commands are not allowed

on the source Redis instance, or the network between the source and target Redis instances is disconnected.

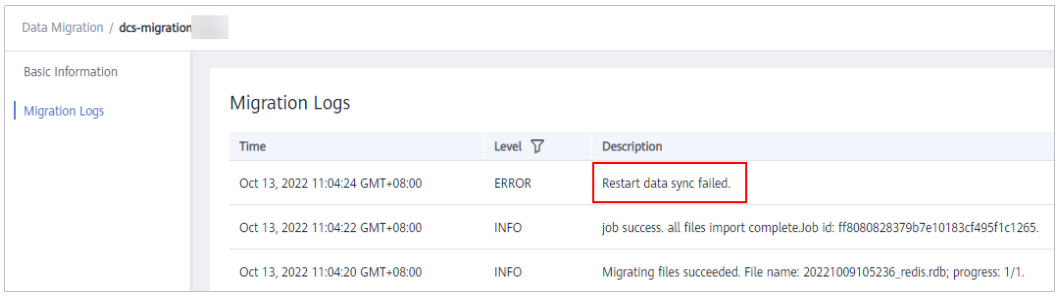
This section describes how to troubleshoot data migration failures on the DCS console.

Procedure

Step 1 Click the name of a migration task and then go to the **Migration Logs** page.

Step 2 Check the migration logs. Rectify the fault based on the error log. For details, see **Handling Migration Errors**.

Figure 8-1 Viewing migration logs



Step 3 Check whether you used the appropriate migration scheme.

An appropriate scheme varies depending on the scenario. That is, the scheme for migrating data from self-hosted Redis to a DCS instance, from other vendors' Redis to a DCS instance, and between DCS instances are different. If you migrate data between DCS instances, the source instance cannot be a higher version than the target instance.

If the migration scheme is inappropriate, the migration will fail. For details about migration schemes, see **Migration Tools and Schemes**.

Step 4 Check whether the **SYNC** and **PSYNC** commands are enabled on the source Redis instance and whether the underlying resources of the migration task are connected to the source and target Redis instances.

This operation is required only for online migration.

For online migration, the source and target Redis instances must be connected, and the **SYNC** and **PSYNC** commands must be enabled on the source Redis instance. Otherwise, the migration will fail.

- Check the network.

  Check whether the source Redis instance, the target Redis instance, and the migration VM are configured with the same VPC. If they are in the same VPC, check the security group rules (for DCS Redis 3.0 and 6.0 professional instances) or whitelists (for DCS Redis 4.0, 5.0, and 6.0 basic instances) to ensure that the IP addresses and ports of the Redis instances are accessible. If they are in different VPCs, **create a VPC peering connection**.

  The source and target Redis instances must be accessible to the underlying VMs used for the migration task. For details about how to configure a security group or whitelist, see **How Do I Configure a Security Group?** or **Managing IP Address Whitelist**.

To allow the VM resources used by the migration task to access the source and target Redis instances, outbound rules of the migration security group must allow traffic over the IP addresses and ports of the source and target instances. For details, see **How Do I Configure a Security Group?**

If the source and target Redis instances are on different clouds, create a connection by referring to **Direct Connect documentation**.

NOTE

The VM used by a migration task uses an IP address. Configuring the whitelist (Redis 4.0/5.0/6.0 basic) or inbound security group rules (Redis 3.0/6.0 professional) of instances is to allow the migration VM to access the source and target Redis.

- Check the commands.

  By default, the **SYNC** and **PSYNC** commands are disabled by cloud vendors. To enable the commands, contact the O&M personnel of the cloud vendors.

  - Migration within HUAWEI CLOUD:
    - By default, the **SYNC** and **PSYNC** commands can be used when self-hosted Redis is migrated to DCS.
    - During online migration between Huawei Cloud DCS instances in the same region under the same account, the **SYNC** and **PSYNC** commands are automatically enabled.
    - During online migration between Huawei Cloud DCS instances in different regions or under different accounts within a region, the **SYNC** and **PSYNC** commands are not automatically enabled, and online migration cannot be used. You can migrate data using backup files instead.
  - Migration from other cloud vendors to HUAWEI CLOUD:

    Generally, cloud vendors disable the **SYNC** and **PSYNC** commands. If you want to use online migration, contact the O&M personnel of the source cloud vendor to enable the commands. For offline migration, you can import backup files.

Step 5 Check the source Redis instance for big keys. For details, see **Analyzing Big Keys and Hot Keys**.

If the source Redis instance has big keys, split them into small keys before migration.

Step 6 Check the specifications of the target Redis instance and whether other tasks are being performed on the instance.

If the memory of the target Redis instance is smaller than the size of the data to be migrated, the memory will be used up during the migration and the migration will fail.

If a master/standby switchover is being performed on the target Redis instance, contact technical support to stop the master/standby switchover task and start it only after the data migration is completed.

Step 7 If you migrate data from a single-node or master/standby instance to a cluster instance, check the following items:

- By default, a Proxy Cluster instance has only one database (DB0). Before you migrate data from a single-node or master/standby instance to a Proxy

Cluster instance, check whether any data exists on databases other than DB0. If yes, enable multi-DB for the Proxy Cluster instance by referring to **Enabling Multi-DB**.

- By default, a Redis Cluster instance has only one DB (DB0). Before you migrate data from a single-node or master/standby instance to a Redis Cluster instance, check whether any data exists on databases other than DB0. To ensure that the migration succeeds, move all data to DB0 by referring to **Online Migration with Rump**.

Step 8 Check whether the migration task is performed correctly.

Check whether the IP address and the instance password are correct.

Step 9 Check the whitelist.

Step 10 If the fault persists, contact technical support.

----End

8.18 Can I Migrate Data from a Lower Redis Version to a Higher One?

Yes. Redis is backward compatible.

The version of the source Redis (DCS, self-built, or another cloud) can be earlier than or the same as the target DCS instance.

