

Alibaba Cloud ApsaraDB for Redis

User Guide

Issue: 20190422

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK.
<code>Courier font</code>	It is used for commands.	Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>

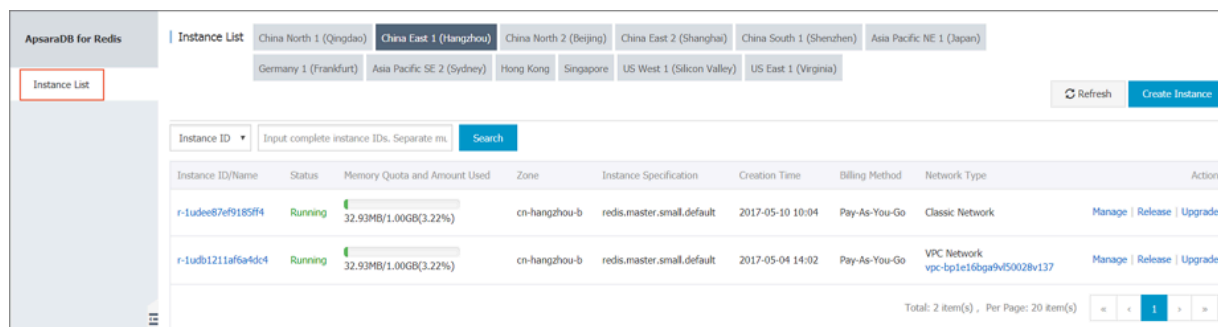
Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>switch {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Log on to the console.....	1
2 Manage instances.....	2
2.1 Change configuration.....	2
2.2 Parameter settings	3
2.3 Set the network type.....	3
2.4 Modify retention period for classic network.....	4
2.5 Upgrade a minor version.....	5
2.6 Clear data.....	5
2.7 Release an instance.....	6
3 Backup and recovery.....	7
3.1 Back up and recover data on the Redis Console.....	7
4 Alarm settings.....	10
5 Performance monitoring.....	11
6 Migrate data.....	14
6.1 Migrate from on-premises Redis to ApsaraDB Redis.....	14
6.1.1 Migrate data from an on-premises Redis cluster to an ApsaraDB Redis cluster.....	14
6.1.2 Migrate data with redis-port.....	16
6.1.3 Use Data Transmission Service (DTS) to migrate data.....	18
6.1.4 Use RDB files to migrate from on-premises Redis to ApsaraDB Redis.....	26
6.1.5 Use AOF to migrate on-premises Redis data to ApsaraDB Redis.....	27
6.2 Migrate from ApsaraDB Redis to on-premises Redis.....	28
6.2.1 Migrate backup sets.....	28
6.3 Migrate between ApsaraDB Redis instances.....	29
6.3.1 Use Rump to migrate data between ApsaraDB Redis databases.....	29
6.3.2 Use the redis-port tool to migrate data between ApsaraDB Redis instances of different accounts.....	31
7 Application for payment and review.....	33

1 Log on to the console

Click [here](#) to access the Redis console. The Instance List page is displayed, as shown in the following figure.



The screenshot displays the 'Instance List' page for ApsaraDB for Redis. The page features a top navigation bar with various region tabs, a search bar, and a table of instances. The table has the following columns: Instance ID/Name, Status, Memory Quota and Amount Used, Zone, Instance Specification, Creation Time, Billing Method, Network Type, and Action. Two instances are listed, both in a 'Running' state. The page also includes a search bar, a refresh button, and a 'Create Instance' button.

Instance ID/Name	Status	Memory Quota and Amount Used	Zone	Instance Specification	Creation Time	Billing Method	Network Type	Action
r-1udee87ef9185ff4	Running	32.93MB/1.00GB(3.22%)	cn-hangzhou-b	redis.master.small.default	2017-05-10 10:04	Pay-As-You-Go	Classic Network	Manage Release Upgrade
r-1udb1211af6a4dc4	Running	32.93MB/1.00GB(3.22%)	cn-hangzhou-b	redis.master.small.default	2017-05-04 14:02	Pay-As-You-Go	VPC Network vpc-bp1e16bga9v150028v137	Manage Release Upgrade

Total: 2 item(s) , Per Page: 20 item(s)

2 Manage instances

2.1 Change configuration

Alibaba Cloud ApsaraDB for Redis provides two billing methods: Monthly/annually payment and pay-by-flow method. The latter can be converted to the former. The configuration of the two mode can be changed.

Background

Changing the instance configuration incurs a change in charges. For more information about the billing standard, see [ApsaraDB for Redis pricing](#).

Attention

- Pay-As-You-Go and Subscription instances both support real-time configuration changes.
- Cluster instances and non-cluster instances can be switched between each other.
- During the configuration change, the instance may be disconnected for a few seconds. Please change the configuration during off-peak hours.

Change the configuration of a Pay-As-You-Go instance

1. Log on to the [Redis console](#).
2. Locate the target instance in the instance list. Click Change Configurations in the Action column.
3. On the Upgrade page, change the configuration and click Activate.

It notifies the user with a successful resetting and takes an immediate effect upon resetting configuration. (Pay by new mode at once)

Change the configuration of a Subscription instance

1. Log on to the [Redis console](#).
2. Locate the target instance in the instance list. Click Upgrade or Downgrade in the Action column.
3. On the Upgrade page, change the configuration and click Pay.
4. Review the order information and click Pay.

The procedure ends.

2.2 Parameter settings

ApsaraDB for Redis allows you to customize some instance parameters. For more information about the parameters that can be modified, see Parameter Settings on the [Redis console](#).

Background

As ApsaraDB for Redis is completely compatible with the native database service, their parameter setting methods are similar for users. You can modify parameters through the ApsaraDB for Redis Console by referring to this example.

For a description of the database parameters, click the following link to see the official documentations for different database versions.

- [redis.conf for Redis 3.0](#)
- [redis.conf for Redis 2.8](#)

Procedure

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Parameter Settings in the left-side navigation pane.
4. Select the expected parameter and click Modify.
5. Modify the parameter values and click OK.

2.3 Set the network type

Background

ApsaraDB supports two types of network: classic and Virtual Private Cloud (VPC), a classic network and VPC have following differences:

- Classic network: The cloud services on a classic network are not isolated, and unauthorized access to a cloud service is blocked only by the security group or whitelist policy of the service.
- VPC : The VPC helps you build an isolated network environment on Alibaba Cloud. You can customize the route table, IP address range, and gateway in VPC. In addition, you can combine your machine room and cloud resources in the Alibaba

Cloud VPC into a virtual machine room through a leased line or VPN to migrate applications to the cloud smoothly.

Note:

- A classic network can be switched to VPC, but not vice versa.
- When switching the classic network to VPC, you can choose to retain the classic network IP address.

Prerequisites

To switch a classic network type Redis instance to a VPC network type instance, you should firstly create a VPC and a switch in the same region of the instance.



Notice:

Choose the same zone as the Redis instance when creating a switch.

Procedure

1. Log on to [Redis Console](#), find the target instance, and click Manage.
2. Click Switch to VPC on the Instance Information page.
3. On the Switch to VPC page, select VPC and VSwitch. Select whether to retain the classic network IP address and the retention period, and then click OK.



Notice:

On the Instance Information page, Click Refresh to view the access address.

2.4 Modify retention period for classic network

After setting the retention period for the access address of a classic network, you can prolong the retention time on the console before the time expires.

In the hybrid access period, you can change the retention time of the original classic network at any time as needed. The expiration date is recalculated from the new date. For example, if the Intranet address of the original classic network is set to expire on August 18, 2017 and you change the expiration date to 14 days later on August 15, 2017, the Intranet address is released on August 29, 2017.

Procedure

1. Log on to the [Redis console](#).

2. On the Instance List page, find the target instance and click Manage to go to the Instance Information page.
3. In the Retained Connection Address of the Classic Network area, click Modify Retention Period.
4. In the displayed dialog box, select a new expiration date and click OK.

2.5 Upgrade a minor version

Background

Deep kernel optimization is performed on ApsaraDB for Redis to fix security vulnerabilities and improve the stability of service. You can upgrade to the latest kernel version in one click on the console.



Notice:

- The system automatically detects the kernel version of your instance. If the current version is the latest, the Basic Information page of the console does not show the Upgrade Minor Version button.
- Upgrading the kernel version interrupts your connection for 30 seconds, so perform upgrade during off-peak hours and make sure your applications have a reconnection mechanism.

Procedure

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Click Upgrade Minor Version in the Basic Information column.
4. Click Upgrade Now in the Upgrade Minor Version window.

On the Basic Information page, the instance status shows Upgrading minor version... . Upgrade is completed when the instance status shows In use.

2.6 Clear data



Notice:

The data cleanup operation clears all data of the target instance and the cleared data cannot be recovered.

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Click Clear Data and then click OK in the displayed confirm box.
4. On the Mobile Phone Verification page, obtain and enter the verification code to complete data cleanup.

2.7 Release an instance



Note:

Pay-As-You-Go instance can be release at any time, whereas subscribed instance cannot be manually deleted or released.

Procedure

1. Log on to [Redis Console](#) and find the target instance.
2. Click Instance ID or Manage to go to the Instance Information page.
3. Click Release and click OK in the popup window.
4. On the Mobile Phone Verification page, obtain and enter the verification code to release the instance.

3 Backup and recovery

3.1 Back up and recover data on the Redis Console

As an increasing number of businesses use ApsaraDB for Redis as the ultimate persistent storage engine, users have posed higher data reliability requirements. The ApsaraDB for Redis backup and recovery solution enables comprehensive data reliability upgrade.

Automatic backup (backup policy setting)

Background

As more and more applications use ApsaraDB for Redis for persistent storage, conventional backup mechanisms are required to quickly recover data in the event of misoperation. Alibaba Cloud runs RDB snapshot backup on slave nodes to protect the performance of your instance during the backup process. Alibaba Cloud also provides convenient console operations, so you can customize the backup settings.

Procedure

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Backup and Recovery in the left-side navigation pane .
4. Click Backup Settings.
5. Click Edit to customize the automatic backup cycles and times.



Notice:

By default, backup data is retained for 7 days. This setting cannot be modified.

6. Click OK to complete automatic backup setting.

Manual backup (instant backup)

In addition to the general backup settings, you can initiate a manual backup request on the console at any time.

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Backup and Recovery in the left-side navigation pane .

4. Click Create Backup in the upper-right corner.
5. Click OK to instantly back up the instance.

**Notice:**

On the Backup Data page, you can select time ranges to query historical backup data. By default, backup data is retained for 7 days, so you can query historical backup data from the last 7 days.

Backup archiving

Background

Due to industry regulatory or corporate policy requirements, you may need to regularly archive Redis data backups. ApsaraDB for Redis provides a backup archiving function at no charge currently and saves automatic and manual backup files to OSS. Now, Alibaba Cloud stores your backup archives on OSS for 7 days at no charge. After 7 days, the files are automatically deleted.

If you must retain these archives for a longer period of time, you can copy the link on the console and manually download the database backup files for local storage.

Procedure

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Backup and Recovery in the left-side navigation pane .
4. On the backup data page, select the backup data to be archived and click Download.

Data recovery

The data recovery function minimizes the damage caused by database misoperations . Now, ApsaraDB for Redis supports data recovery from backups.

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Backup and Recovery in the left-side navigation pane .
4. Click the Backup Data tab on the Backup and Recovery page.
5. Select the time range for recovery and click Search. Then select the target backup file and click Recover Data.

6. In the Data Recovery window, click OK to recover the data directly to the original instance. Alternatively, you can choose Clone Instance to recover the backup data to a new instance. After verifying that the recovered data is correct, you can recover the data to the original instance.

**Notice:**

As the data recovery operation is highly risky, we suggest using the clone instance method if time permits. This method creates a Pay-As-You-Go instance based on the backup data set to be recovered. After verifying that the data is correct, you can recover the data to the original instance.

Clone instance

Background

During routine maintenance projects, O&M engineers often need to quickly deploy a new application. When application deployment is relatively simple, a new instance can be conveniently created based on an ECS image file. At the database level, however, deployment is more complex. O&M engineers must purchase or install a new database and then initialize relevant database scripts (to create tables, triggers, and views). In such a scenario, many trivial operations must be performed and the error rate is relatively high. Especially in the gaming industry with fast service activation, the rapid deployment of new applications often needs to be repeated many times each day.

To address this pain point, ApsaraDB for Redis develops the clone instance function, enabling you to clone a new subscribed instance or Pay-As-You-Go instance from backup files quickly. Then you can perform complex operations of database development and deployment with a single click on a graphic interface, significantly improving productivity.

Procedure

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Backup and Recovery in the left-side navigation pane.
4. On the Backup Data page, select the expected backup data set and click Clone Instance.

4 Alarm settings

Background information

ApsaraDB for Redis provides an instance monitoring function and sends an SMS message to you when detecting an instance exception.

Monitoring and alarming are implemented through CloudMonitor. CloudMonitor enables you to set metrics and notify all contacts in the alarm contact group when the alarm policies of the metrics are triggered. You can maintain an alarm contact group corresponding to an alarm metric so that relevant contacts are promptly notified when an alarm occurs.

Procedure

1. Log on to the [Redis console](#) and find the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Alarm Settings in the left-side navigation pane.
4. Click Alarm Settings to go to the CloudMonitor console. You can click Refresh to manually refresh the current status of the monitoring metrics.
5. Select Alarm Rules > Create Alarm Rule.
6. Add alarm rules on the Batch Alarm Rule Settings page.
7. Click Next to set the notification object. You can click Quickly Create a Contact Group to create an alarm contact or alarm contact group.
8. Click Confirm and then click Close.



Notice:

After the alarm setting is completed, you can modify, disable, and delete alarm rules on the Alarm Rules page of the CloudMonitor console. You can also view the alarm history on this page.

5 Performance monitoring

Background information

ApsaraDB for Redis provides 10 monitoring groups. You can customize metrics on the ApsaraDB for Redis console based on business requirements, or enable real-time monitoring for ApsaraDB for Redis instances using DMS for Redis.

Metric descriptions

Monitoring group	Data metric	Description
Basic monitoring group	The basic instance monitoring information	Includes QPS, bandwidth, and memory usage.
Keys monitoring group	Monitoring statistics on the use of key value-related commands	Number of times that commands used to delete keys, determine whether a key exists, and perform other such operations are called.
String monitoring group	Monitoring statistics on the use of string data-related commands	Number of times that string data commands, such as append and mget, are called.
Hashes monitoring group	Monitoring statistics on the use of hash data-related commands	Number of times that hash data commands, such as hget and hdel, are called.
Lists monitoring group	Number of times that list data commands, such as blpop and brpop, are called.	Number of times that List data commands, such as blpop and brpop, were called
Sets monitoring group	Monitoring statistics on the use of set data-related commands	Number of times that set data commands, such as saadd and scard, are called .
Zsets monitoring group	Monitoring statistics on the use of zset data-related commands	Number of times that zset data commands, such as zadd and zcard, are called.

Monitoring group	Data metric	Description
HyperLog monitoring group	Monitoring statistics on the use of HyperLogLog data-related commands	Number of times that HyperLogLog data commands, such as pfadd and pfcount, are called.
Pub/Sub monitoring group	Monitoring statistics by using commands related to pub/sub functions	Number of times that pub/sub function commands, such as publish and subscribe, are called.
Transaction monitoring group	Monitoring statistics on the use of transaction-related commands	Number of times that transaction-related commands, such as watch, multi, and exec, are called.

Start real-time monitoring

1. Log on to the [Redis console](#) and locate the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Click Log on to Database in the upper right corner.
4. On the data console logon page, enter the ID and password of the ApsaraDB for Redis instance to go to the homepage of DMS for Redis.
5. On the Performance Monitoring page, click Real-time Monitor.

Custom metrics

1. Log on to the [Redis console](#) and locate the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Performance Monitoring in the left-side navigation pane.
4. Click Custom Metrics, select the expected monitoring group, and click OK.

View historical monitoring data

1. Log on to the [Redis console](#) and locate the target instance.
2. Click the instance ID or Manage to go to the Instance Information page.
3. Select Performance Monitoring in the left-side navigation pane.
4. On the Performance Monitoring page, query the historical monitoring data of the instance.



Notice:

- You can select a time range to query historical monitoring data.
- Cluster instances support viewing of the historical monitoring data of each data node. You can click a data node in Instance Architecture Diagram on the Instance Information page or select Data Node on the Performance Monitoring page of a cluster instance to query the historical monitoring data of the data node.

6 Migrate data

6.1 Migrate from on-premises Redis to ApsaraDB Redis

6.1.1 Migrate data from an on-premises Redis cluster to an ApsaraDB Redis cluster

You can use redis-sync-manager to migrate data from an on-premises Redis cluster to an ApsaraDB Redis cluster.

Prerequisites

- You have specified redis-port in the `$PATH` environment variable, because redis-sync-manager depends on [redis-port](#) for the migration.
- You must be aware of the possible memory usage requirements for source and target basic data, and the current concurrency.
- Make sure that no slots in the source cluster are in an intermediate migration state.

Migration tools

- **Redis-port:** is used to synchronize data from a single Redis process to the target ApsaraDB for Redis cluster.
- **Redis-sync-manager:** serves as a supplement to redis-port (redis-port is encapsulated in redis-sync-manager) and is used to synchronize data from the source Redis cluster to the target ApsaraDB for Redis cluster.

Download the tool

[Redis-sync-manager](#)

Instructions

Run the following command to migrate data from a user-created Redis cluster to an ApsaraDB for Redis cluster:

```
./ redis - sync - manager -- from = src_host : src_port -- target  
= dst_host : dst_port [-- password = src_passwo rd ] [-- auth =  
dst_passwo rd ] [-- filterkey =" str1 | str2 | str3 "] [-- targetdb  
= DB ] [-- rewrite ] [-- bigkeysize = SIZE ] [-- logfile = REDISPORT  
. LOG ] [-- httpport = HTTPPORT ] [-- sync - parallel = INT ] [--  
sync - role =" master | slave "]
```

Migration mechanism of redis-sync-manager

Redis-sync-manager interacts with `src_host : src_port` to obtain information about the cluster topology through the `cluster nodes` command first. Then, a `IP : PORT` list of the shards to be synchronized is returned based on the `--sync - role` parameter setting. Finally, redis-sync-manager calls redis-port to synchronize data. (The concurrency of full data synchronization depends on the `--sync - parallel` parameter setting, and incremental data of each shard in the source cluster is synchronized to the corresponding shard in the target cluster.) The following table describes the related parameters.

Table 6-1: Parameter description

Parameter	Description
<code>src_host</code>	The domain name or IP address of the user-created Redis database. Set this parameter to the IP address of a process in the user-created Redis cluster.
<code>src_port</code>	The port of the user-created Redis database. Set this parameter to the port that corresponds to the IP address specified for <code>src_host</code> .
<code>src_password</code>	The password of the user-created Redis database.
<code>dst_host</code>	The domain name of the ApsaraDB for Redis database.
<code>dst_port</code>	The port of the ApsaraDB for Redis database.
<code>dst_password</code>	The password of the ApsaraDB for Redis database.
<code>str1 str2 str3</code>	Filters keys with <code>str1</code> , <code>str2</code> , or <code>str3</code> .
<code>DB</code>	The database to be synchronized to ApsaraDB for Redis.
<code>rewrite</code>	Overwrites a key that has already been written.
<code>bigkeysize=SIZE</code>	Indicates that when the written value is greater than <code>SIZE</code> , the big key write mode is used.
<code>--logfile=REDISPORT.LOG</code>	The name of a log file, for example, <code>edis-sync-manager.log</code> . During migration, different log files are generated for the migrated shards, and the serial number of the shard that is being migrated is automatically added to the name of its corresponding log file. Default value: <code>logs / redis - sync - manager . log</code> .
<code>--sync-parallel=INT</code>	Indicates whether concurrency is supported for data synchronization and displays the possible memory usage. Default value: <code>1</code> .

Parameter	Description
<code>--sync-role="master slave"</code>	Indicates the synchronization order of primary or secondary database of the source cluster. Default value: <code>master</code> .

Migration example

```

$redis-sync-manager --from=127.0.0.1:18380 --target=127.0.0.1:18385 --ncpu=2 --sync-role=slave --rewrite --sync-parallel=1 --auth= --password= --logfile=sync.log --pidfile=sync.pid
2018/10/24 14:42:29
changed param:
  --from=127.0.0.1:18382
  --logfile=sync.log.0-5460
  --pidfile=sync.pid.0-5460
  --httpport=16000
cmd:
  redis-port sync --from=127.0.0.1:18382 --password= --ncpu=2 --auth= --logfile=sync.log.0-5460 --pidfile=sync.pid.0-5460 --target=127.0.0.1:18385 --httpport=16000 --rewrite
2018/10/24 14:42:50 [FROM:127.0.0.1:18382]sync rdb done

2018/10/24 14:42:50
changed param:
  --httpport=16001
  --from=127.0.0.1:18383
  --logfile=sync.log.5461-10922
  --pidfile=sync.pid.5461-10922
cmd:
  redis-port sync --target=127.0.0.1:18385 --httpport=16001 --rewrite --from=127.0.0.1:18383 --password= --ncpu=2 --auth= --logfile=sync.log.5461-10922 --pidfile=sync.pid.5461-10922
2018/10/24 14:43:15 [FROM:127.0.0.1:18383]sync rdb done

2018/10/24 14:43:15
changed param:
  --from=127.0.0.1:18384
  --logfile=sync.log.10923-16383
  --pidfile=sync.pid.10923-16383
  --httpport=16002
cmd:
  redis-port sync --from=127.0.0.1:18384 --password= --ncpu=2 --auth= --logfile=sync.log.10923-16383 --pidfile=sync.pid.10923-16383 --target=127.0.0.1:18385 --httpport=16002 --rewrite
2018/10/24 14:43:38 [FROM:127.0.0.1:18384]sync rdb done

2018/10/24 14:43:38 127.0.0.1:18382 is sync is finished, syncing incremental data
2018/10/24 14:43:38 127.0.0.1:18383 is sync is finished, syncing incremental data
2018/10/24 14:43:38 127.0.0.1:18384 is sync is finished, syncing incremental data
2018/10/24 14:43:38 there are 3 shard needed to sync, 3 is finished , 0 is failed. please check...

```

The callouts in the picture are described as follows.

1. The synchronized information and synchronization status of each shard are displayed.
2. An `IP : PORT` is selected for each shard based on the `-- sync - role` parameter setting, and the IP:PORTs are printed out one by one.
3. The synchronization status of each shard is printed to the log file.

6.1.2 Migrate data with redis-port

You can use `redis-port` to migrate data from a self-managed Redis database to an ApsaraDB for Redis instance.

Prerequisites


- You have created a Linux-based Elastic Compute Service (ECS) instance in the VPC where the target ApsaraDB for Redis instance resides.
- You have downloaded [redis-port](#) in the ECS instance mentioned above.
- You have run `chmod u + x redis - port` to change `redis-port` into an executable file.

Procedure

1. Log in to the Linux system built in your ECS instance.
2. In the directory where redis-port resides, run below code to start the migration.

```
./ redis - port sync -- from = src_host : src_port -- Password
= src_passwo rd -- target = dst_host : dst_port -- auth
= dst_passwo rd [-- filterkey =" str1 | str2 | str3 "] [--
targetdb = dB ] [-- rewrite ] [-- bigkeysize = size ] [-- logfile =
redisport . log ]
```

Table 6-2: Redis-port Arguments

Argument	Description
src_host	Domain name (or IP) of the self-managed Redis database.
src_port	Port of the self-managed Redis database.
src_password	Password of the self-managed Redis database.
dst_host	Domain name of the ApsaraDB for Redis instance.
dst_port	Port of the ApsaraDB for Redis instance.
dst_password	Password of the ApsaraDB for Redis instance.
str1 str2 str3	Filter keys with str1, str2, or str3.
DB	Index of the self-managed Redis DB to be migrated.
rewrite	<p>Overwrite identical keys that already exist in the ApsaraDB for Redis instance.</p> <div>  Notice: If this argument is not set and identical keys exist in both databases, the migration may not run properly. </div>
bigkeysize	While writing keys larger than the value of size, enable big-key writing.
logfile	Specify a file to save the logs.

3. Monitor the logs to make sure the migration procedure runs as expected.

```
[root@redisTest ~]# ./redis-port sync --from=127.0.0.1:6379 --password= --target=r- .redis.rds.aliyuncs.com:6379 --auth=
2018/12/06 17:34:47 [INFO] set ncpu = 2, parallel = 2 filterdb = 0 targetdb = -1
2018/12/06 17:34:47 [INFO] set ncpu = 2, parallel = 2 filterdb = 0 targetdb = -1
2018/12/06 17:34:47 [INFO] sync from '127.0.0.1:6379' to 'r- .redis.rds.aliyuncs.com:6379' http '%!s(int=0)'
2018/12/06 17:34:47 [INFO] sync from '127.0.0.1:6379' to 'r- .redis.rds.aliyuncs.com:6379'
2018/12/06 17:34:47 [INFO] rdb file = 1124803
2018/12/06 17:34:47 [INFO] Aux information key:redis-ver value:3.2.12
2018/12/06 17:34:47 [INFO] Aux information key:redis-bits value:64
2018/12/06 17:34:47 [INFO] Aux information key:ctime value:1544088887
2018/12/06 17:34:47 [INFO] Aux information key:used-mem value:3708224
2018/12/06 17:34:47 [INFO] db_size:5000 expire_size:0
2018/12/06 17:34:48 [INFO] total=1124803 - 441129 [ 39%] entry=1919
2018/12/06 17:34:49 [INFO] total=1124803 - 868155 [ 77%] entry=3813
2018/12/06 17:34:50 [INFO] total=1124803 - 1124803 [100%] entry=5000
2018/12/06 17:34:50 [INFO] sync rdb done
2018/12/06 17:34:51 [INFO] sync: +forward=1 +nbypass=0 +nbytes=14
2018/12/06 17:34:52 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:53 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:54 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:55 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
2018/12/06 17:34:56 [INFO] sync: +forward=0 +nbypass=0 +nbytes=0
```



Note:

- Check the logs after `sync rdb done`, if the `+ nbytes` value of an entry containing `+ forward = 1` is greater than 14, it is an incremental-synchronization log. You can monitor the status of the incremental synchronization to determine the best time to switch databases.
- After full synchronization, the synchronization source (i.e. the self-managed Redis database) will send periodic `ping` requests, generating logs with `+ forward = 1` and `+ nbytes = 14`. These are not incremental-synchronization logs.

6.1.3 Use Data Transmission Service (DTS) to migrate data

You can use DTS to migrate data from an external Redis instance to an ApsaraDB for Redis instance.

Migrate data from an external Redis instance to an ApsaraDB for Redis instance

You can use [DTS](#) to migrate data between two Redis instances. If the source instance is an external Redis instance that supports incremental data synchronization, then data migration does not affect the write operation on your local applications.

This section describes the procedure of using DTS to migrate data from an external Redis instance to an ApsaraDB for Redis instance. You can follow the same procedure to migrate data from an ApsaraDB for Redis instance to another one.

Migration modes

If the source instance is an external Redis instance, both standard migration and incremental data migration are supported. If the source instance is an ApsaraDB for Redis instance, only standard migration is supported. Standard migration and incremental data migration have the following features and restrictions:

- **Standard migration**

DTS migrates all of the existing keys on an external Redis instance to an ApsaraDB for Redis instance.

- **Incremental data migration**

During incremental data migration, updated keys on an external Redis instance are synchronized to an ApsaraDB for Redis instance. After the synchronization process, the ApsaraDB for Redis instance starts to replicate data dynamically from the external Redis instance. By using incremental data migration, an external Redis instance can smoothly migrate data to an ApsaraDB for Redis instance while providing services normally.

Migration features

Redis incremental data migration supports the following commands:

- APPEND
- BITOP, BLPOP, BRPOP, BRPOPLPUSH,
- DECR, DECRBY, DEL,
- EVAL, EVALSHA, EXEC, EXPIRE, EXPIREAT,
- FLUSHALL, FLUSHDB,
- EOADD, GETSET,
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, HSETNX,
- INCR, INCRBY, INCRBYFLOAT,
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, LTRIM,
- MOVE, MSET, MSETNX, MULTI,
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, PUBLISH,
- RENAME, RENAMENX, RESTORE, RPOP, RPOPLPUSH, RPUSH, RPUSHX,
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, SUNIONSTORE,
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, ZREMRANGEBYSCORE

Prerequisites

If the source Redis instance is an external Redis instance connected to an Alibaba Cloud VPC through a leased line, or an ApsaraDB for Redis instance in a VPC, then a proxy is required for data forwarding.

To allow DTS to connect to VPC-connected Redis instances or external Redis instances connected to Alibaba Cloud through a leased line, you must select an ECS instance with an EIP assigned, and deploy Nginx on the ECS instance to facilitate Nginx proxy .

The Redis instance and the ECS instance must be connected to the same VPC. For external Redis instances that are connected to Alibaba Cloud through a leased line, the VPC where the ECS instance resides must be an Alibaba Cloud VPC.

The following section describes how to use Nginx to configure forwarding settings, in order to allow the DTS server to connect to a VPC-connected Redis instance.

1. Deploy Nginx

Run the following command on the ECS instance to deploy Nginx .

```
yum install nginx
```

2. Configure Nginx forwarding settings

After you have installed Nginx, modify the configuration file `/etc/nginx/nginx.conf` and set the backend to listen to the Redis instance. Comment out HTTP configuration in the configuration file and add TCP configuration. The HTTP configuration to be commented out are as follows:

```
# {
# log_format main '$remote_addr - $remote_user [$time_local] "$request" '
#                 '$status $body_bytes_sent "$http_referer" '
#                 '"$http_user_agent" "$http_x_forwarded_for"';
#
# access_log /var/log/nginx/access.log main;
#
# sendfile        on;
# tcp_nopush      on;
# tcp_nodelay      on;
# keepalive_timeout 65;
# types_hash_max_size 2048;
#
# include /etc/nginx/mime.types;
# default_type application/octet-stream;
#
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
# include /etc/nginx/conf.d/*.conf;
#}
```

Add the following TCP configuration to the configuration file:

```
stream {
    upstream backend {
        hash $ remote_add r consistent ;
        # Set the connection string and port for
        connecting the backend server to the Redis instance
        . The connection timeout period is 10 seconds and
        the maximum number of attempts is three .
        server r - bplb294374 634044 . redis . rds . aliyuncs .
        com : 6379 max_fails = 3 fail_timeo ut = 10s ;
    }
    server {
        # Nginx connection port
        Listen 3333 ;
    }
}
```

```
# Set the timeout period to 20 seconds for
connecting Nginx to the backend server .
proxy_connect_timeout 20s ;
# If five minutes have passed since the last
successful access ( connection or read / write ) to the
backend server , the connection is timed out and
disconnected .
proxy_timeout 5m ;
# Redirect the TCP connection , data reception ,
and data forwarding to backend server " backend ".
proxy_pass backend ;
}
}
```

For example, the connection string of the Redis instance that you want to connect to is `r - bp1b294374 634044 . redis . rds . aliyuncs . com : 6379` , and the Nginx connection port is 3333, set the TCP configuration as follows:

```
stream{
    upstream backend{
        hash $remote_addr consistent;
        server r-bp1b294374634044.redis.rds.aliyuncs.com:6379 max_fails=3 fail_timeout=10;
    }
    server{
        listen 3333;
        proxy_connect_timeout 20s;
        proxy_timeout 5m;
        proxy_pass backend;
    }
}
```

3. Connect to the Redis instance through the Nginx connection port

After the preceding configuration has been completed, run Nginx to start the Nginx proxy service.

If the EIP of ECS instance where Nginx is deployed is `114 . 55 . 89 . 152` , you can then directly use `redis_cli` to connect to the connection port and test whether the proxy service is running normally.

```
[root@iZ23bsclllgZ ~]# redis-cli -h 114.55.89.152 -p 3333
114.55.89.152:3333> auth
OK
114.55.89.152:3333> ping
PONG
114.55.89.152:3333> get a1
"10"
```

As shown in the preceding figure, you can connect to the Redis instance through the Nginx connection port.

When you configure a migration task in DTS, if the source instance is a VPC-connected Redis instance or an external Redis instances connected to Alibaba Cloud through a leased line, you can directly specify the Nginx proxy connection address.

After DTS starts supporting VPC, no proxy service will be required for VPC-connected Redis instances or external Redis instances connected to Alibaba Cloud through a leased line. For the schedule of enabling DTS to support VPC, see the relevant notification on the Alibaba Cloud website.

Configure a migration task

After all of the prerequisites are met, you can start to configure a migration task. This example describes how to migrate data from an external Redis instance connected to Alibaba Cloud through a leased line to an ApsaraDB for Redis instance in the Classic network .

1. Log on to the [DTS console](#) and click Create Migration Task in the upper-right corner to configure a migration task.

2. Enter the instance connection information

In this step, you must enter the task name, the information for connecting to the external Redis instance, and the information for connecting to the ApsaraDB for Redis instance. The connection information includes the following parameters :

- Task name

DTS automatically generates a task name for each task. Task names are not required to be unique. You can modify the task name as required. We recommend that you choose an appropriate task name for your business.

- Source instance information

Instance type: Select an external Redis instance with an EIP assigned.

Instance region: For an external Redis instance, select a region that is geographically closest to the instance. The closer the region to the Redis instance, the higher the migration performance.

Database type: Select Redis.

Instance mode: The mode is set to standalone by default. Support for Redis clusters will soon be available.

Host name or IP address: The connection string of the external Redis instance. If Nginx proxy is configured, set this parameter to the address that is used to connect to Nginx.

Port: The listening port of the external Redis instance. If Nginx proxy is configured, set this parameter to the Nginx port.

Database password: The password used to connect to the external Redis instance. This parameter is not required. If the external Redis instance does not have a password, this parameter can be left empty.

- Target instance information

Instance type: Redis instance.

Instance region: The region where the ApsaraDB for Redis instance has been created.

Redis instance ID: The ID of the target ApsaraDB for Redis instance to which you want to migrate data.

Database password: The password used to connect to the ApsaraDB for Redis instance.

3. After you have configured the connection information, click **Authorize Whitelist** and **Enter into Next Step** in the lower-right corner to select migration objects.

4. Select migration objects and migration types

In this step, you must configure migration objects and migration types.

- Migration types

For Redis instances, DTS supports standard migration and incremental data migration.

If you want to migrate all of the data, select **Standard Migration** as the migration type.

If you need to migrate data without affecting the write operation on the source instance, you can select **Standard Migration + Incremental Migration** as the migration type.

- Migration objects

In this step, you must select a database to migrate. Currently, Redis only supports whole-database migration. You must select a database instead of selecting keys.

5. Pre-check

Pre-check is required before migration tasks start. A migration task can be started only after the pre-check has been passed. For more information about the pre-check, see [Pre-check](#).

If the pre-check fails, check the details of the failure by clicking the button to the right of the specific check items in the check list, troubleshoot the issues, and perform a pre-check again.

6. Start a migration task

After the pre-check has been passed, you can start the migration task. You can view the migration status and progress in the task list after the task is successfully started.

After the preceding operations are completed, you have successfully configured a migration task to migrate data from an external Redis instance to an ApsaraDB for Redis instance .

Pre-check

Before DTS starts the migration, it will perform a pre-check. This section describes the pre-check items.

Check item	Description	Remarks
Source database connectivity	Checks the connectivity between the DTS server and the external Redis instance.	<ol style="list-style-type: none">1. Checks whether the information you have entered is correct. If the information is incorrect, modify the information and run the pre-check again.2. Checks whether the port allows connections initiated from other servers.
Target database connectivity	Checks the connectivity between the DTS server and the target ApsaraDB for RDS instance.	Checks whether the information you have entered is correct. If it is incorrect, modify the information and run the pre-check again.
One-to-one check of databases	Checks whether data is migrated from multiple databases to one database.	Currently, DTS does not support migrating data from multiple databases to one database. If this occurs, modify the task configuration and run the pre-check again.
Checks whether the target database is empty.	Checks whether the database that you want to migrate data to on the target Redis instance is empty.	If this check fails, delete all keys in the target database and run the pre-check again.
Checks for topological conflicts of incremental migration tasks	Checks whether other incremental migration tasks are running on the target Redis instance.	If this check fails, terminate other incremental migration tasks and run the pre-check again.

6.1.4 Use RDB files to migrate from on-premises Redis to ApsaraDB Redis

The redis-port tool allows you to use RDB files to synchronize the data of an on-premises Redis database to ApsaraDB Redis.

Download redis-port

[Redis-port address](#)

Example

```
./ redis - port restore -- input = x / dump . rdb -- target
= dst_host : dst_port -- auth = dst_passwo rd [-- filterkey
= " str1 | str2 | str3 " ] [-- targetdb = DB ] [-- rewrite ] [--
bigkeysize = SIZE ] [-- logfile = REDISPORT . LOG ]
```

Parameter description

- x/dump.rdb: indicates the dump file path of the user-created Redis database.
- dst_host: indicates the domain name of the ApsaraDB for Redis database.
- dst_port: indicates the port of the ApsaraDB for Redis database.
- dst_password: indicates the password of the ApsaraDB for Redis database.
- str1|str2|str3: filters keys with str1, str2, or str3.
- DB: indicates the database to be synchronized to ApsaraDB for Redis.
- rewrite: overwrites a key that has already been written.
- bigkeysize=SIZE: indicates that when the written value is greater than SIZE, the big key write mode is used.

Check the data synchronization status based on the redis-port log

```
$ ./bin/redis-port restore --input=/... --target=... --auth=...
2018/01/24 10:31:20 [INFO] set ncpu = 24, parallel = 24 filterdb = 0 targetdb = -1
2018/01/24 10:31:20 [INFO] restore from '/home/pan.liangp/run.4.0/dmp_test.rdb' to '10.218.144.110:8179'
2018/01/24 10:31:20 [INFO] Aux information key:redis-ver value:4.0.2
2018/01/24 10:31:20 [INFO] Aux information key:redis-bits value:64
2018/01/24 10:31:20 [INFO] Aux information key:ctime value:1516359975
2018/01/24 10:31:20 [INFO] Aux information key:used-mem value:440742200
2018/01/24 10:31:20 [INFO] Aux information key:repl-stream-db value:0
2018/01/24 10:31:20 [INFO] Aux information key:repl-id value:e19e45132deb3be68a1e97a17fced7108fd1e02
2018/01/24 10:31:20 [INFO] Aux information key:repl-offset value:46308728805
2018/01/24 10:31:20 [INFO] Aux information key:aof-preamble value:0
2018/01/24 10:31:20 [INFO] db.size:2977 expire_size:0
2018/01/24 10:31:21 [INFO] total = 177007551 - 40387120 [ 22%] entry=770
2018/01/24 10:31:22 [INFO] total = 177007551 - 86511868 [ 48%] entry=863
2018/01/24 10:31:23 [INFO] total = 177007551 - 126672277 [ 71%] entry=1472
2018/01/24 10:31:24 [INFO] total = 177007551 - 159105721 [ 89%] entry=1472
2018/01/24 10:31:25 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:26 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] restore: rdb done
```

When `restore : rdb done` appears, data synchronization is completed.

6.1.5 Use AOF to migrate on-premises Redis data to ApsaraDB Redis

The redis-cli tool allows you to use AOF to migrate the data of on-premises Redis databases to ApsaraDB Redis.

Redis-cli is the native command line interface of Redis. ApsaraDB Redis allows you to use redis-cli to import data from existing Redis databases to ApsaraDB for Redis for seamless migration. You can also [import data through DTS](#).

Notes

- Because ApsaraDB Redis supports only access from the Alibaba Cloud intranet, you can perform the following steps only on Alibaba Cloud ECS instances. If your Redis instance is not on an Alibaba Cloud ECS instance, copy the existing AOF to an Alibaba Cloud ECS instance before importing data.
- Redis-cli is the native command line interface of Redis. If you cannot use redis-cli on your ECS instance, download and install Redis before using redis-cli.

Procedure

Perform the following steps if you have created a Redis instance on your Alibaba Cloud ECS instance:

1. Enable the AOF function on the existing Redis instance (skip this step if the AOF function has been enabled).

```
# redis-cli -h old_instance_ip -p old_instance_port config set appendonly yes
```

2. Use AOF to import data to an ApsaraDB Redis instance (assume that the generated AOF is named append.aof).

```
# redis - cli - h aliyun_red is_instanc e_ip - p 6379 - a password -- pipe < appendonly . aof
```



Notice:

If the AOF function does not need to be enabled for the source Redis instance, you can run the following command to disable the function after data is imported:

```
# redis-cli -h old_instance_ip -p old_instance_port config set  
appendonly no
```

6.2 Migrate from ApsaraDB Redis to on-premises Redis

6.2.1 Migrate backup sets

You can use the redis-port tool to migrate ApsaraDB for Redis backup sets to an on-premises Redis database.

Downloading backup set data from the ApsaraDB Redis console

1. Log on to the [ApsaraDB Redis console](#) and locate the target instance on the Instance List page.
2. Click the instance ID, or click the vertical dots in the Action column and choose Manage from the shortcut menu, to go to the Instance Information page.
3. View the number of DB nodes in Architecture Diagram.
4. On the Backup and Recovery page, download backup set data based on the number of DB nodes.

Download redis-port

[Redis-port address](#)

Example

```
./ redis - port      restore      -- input = x / dump . rdb      -- target  
= dst \ _host : dst \ _port      -- auth = dst \ _password      [--  
filterkey =" str1 | str2 | str3 "]" [-- targetdb = DB ] [-- rewrite ]  
[-- bigkeysize = SIZE ] [-- logfile = REDISPORT . LOG ]
```



Note:

You need to perform the recovery procedure once for the backup set of each database.

Parameter description

- x/dump.rdb: indicates the dump file path of an ApsaraDB for Redis backup set.
- dst_host: indicates the domain name or IP address of the user-created Redis database.

- `dst_port`: indicates the port of the user-created Redis database.
- `dst_password`: indicates the password of the user-created Redis database.
- `str1|str2|str3`: filters keys with `str1`, `str2`, or `str3`.
- `DB`: indicates the database to be synchronized to the user-created Redis database.
- `rewrite`: overwrites a key that has already been written.
- `bigkeysize=SIZE`: indicates that when the written value is greater than `SIZE`, the big key write mode is used.

Check the data recovery status based on the redis-port log

```

$ ./bin/redis-port restore --input=/home/pan.liang/run_4.0/dump_test.rdb --target=10.218.144.110:8179 --auth=
2018/01/24 10:31:20 [INFO] set ncpu = 24, parallel = 24 filterdb = 0 targetdb = -1
2018/01/24 10:31:20 [INFO] restore from '/home/pan.liang/run_4.0/dump_test.rdb' to '10.218.144.110:8179'
2018/01/24 10:31:20 [INFO] Aux information key:redis-ver value:4.0.2
2018/01/24 10:31:20 [INFO] Aux information key:redis-bits value:64
2018/01/24 10:31:20 [INFO] Aux information key:ctime value:1516359975
2018/01/24 10:31:20 [INFO] Aux information key:used-mem value:448742200
2018/01/24 10:31:20 [INFO] Aux information key:repl-stream-db value:0
2018/01/24 10:31:20 [INFO] Aux information key:repl-id value:e19e45132deb3be68a1e97a17fced710fd1e02
2018/01/24 10:31:20 [INFO] Aux information key:repl-offset value:46308728805
2018/01/24 10:31:20 [INFO] Aux information key:aof-preamble value:0
2018/01/24 10:31:20 [INFO] db_size:2977 expire_size:0
2018/01/24 10:31:21 [INFO] total = 177007551 - 40387120 [ 22%] entry=770
2018/01/24 10:31:22 [INFO] total = 177007551 - 86511868 [ 48%] entry=863
2018/01/24 10:31:23 [INFO] total = 177007551 - 126672277 [ 71%] entry=1472
2018/01/24 10:31:24 [INFO] total = 177007551 - 159105721 [ 89%] entry=1472
2018/01/24 10:31:25 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:26 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] total = 177007551 - 177007551 [100%] entry=2977
2018/01/24 10:31:27 [INFO] restore: rdb done

```

When `restore : rdb done` appears, data recovery is completed.

6.3 Migrate between ApsaraDB Redis instances

6.3.1 Use Rump to migrate data between ApsaraDB Redis databases

You can use Rump to migrate data between databases within the same ApsaraDB Redis instance or between different ApsaraDB Redis instances.

Prerequisites

- You have created a Linux-based Elastic Compute Service (ECS) instance in the VPC where the target ApsaraDB for Redis instance resides.
- You have downloaded [Rump](#) in the ECS instance you created. You have change the file type of Rump to an executable file.

Context

You will encounter the following problems when migrating Redis data from cloud service providers:

- The data acquisition commands such as `slaveof` and `bgsave` are not supported.

- The `keys` command is likely to cause congestion on the server and affect operating services.

Migration mechanism of Rump

Rump executes a single `scan` command to acquire multiple key lists from the source ApsaraDB for Redis instance. Rump retrieves the key content with the `dump` command, and obtains the expiration time of the keys through the `pttl` command. Then, Rump uses the `restore` command to synchronize the keys to the target ApsaraDB for Redis instance in pipeline mode.

Benefits:

- The `keys` command is replaced with the `scan` command to avoid consequential congestion on the server.
- Data of any type can be synchronized.
- No temporary files are used.
- Channels with buffer are used to improve performance of the source server.
- The pipeline mode is adopted to save network bandwidth.

Procedure

Run the following command in the ECS instance to synchronize data:

```
rump - from source_addr r - fromPwd source_pwd - to  
dest_addr - toPwd dest_pwd [- size size ] [- replace ]
```

Table 6-3: Parameter description

Parameter	Description
source_addr	The address of the source ApsaraDB for Redis instance, in format of <code>redis://host:port/db</code> . <code>host</code> and <code>port</code> are required parameters while <code>db</code> is an optional one. If <code>db</code> is not specified, the value 0 is used by default.
source_pwd	The password of the source ApsaraDB for Redis instance.
dest_addr	The address of the target ApsaraDB for Redis instance, in the same format as <code>source_addr</code> .
dest_pwd	The password of the target ApsaraDB for Redis instance.
size	The number of keys that are synchronized at a time. Default value: 10.

Parameter	Description
-replace	Indicates whether to overwrite an existing key if it is the same as a new key. If this parameter is specified, the existing key is overwritten. If this parameter is not specified, the existing key is not overwritten and an error message is displayed.

- **Example 1: Import the data from db0 to db1 within the same ApsaraDB for Redis instance.**

```
rump - from redis :// r - 123456789 . redis . rds .
aliyuncs . com : 6379 / 0 - fromPwd from_pass - to
redis :// r - 123456789 . redis . rds . aliyuncs . com :
6379 / 1 - toPwd to_pass - size 100
```

- **Example 2: Import the data from db0 of ApsaraDB for Redis instance A to db1 of ApsaraDB for Redis instance B.**

```
rump - from redis :// r - 123456789 . redis . rds .
aliyuncs . com : 6379 / 0 - fromPwd from_pass - to
redis :// r - 999999999 . redis . rds . aliyuncs . com :
6379 / 1 - toPwd to_pass - size 100
```

6.3.2 Use the redis-port tool to migrate data between ApsaraDB Redis instances of different accounts

You can use the redis-port tool to migrate data from one ApsaraDB Redis instance to another instance under a different Alibaba Cloud account.

Prerequisites

- You have created a Linux-based Elastic Compute Service (ECS) instance in the VPC where the target ApsaraDB for Redis instance resides.
- You have downloaded the [redis-port](#) tool in the ECS instance you created.
- You have run the `chmod u + x redis - port` command to change redis-port into an executable file.
- You have run the `mkdir logs` command in the directory where redis-port is located.

Procedure

1. Log on to the [ApsaraDB for Redis console](#).

2. On the Instance List page, locate the source ApsaraDB for Redis instance. Then click the instance ID, or click the vertical dots in the Action column and choose Manage from the shortcut menu.
3. Click Backup and Recovery in the left-side navigation pane.
4. Locate the target backup file in the backup file list, and click Download in the Action column.

**Note:**

To create a backup file instantly, click Create Backup in the upper-right corner of the Backup and Recovery page. In the Backup Instance message box that appears, click Confirm.

5. In the displayed Download Backup File dialog box, click Get URL for Intranet.
6. In the ECS instance, download the backup file from the address copied in the previous step.

**Note:**

For an ApsaraDB for Redis instance of the cluster type, multiple backup files will be generated based on the number of sub-nodes, and you need to download all of the files.

7. Run the following command to import all the backup files to the target database:

```
./ redis - port restore - i backup file name - t domain  
name or IP address of the target database : port  
number -- auth = ' password of the target database '
```

Result

If restore: rdb done appears, data import succeeds. Migration is completed.

7 Application for payment and review

Background

To guarantee service availability, ApsaraDB for Redis provides the SLA indicators for your instance from the last two months. Indicators with a value greater than or equal to 99.95% are normal (highlighted in green). If there are indicators with a value smaller than 99.95% (highlighted in red), you can apply for compensation on the console.

Please refer to the calculation methods and compensation standards for availability of services.

Compensation request time limit

You can apply for the compensation after the fifth working day of every month for the instance that fails to meet the availability standards during the previous month. The application period is limited for up to two months following the month when your ApsaraDB for Redis instance fell short of the availability standard. Application beyond this period isn't accepted.

Procedure

1. Log on to the [Redis console](#).
2. Check the target instance and click Apply for compensation.
3. On the SLA Compensation Management page, submit your application and click Confirm Compensation Application.



Note:

- After submitting your application, you can view the application records on the Applied compensation page.
- If you have any doubt about the compensation amount, click Appeal on the SLA Compensation Management page, or click Apply for Review on the Instance List page to open a ticket to apply for compensation.