ALIBABA CLOUD

Alibaba Cloud

ApsaraDB for Redis

User Guide

Document Version: 20200928

Alibaba Cloud

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice 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.
Warning	A warning notice 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 restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Console logon

Before using ApsaraDB for Redis, you must register an Alibaba Cloud account and log on to the Alibaba Cloud console.

Procedure

1. Sign up with Alibaba Cloud.
2. Log on to the ApsaraDB for Redis console.

For more information about the console, see Alibaba Cloud console.

2.Use a Lua script

ApsaraDB for Redis instances of all editions support Lua commands.

Support for Lua commands

The use of Lua scripts improves the performance of Redis. With the built-in Lua support, Redis is able to perform an efficient check-and-set (CAS) operation on commands. It also allows you to combine and run multiple commands in an efficient manner.

> **Note** If you fail to run an EVAL command and the error message `ERR command eval not support for normal user` is returned, you can upgrade the minor version and try again. For more information, see Upgrade the minor version. During the upgrade, the instance may be disconnected and become read-only within seconds. We recommend that you upgrade the version of an instance during off-peak hours.

Limits on Lua scripts

To ensure that all operations in a Lua script are performed in the same hash slot, the cluster edition of ApsaraDB for Redis sets the following limits on a Lua script:

- The Lua script uses the **redis.call/redis.pcall** function to call Redis commands. For these commands, all the keys must be passed by using the KEYS array, which cannot be replaced by Lua variables. Otherwise, the following error message is returned:

  ```
  -ERR bad lua script for redis cluster, all the keys that the script uses should be passed using the KE
  YS arrayrn
  ```

- All the keys that the script uses must be allocated in the same hash slot. Otherwise, the following error message is returned:

  ```
  -ERR eval/evalsha command keys must be in same slotrn
  ```

- Keys must be passed for all the commands to be called. Otherwise, the following error message is returned:

  ```
  -ERR for redis cluster, eval/evalsha number of keys can't be negative or zerorn
  ```

> **Note** If you can ensure that all operations are performed in the same hash slot in the code and want to break the Lua limits of Redis Cluster, you can set the script_check_enable parameter to *0* in the console. Then, the system does not check the Lua script at the backend. For more information about how to set this parameter, see Parameter overview and configuration guide.

3.Manage instances

3.1. Lifecycle management

3.1.1. Change specifications

ApsaraDB for Redis supports the subscription and pay-as-you-go billing methods. You can change the billing method from pay-as-you-go to subscription. Both billing methods allow you to modify the specifications of an instance.

Background

Both the pay-as-you-go and subscription billing methods allow you to upgrade or downgrade instances at any time. You can change the architecture type of an instance to either cluster architecture or non-cluster architecture. To change the architecture type and downgrade an instance, you must change the architecture type before you downgrade the specifications. For example, to change an 8 GB standard instance to a 4 GB cluster instance, you must first change the instance type of the standard instance to a cluster instance. Then, downgrade the capacity of the cluster instance from 8 GB to 4 GB.

> **Notice** After you change a non-cluster instance to a cluster instance or change a cluster instance to a non-cluster instance, the original alert settings become invalid. You must reconfigure the alert settings. For more information, see Alert settings.

The duration of a specification change depends on various factors such as network conditions, task queue size, and data volume. We recommend that you change specifications during off-peak hours and make sure that your applications have automatic reconnection mechanisms.

Pricing

For more information, see Instance scaling.

Impacts

- During the specification change, the instance may be disconnected for less than 30 seconds once or twice.

  > **Note** For the Luttece client, the instance may be disconnected for 2 to 10 minutes.

- To accelerate the synchronization of incremental data between the new instance and original instance and avoid data dual-writing caused by DNS cache, the instance becomes read-only for less than 1 minute during the specifications change. This ensures data consistency between the new instance and original instance.
- When you change specifications, if you set **Switching Time** to **Switching during serviceable time**, the instance will be changed when the time arrives. For more information, see Set a maintenance window.

Procedure (pay-as-you-go)

1. Log on to the ApsaraDB for Redis console.

2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the upper-right corner of the page, click **Change Configurations** .
5. On the **Update** page, modify the specifications and click **Activate**.

   If the specifications are changed, a message appears to notify you that the operation is successful. Then, this pay-as-you-go instance is billed based on the new specifications in the current billing cycle.

Procedure (subscription)

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the upper-right corner of the page, click **Upgrade** or **Downgrade**.

> **Note** Higher specifications are charged more than lower specifications. For more information, see **Upgrade** and **Downgrade**. For example, the price of an 8 GB read/write splitting instance with five read replicas is higher than that of a 16 GB cluster instance. If you want to change a 16 GB cluster instance to an 8 GB read/write splitting instance with five read replicas, you must upgrade the instance.

5. On the **Update** page, specify the specifications and click **Pay**.
6. Complete the payment.

Related API operations

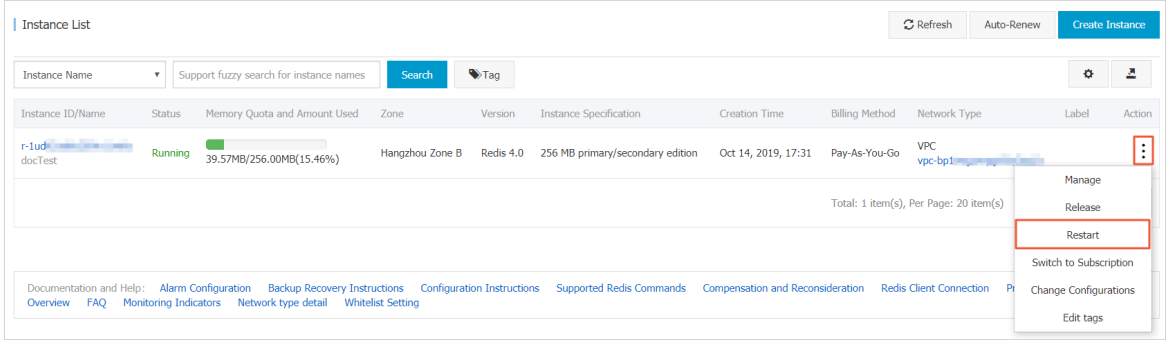
ModifyInstanceSpec

3.1.2. Restart an instance

You can restart an instance in the instance list of the Redis console.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select the region where the instance resides.
3. On the **Instance List** page, find the target instance and click **Restart** in the **Action** column.
4. Click **Restart**.



> **Notice** During the restart, the instance may be disconnected within seconds. We recommend that you restart an instance during off-peak hours and ensure that your application supports automatic reconnection.

5. In **restart** dialog box that appears, select whether to upgrade the minor version, select a restart time, and then click **confirm**.
   - By default, the minor version of an instance is upgraded to the latest when the instance is restarted. If you do not want to upgrade the minor version, clear **upgrade the minor version**.
   - You can choose from two restart times:
     - **Restart immediately** to restart the instance immediately.
     - **Restart in maintenance time window** in the available maintenance window, restart the instance. For more information, see Set a maintenance window.

3.1.3. Switch to subscription

After you purchase a Pay-As-You-Go instance, you can change its billing method to subscription as needed.

Precautions

- You cannot change the billing method of a subscription instance to Pay-As-You-Go. To maximize resource usage, we recommend that you evaluate your usage model carefully before you change the billing method of an instance.
- Within the contract period, you cannot directly release a subscription instance.
- After the billing method of an instance is changed to subscription, the instance is immediately billed in subscription mode.
- When you change the billing method of a Pay-As-You-Go instance to subscription, the system generates a purchase order. The changed billing method takes effect only after you pay for this order. If you do not pay or fail to pay, an unpaid order is listed on the Order management page of your Alibaba Cloud account. In this case, you cannot purchase any instances or change the billing method of any instances.

> **Note**
>
> - If you have an unpaid order for changing the billing method of a Pay-As-You-Go instance to subscription and you upgrade the configuration of this Pay-As-You-Go instance, the amount of the unpaid order is insufficient to cover the changed billing method due to the changed instance components. In this case, the system forbids you to pay for this order. You must void this unpaid order and change the billing method of the instance again.
> - If you want to cancel an unpaid order, you can void it on the Order management page of the console.

Prerequisites

- The billing method of an instance is Pay-As-You-Go. The instance is in the **Running** status.

> **Note** Before you pay for a purchase order for changing the billing method of a Pay-As-You-Go instance to subscription, if the status of this instance changes (for example, to **Locked**), you may fail to pay for this order. You can continue to pay only after the instance status changes to Running.

- The instance has no unpaid order for changing the billing method.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the Instance List page, find the target instance and click **Switch to Subscription** in the **Action** column.
4. Adjust the slider of **Duration** to select a subscription period.
5. Click **Confirm** and pay for the generated order as prompted.

Related API operations

API	Description
TransformToPrePaid	You can call this operation to change the billing method of an ApsaraDB for Redis instance from Pay-As-You-Go to subscription.

3.1.4. Renew an instance

After a subscription instance expires, you must renew the instance within days after the expiration to continue the use of the instance. To avoid service interruption caused by an expired subscription, we recommend that you manually renew the instance or enable auto-renewal before the instance expires.

Prerequisites

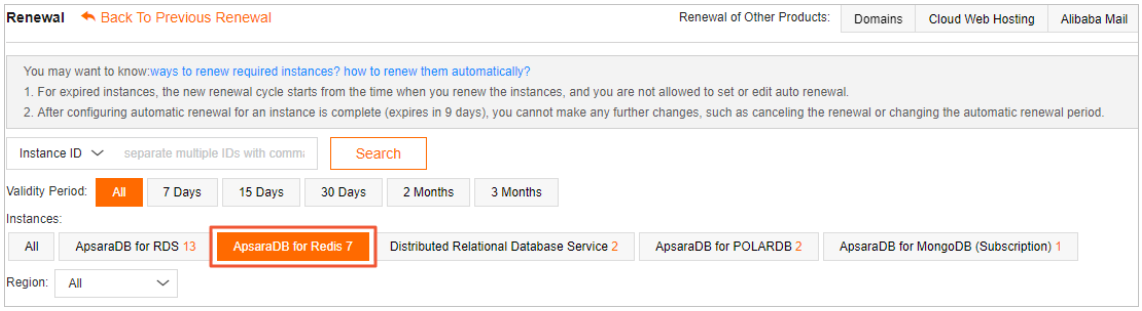
The billing method of the instance is subscription.

Overdue payment rules

For more information about expiration, overdue payments, and renewal rules, see Expiration, overdue payments, and renewal.

Procedure (auto-renewal)

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, choose **Expenses > Renew Management**.
3. Perform the following steps for different versions of the **Renew Management** page:
   - The latest version of Renew Management: Click **ApsaraDB for Redis** in the Instances field.



- Earlier version of Renew Management: In the left-side navigation pane, click **ApsaraDB for Redis**.

4. The **Manually Renew** tab appears. On this tab, find the instance that you want to renew and click **Enable Auto Renewal** in the **Actions** column for the instance.
5. In the dialog box that appears, read the prompt and specify a required auto-renewal cycle.

> **Note** After you enable auto-renewal, the system will automatically renew the instance based on the auto renewal cycle that you specify. For example, if you specify a three-month auto-renewal cycle, you will be charged a three-month subscription fee upon each auto-renewal.

6. Click **Auto Renew**.

Procedure (manual renewal)

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the Instances page, find the instance that you want to renew, and in the **Actions** column, choose > **Renew**.
4. On the **Renew** page, select a renewal duration.
5. Read and accept the ApsaraDB for Redis Agreement of Service and then click **Pay**.
6. Complete the payment.

Related API operations

API	Description
ModifyInstanceAutoRenewalAttribute	Enables or disables auto-renewal for ApsaraDB for Redis instances.
DescribeInstanceAutoRenewalAttribute	Queries the auto-renewal status of an ApsaraDB for Redis instance.

3.1.5. Upgrade the major version

You can upgrade the engine version, which is the major version, of an ApsaraDB for Redis instance with one click in the console. For example, you can upgrade the major version from Redis 2.8 to Redis 4.0.

Prerequisites

The engine version of an instance is not the latest.

Context

ApsaraDB for Redis provides some features only for specific engine versions. If the engine version of your instance is too earlier, you can follow the procedure in this topic to upgrade the version.

The method and time required for upgrading the major version vary with the architecture of an instance:

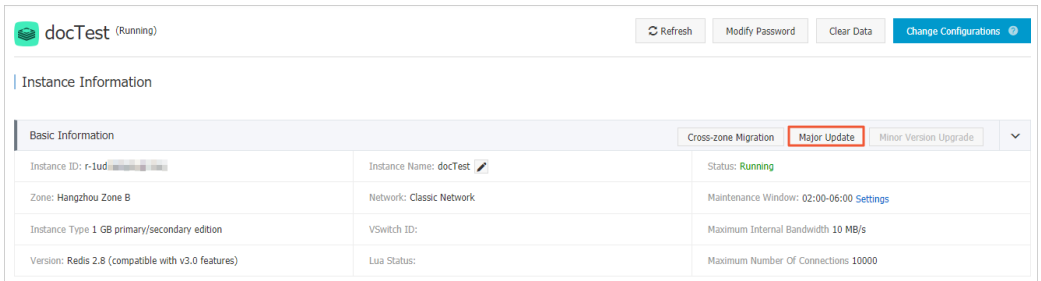
- For the cluster edition, read/write splitting edition, or standard disaster recovery edition, the instance is upgraded during cross-server migration. The upgrade duration depends on the data volume. The instance may be disconnected within 30 seconds and become read-only within 60 seconds.
- For the standard non-disaster recovery edition, the instance is upgraded on the local server. The upgrade takes effect within 5 minutes and has no impact on the ApsaraDB for Redis service. If the resources of the local server are insufficient, cross-server migration is required. The impact is the same as that for the cluster edition.

> **Note** We recommend that you upgrade the version of an instance during off-peak hours and ensure that your application supports automatic reconnection.

Procedure

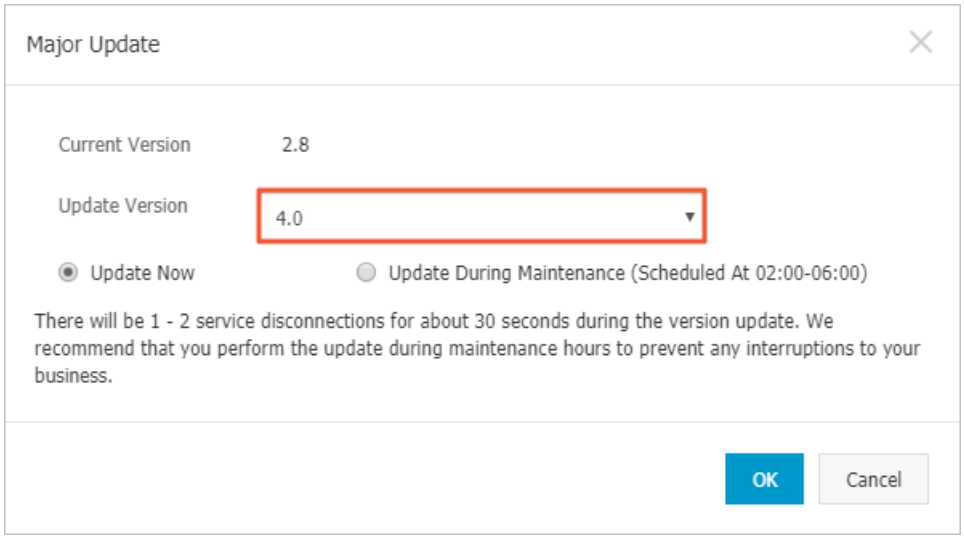
1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, click the target instance ID or **Manage** in the **Action** column for the target instance.
4. On the **Instance Information** page, click **Major Update** in the upper-right corner of the **Basic Information** section.

   Upgrade the major version

5. In the **Major Update** dialog box, select the target version.

   Select the target major version



6. Select **Update Now** or **Update During Maintenance** as required, and then click **OK**.

> **Note** We recommend that you set the maintenance window to a period during off-peak hours and select **Update During Maintenance** to minimize the impact on normal business during the upgrade.

Related API operations

API	Description
ModifyInstanceMajorVersion	Call this OpenAPI to upgrade the major version of a Redis instance.

3.1.6. Upgrade the minor version

Alibaba Cloud optimizes the kernel of ApsaraDB for Redis in a continuous way to fix security vulnerabilities and improve service stability. You can upgrade the kernel version (minor version) of an ApsaraDB for Redis instance with a few clicks in the console.

Background

The method and time required for minor version upgrades vary based on the architecture of an instance:

- For cluster instances, read/write splitting instances, and standard instances, use cross-server migration to upgrade the kernel versions of the instances. Standard instances must be deployed in the zones that support zone-disaster recovery. The upgrade duration depends on the amount of data. The instance may be disconnected within 30 seconds and become read-only within 60 seconds.
- If your standard instances run in the zones that do not support zone-disaster recovery, use the local servers to upgrade the kernel versions of the instances. In most cases, an upgrade starts within 5 minutes after you confirm the upgrade. The upgrade duration depends on the data volume. Your instances may remain disconnected for less than 30 seconds. During a period of less than 60 seconds, you may have only read access to your instances. If the resources of your local servers are insufficient, cross-server migration is required to upgrade

the kernel version. In this scenario, the impact of the upgrade is the same as that for the instances in the cluster edition.

> Notice
>
> - We recommend that you upgrade the kernel version of an instance during off-peak hours and ensure that your application supports automatic reconnection.
> - The system automatically checks the kernel version of an instance. If the latest version is used, the **Minor Version Upgrade** button is unavailable and displayed in gray for the instance. You can find this button in the upper-right corner of the **Basic Information** section on the Instance Information page of the ApsaraDB for Redis console.
> - If an ApsaraDB for Redis instance is created in a zone that supports zone-disaster recovery, such as **China (Hangzhou) Zone (B+F)**, the instance uses the zone-disaster recovery architecture. For more information, see Zone-disaster recovery.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. On the **Instance Information** page, click **Minor Version Upgrade**.
5. In the **Minor Version Upgrade** dialog box that appears, specify the upgrade period of time.
   - **Upgrade Now**: upgrade the minor version immediately.
   - **Upgrade During Maintenance**: you can upgrade the minor version within the maintenance window. For more information, see Set a maintenance window.

   > Note During the upgrade process, your ApsaraDB for Redis instance may be disconnected for less than 30 seconds and be read-only for less than 60 seconds. We recommend that you upgrade your instance during off-peak hours.
6. Click **OK**.

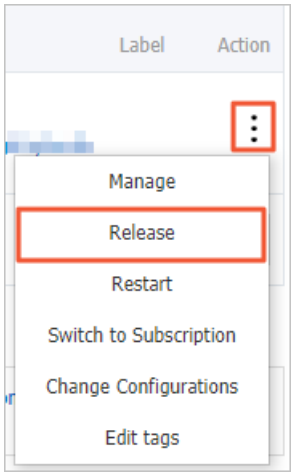
3.1.7. Release an instance

You can release a pay-as-you-go instance at any time.

> Note You cannot delete or release a subscription instance.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, find the target instance and click **Release** in the **Action** column.



Warning Make sure you have backed up the instance if the data stored in it is still useful. To back up an instance, see Back up and restore data in the console.

4. In the **Release Instance** dialog box, click **OK**.

Related API operations

DeleteInstance

3.2. Network connection management

3.2.1. Switch to VPC network

This topic describes how to switch Redis instances that are deployed in classic networks to Virtual Private Cloud (VPC) networks.

Background

ApsaraDB supports two types of networks: the classic network and VPC network. The differences between the classic network and VPC network are as follows:

- Classic network: Cloud services in a classic network are not isolated. Unauthorized access to a cloud service is blocked only by a security group or a whitelist policy of the service.
- VPC: You can use a VPC network to build an isolated network environment in Alibaba Cloud. You can customize the route table, IP address range, and gateway for a VPC network. You can also use a physical connection or Virtual Private Network (VPN) to combine your on-premises data center with cloud resources in Alibaba Cloud VPC networks to build a virtual data center and smoothly migrate your applications to the cloud.

Precautions

- You can switch the network type from the classic network to VPC network, but not conversely.
- When you switch the network type from the classic network to VPC network, you can choose to keep the connection address of the classic network.

Prerequisites

The current network type of an ApsaraDB for Redis instance is the classic network. A VPC and a VSwitch are created in the region where the ApsaraDB for Redis instance is deployed. For more information, see Create a VPC.

> **Notice** You must create a VSwitch in the zone where the target ApsaraDB for Redis instance is deployed.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. On the **Instance Information** page, click **Switch to VPC Network**.
5. In the **Switch to VPC Network** dialog box, select a **VPC** and a **VSwitch**, select an option for **Retain the connection address of the classic network**, select a retention period for **Retention Days**, and then click **OK**.

   > **Notice**
   > - You can click **Refresh** on the **Instance Information** page to view the endpoints of the classic network and VPC network.
   > - If **OK** is disabled, you need to check whether you have selected a **VSwitch**. If no VSwitch is available in the current VPC network, you need to create a VSwitch. For more information, see **VPC User Guide > VSwitch management**.

Troubleshooting

For more information about how to fix issues that may occur when you connect to an ApsaraDB for Redis instance, see Troubleshooting for connection issues in ApsaraDB for Redis.

Related operations

SwitchNetwork

3.2.2. Modify endpoints

ApsaraDB for Redis allows you to modify internal and public endpoints for instances. When changing the ApsaraDB for Redis instance, you can change the endpoint of the new instance to the endpoint of the original instance without the need to modify the application.

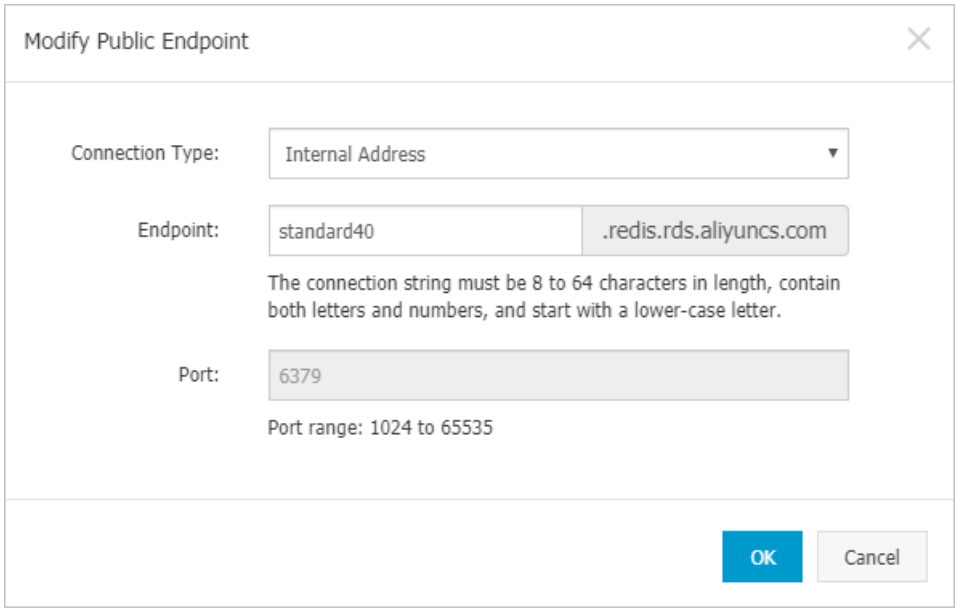
Prerequisites

The instance is running properly.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the homepage, select the region where the instance is located.
3. On the **Instance Information** page, click the instance ID, or click **Manage** in the **Actions** column corresponding to the instance.

4. On the **Instance Information** page, find the **Connection Information** section and click **Modify Public Endpoint**.
5. In the **Modify Public Endpoint** dialog box that appears, set **Connection Type** and **Endpoint**. Click **OK**.



The format of the endpoint is `<Prefix>.redis.rds.aliyuncs.com`. The default prefix of the internal endpoint is the instance ID. The custom prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.

Related operations

Operation	Description
ModifyDBInstanceConnectionString	Call this API to modify the endpoints of an ApsaraDB for Redis instance.

3.2.3. Apply for public endpoints

ApsaraDB for Redis provides an internal endpoint by default. To access an ApsaraDB for Redis instance over the Internet, apply for a public endpoint.

For more information, see Through the Internet.

3.2.4. Enable a direct connection

This topic describes how to connect to a cluster instance of ApsaraDB for Redis through a direct connection. By default, cluster instances can only be connected through a proxy server. If you want to bypass the proxy server, you can apply for a private endpoint to enable direct connections to a cluster instance. This reduces the number of connections and improves service performance.

Prerequisites

- The ApsaraDB for Redis cluster instance is used.

- The engine version of the cluster instance is Redis 4.0 (Community Edition) or Redis 5.0 (Community Edition and Enterprise Edition).
- The instance runs in a Virtual Private Cloud (VPC) network.
- SSL Certificate is disabled.
- The number of available virtual IP addresses (VIPs) in the VPC network is greater than or equal to the number of data shards plus one.

> **Note** For example, if the cluster contains eight data shards, the VPC network must provide nine or more VIPs. Otherwise, direct connections cannot be enabled.

Context

After you enable direct connection, you can connect to your cluster instance by using a private endpoint. To enable a direct connection, set the connection string on your client to the private endpoint of your cluster instance. Then, you can bypass the proxy server and access the data shards from your client. Compared with the proxy mode, the direct connection mode helps reduce the response time of ApsaraDB for Redis because requests do not need to pass through a proxy server.

Precautions

If you enable direct connections, the following operations are not allowed:

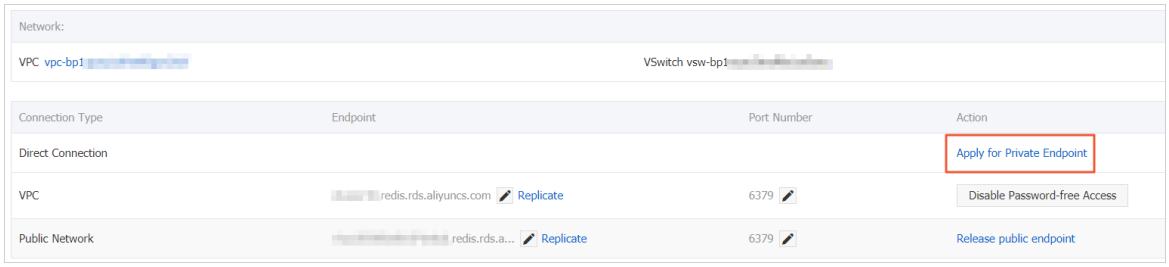
- Change specifications
- Upgrade the major version
- Migrate an instance across zones

To perform these operations, you can Release a private endpoint. After you complete these operations, you can apply for a private endpoint again.

> **Note** These operations will be supported in the direct connection mode of later versions.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the Instances page, click **Instance ID**, or choose More > **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **Connection** in the left-side navigation pane.
5. On the **Connection** page, click **Apply for Private Endpoint** in the **Actions** column of **Direct Connection**.



6. In the **Apply for Private Endpoint** dialog box that appears, follow these steps:
   i. Modify the prefix of the private endpoint. An endpoint of ApsaraDB for Redis follows this format: `<prefix>.redis.rds.aliyuncs.com`. The custom prefix of the endpoint must be 8 to 64 characters in length and can contain only lowercase letters and digits. It must start with a letter.
   ii. Specify a port number. The port number must be 1024 to 65535.
   iii. Click **OK**.

FAQ

- Q: Is the ApsaraDB for Redis service disrupted when I enable direct connection?

  A: No. Enabling direct connection does not cause service disruption.
- Q: Can I connect to an instance by using a direct connection when the proxy mode is enabled?

  A: Yes.
- Q: My instance meets the requirements of the prerequisites. What can I do if I cannot see Apply for Private Endpoint in the Actions column?

  A: You can upgrade the minor version of your instance to the latest one. For more information, see Upgrade the minor version.

What's next

- Add the client IP address to the whitelist of the instance. For more information, see Set IP address whitelists.
- Use a private endpoint to connect to an ApsaraDB for Redis instance.

3.2.5. Release a private endpoint

If you no longer use a private endpoint or want to perform operations that are not supported in direct connection mode, you can release the private endpoint to disable the direct connection mode. For example, release the private endpoint before you change configurations or upgrade a major version.

Prerequisites

- A private endpoint is available. For more information, see Apply for a private endpoint.
- The private endpoint configured in the application is changed to another available endpoint. For example, the internal endpoint in proxy mode.

> **Warning** After a private endpoint is released, the client cannot use it to connect to the ApsaraDB for Redis cluster. We recommend that you change the connection configuration in your application before you release the private endpoint.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.

4. In the left-side navigation pane, click **Connection**.
5. On the **Connection** page, click **Release Private Endpoint** in the **Actions** column of **Direct Connection**.

6. In the **Release Private Endpoint** dialog box, click **OK**.

3.2.6. Modify the port for the endpoint

You can customize a port in the range of 1024 to 65535 for an Apsaradb for Redis instance.

Prerequisites

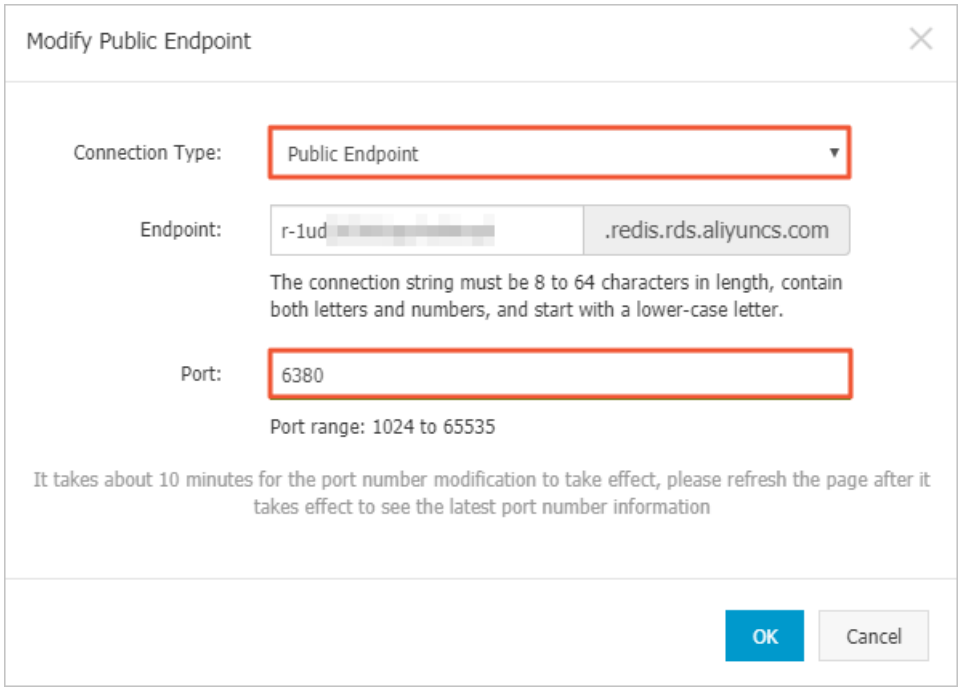
The Apsaradb for Redis instance is in the running state.

Modification impact

After the port is modified, you need to use the new port when you access the ApsaraDB for Redis instance. Make sure that the setting of the application is modified accordingly.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select the region where the instance is located.
3. On the **Instance List** page, click the instance ID.
4. On the **Instance Information** page, click **Modify Public Endpoint** in the upper-right corner of the **Connection Information** section.
5. In the **Modify Public Endpoint** dialog box, complete the following operations.
   i. Select a **Connection Type** according to your scenario.
   ii. Modify the port number.
   iii. Click **OK**.



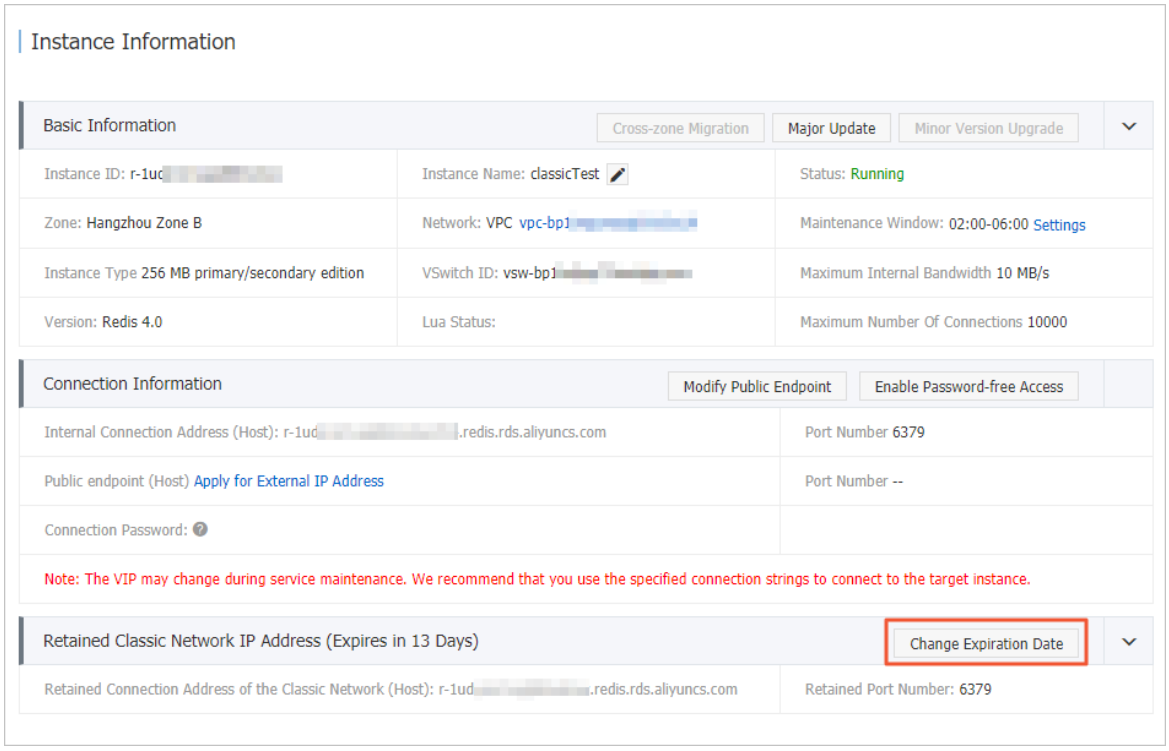
3.2.7. Change the expiration time for the connection address of a classic network

After you set the retention period for the connection address of a classic network, you can change the expiration time in the console to extend the retention period before the address expires.

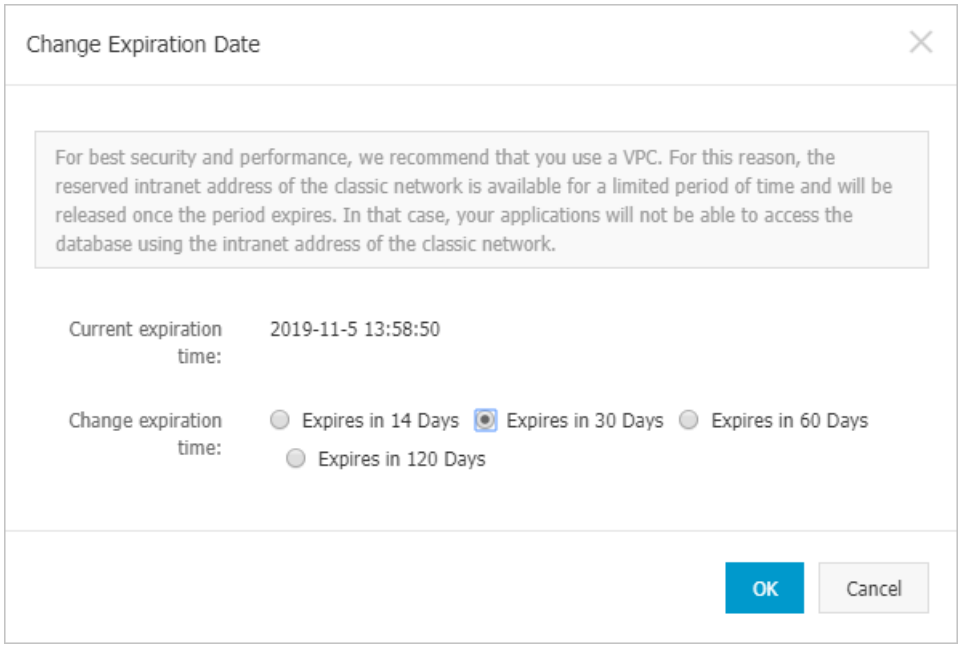
During the period in which your instance can be connected over a classic network or Virtual Private Network (VPC), you can specify the expiration time for the connection address of the classic network to fit your requirements. The setting takes effect immediately. For example, if the connection address of the classic network is about to expire on August 18, 2017 and you change the expiration time to 14 days later on August 15, 2017, the connection address of the classic network is released on August 29, 2017.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. On the **Instances** page, find the target instance and click **Manage** in the Actions column to go to the **Instance Information** page.
4. In the **Retained Classic Network IP Address** section, click **Change Expiration Date**.



5. In the dialog box that appears, select a new expiration time and click **OK**.



Note You can change the expiration time multiple times.

Related operations

API	Description
ModifyInstanceNetExpireTime	Modifies the retention period of the connection address of the classic network.

3.2.8. Release public endpoints

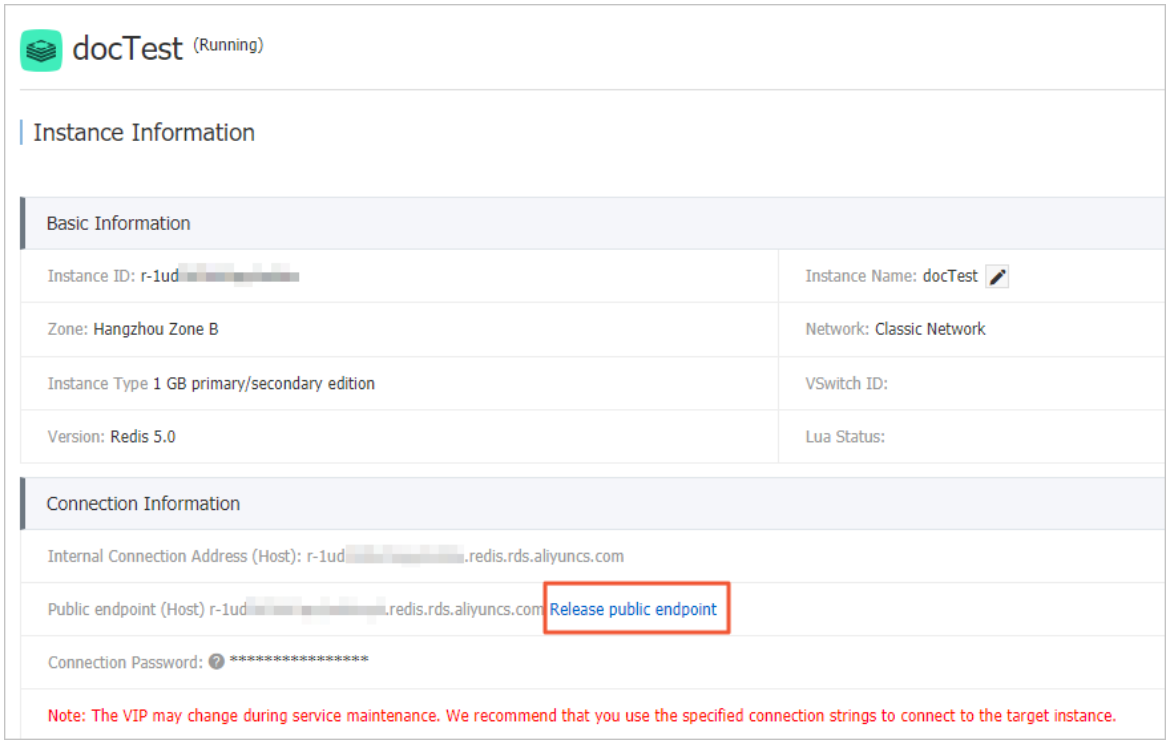
You can release public endpoints that you no longer need.

Prerequisites

The public endpoint is available.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the homepage, select the region where the instance is located.
3. On the **Instance Information** page, click the instance ID, or click **Manage** in the **Action** column corresponding to the instance.
4. In the **Connection Information** section, click **Release public endpoint**.



5. In the **Release public endpoint** dialog box that appears, click **OK**.

3.2.9. Release classic network endpoints

You can release the classic network endpoint that is retained when you migrate an ApsaraDB for Redis instance from a classic network to a virtual private cloud (VPC) . This topic describes how to release a classic network endpoint. Before you migrate an ApsaraDB for Redis instance to another zone, you must release the classic network endpoint.

Prerequisites

Your ApsaraDB for Redis instance is migrated from a classic network to a VPC and the classic network endpoint is retained. For more information, see Switch to VPC network.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. Use either of the following methods to open the **Change Expiration Date** dialog box:
   - On the **Instance Information** page, click **Change Expiration Date** in the **Retained Classic Network IP Address** section.
   - a. In the left-side navigation pane of the **Instance Information** page, click **Connection**.
     b. On the **Connection** page, click **Change Expiration Date** next to **Classic Network**.
5. In the **Change Expiration Date** dialog box, select **Release Now**.

   Warning The classic network endpoint becomes unavailable immediately after you release it. Before you release it, change the endpoint of the ApsaraDB for Redis instance in your application to a VPC endpoint to ensure service availability.

6. Click **OK**.

3.3. System Parameters

3.3.1. Parameter overview and configuration guide

ApsaraDB for Redis allows you to customize instance parameters. This topic describes the parameters of different engine versions and the common methods to specify these parameters in the ApsaraDB for Redis console.

Custom parameters of Redis 5.0

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
#no_loose_audit-read-enabled	Specifies whether to enable read request audit. After you enable this feature, logs of read requests are displayed in audit logs.	Not supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
#no_loose_check-whitelist-always	Specifies whether to verify that the client IP address is in a whitelist of the ApsaraDB for Redis instance if password-free access is enabled in a Virtual Private Cloud (VPC) network. If you set this parameter to yes, the whitelist still takes effect for password-free access over a VPC network. Valid values: • yes: enables the system to check whether a client IP address is in a whitelist. • no: disables the system to check whether a client IP address is in a whitelist. This is the default value.	Supported	Supported	Supported
#no_loose_disabled-commands	Specifies the commands that you want to disable, including **FLUSHALL**, **FLUSHDB**, **KEYS**, **HGETALL**, **EVAL**, **EVALSHA**, and **SCRIPT**. Separate multiple commands with commas (,).	Supported	Supported	Supported
#no_loose_ssl-enabled	Specifies whether to enable SSL encryption. Valid values: • yes: enables SSL encryption. • no: disables SSL encryption. This is the default value.	Not supported	Supported	Supported
#no_loose_sentinel-enabled	Specifies whether to enable the compatibility with the Sentinel mode. Valid values: • yes: enables the compatibility with the Sentinel mode. • no: disables the compatibility with the Sentinel mode. This is the default value.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
read_request_only_ronode_whenrwsplit_enable	Specifies whether to enable directional forwarding for requests from accounts that have read-only permissions. Valid values: • 1: enables directional forwarding. Requests from accounts that have read-only permissions are only forwarded to read replicas. • 0: disables directional forwarding. Requests from accounts that have read-only permissions are forwarded all nodes including the primary node based on weights. This is the default value.	Not supported	Not supported	Supported
appendonly	Specifies whether to enable append-only file (AOF) persistence. Valid values: • yes: enables AOF persistence. This is the default value. • no: disables AOF persistence. Note If you set this parameter to no, AOF persistence is disabled on the master node and remains functional on the replica node.	Supported	Supported	Supported
client-output-buffer-limit pubsub	Specifies output buffer limits of publisher and subscriber clients. The clients are disconnected when the specified limits are reached. Specify this parameter in the following format: `<hard limit> <soft limit> <soft seconds>`. • hard limit: disconnects a client if the output buffer of the client is larger than or equal to the value specified by the hard limit field. Unit: bytes. • soft limit and soft seconds: disconnects a client if the output buffer of the client is larger than or equal to the value specified by the soft limit field and if this situation lasts for a period longer than or equal to the value specified by the soft seconds field. The soft limit value is measured in bytes, and the soft seconds value is measured in seconds.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
dynamic-hz	Specifies whether to enable a dynamic hz value. Valid values: • yes: enables a dynamic hz value. This is the default value. • no: disables a dynamic hz value.	Supported	Not supported	Not supported
hash-max-ziplist-entries	The upper limit of the number of bytes in each key-value pair stored in a hash to support ziplist encoding. The ziplist data structure is used only if both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the hash is less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored in the hash is less than the value of the hash-max-ziplist-entries parameter.	Supported	Supported	Supported
hash-max-ziplist-value	The upper limit of the number of bytes in each key-value pair stored in a hash to support ziplist encoding. The ziplist data structure is used only if both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the hash is less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored in the hash is less than the value of the hash-max-ziplist-entries parameter.	Supported	Supported	Supported
hz	Specifies the frequency at which Redis background tasks are performed. For example, to evict expired keys. You can specify a value from 1 to 500 as the task frequency. The default value is 10. A higher value results in higher CPU consumption but allows the system to delete expired keys and terminate timeout connections more frequently. We recommend that you specify a value smaller than or equal to 100.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
lazyfree-lazy-eviction	Specifies whether to use Lazyfree to evict data. Valid values: • yes: enables Lazyfree to evict data. • no: disables Lazyfree to evict data. This is the default value.	Supported	Supported	Supported
lazyfree-lazy-expire	Specifies whether to enable Lazyfree to delete expired keys. Valid values: • yes: enables Lazyfree to delete expired keys. This is the default value. • no: disables Lazyfree to delete expired keys.	Supported	Supported	Supported
lazyfree-lazy-server-del	Specifies whether to run Lazyfree to delete data asynchronously when running the **DEL** command. Valid values: • yes: enables asynchronous deletion. This is the default value. • yes: disables asynchronous deletion. This is the default value.	Supported	Supported	Supported
list-compress-depth	Specifies the number of entries that are not compressed at both ends of a list. Valid values: 0 to 65535. • 0: does not compress any node of the list. This is the default value. • 1: specifies that the first node from each end of the list is not compressed, but all nodes between these two nodes are compressed. • 2: specifies that the first two nodes from each end of the list are not compressed, but all nodes between these four nodes are compressed. • 3: specifies that the first three nodes from each end of the list are not compressed, but all nodes between these six nodes are compressed. • You can specify other values based on the same rule.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance

list-max-ziplist-size	• The maximum length of a ziplist in a quicklist. A positive value specifies the maximum number of elements in each ziplist of a quicklist. For example, if you set this parameter to 5, each ziplist of a quicklist can contain a maximum of five elements.
	• A negative number indicates the maximum number of bytes in each ziplist of a quicklist. Valid values: [-5, -1], where:
	  ◦ -5: specifies that each ziplist of a quicklist cannot exceed 64 KB (1 KB = 1,024 bytes).
	  ◦ -4: specifies that each ziplist of a quicklist cannot exceed 32 KB.
	  ◦ -3: specifies that each ziplist of a quicklist cannot exceed 16 KB.
	  ◦ -2: specifies that each ziplist of a quicklist cannot exceed 8 KB. This is the default value.
	  ◦ -1: specifies that each ziplist of a quicklist cannot exceed 4 KB.
		Supported	Supported	Supported

maxmemory-policy	Specifies the policy used to evict data when the system runs out of memory. Valid values: • volatile-lru: evicts the approximated least recently used (LRU) keys among keys that have time-to-live (TTL) values configured. • allkeys-lru: evicts the approximated LRU keys. • volatile-lfu: evicts the approximated least frequently used (LFU) keys among keys that have TTL values configured. • allkeys-lfu: evicts the approximated LFU keys. • volatile-random: evicts random keys among keys that have TTL values configured. • allkeys-random: evicts random keys. • volatile-ttl: evicts keys with the minimum TTL. The LRU, LFU, and volatile-ttl policies use approximated randomized algorithms. • noeviction: specifies that the system does not evict any keys, but returns an error for write operations.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
notify-keyspace-events	The types of events of which the Redis server can notify clients. The value of this parameter is any combination of the following characters, • each of which specifies a type of events: • K: keyspace events, published with the __keyspace@<db>__ prefix. • E: key events, published with the __keyevent@<db>__ prefix. • g: generic events that are not related to any specific commands, such as DEL, EXPIRE, and RENAME. • l: events of list commands. • s: events of set commands. • h: events of hash commands. • z: event of sorted set commands. • x: events of expired keys. An expiration event is triggered when an expired key is deleted. • e: the eviction events. An eviction event is triggered when a key is deleted due to the policy specified by the maxmemory-policy parameter. • A: the alias for g$lshzxe.	Supported	Not supported	Not supported
set-max-intset-entries	The maximum number of data entries in a set to support inset encoding. A set uses inset encoding when both of the following conditions are met: 1. The number of entries in the set is less than or equal to the value of the set-max-intset-entries parameter. 2. The set only contains radix-10 integers in the range of 64-bit signed integers.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
slowlog-log-slower-than	Specifies whether to log slow queries. • Negative value: does not log slow queries. • 0: logs all queries. • Positive value: logs queries of which the duration exceeds the specified value. Valid values: 0 to 10000000. Default value: 10000. Unit: microseconds.	Supported	Supported	Supported
slowlog-max-len	The maximum number of slow query logs that can be stored. Valid values: 100 to 10000. Default value: 1024.	Supported	Supported	Supported
stream-node-max-bytes	The maximum amount of memory in bytes that each macro node in a stream can consume. Valid values: 0 to 999999999999999. Unit: bytes. A value of 0 specifies that the amount of consumed memory is not limited.	Supported	Not supported	Not supported
stream-node-max-entries	The maximum number of entries stored on each macro node in a stream. Valid values: 0 to 999999999999999. A value of 0 specifies that the number of entries on each node is not limited.	Supported	Not supported	Not supported
timeout	Specifies a timeout period. The system terminates a connection to a client if the client has been idle for the specified period of time. Valid values: 0 to 100000. Unit: seconds. A value of 0 specifies that no timeout period is specified for connections.	Supported	Not supported	Not supported
zset-max-ziplist-entries	The upper limit of the number of key-value pairs stored in a sorted set to support ziplist encoding. A sorted set uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the sorted set is less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored in the sorted set is less than the value of the zset-max-ziplist-entries parameter.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
zset-max-ziplist-value	The upper limit of the number of bytes in each key-value pair stored in a sorted set to support ziplist encoding. A sorted set uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the sorted set is less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored in the sorted set is less than the value of the zset-max-ziplist-entries parameter.	Supported	Supported	Supported
list-max-ziplist-entries	The upper limit of the number of bytes in each element stored in a list to support ziplist encoding. A list uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each element stored in the list is less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored in the list is less than the value of the list-max-ziplist-entries parameter.	Not supported	Not supported	Not supported
list-max-ziplist-value	The upper limit of the number of bytes in each element stored in a list to support ziplist encoding. A list uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each element stored in the list is less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored in the list is less than the value of the list-max-ziplist-entries parameter.	Not supported	Not supported	Not supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
cluster_compat_enable	Specifies whether to enable the compatibility with the native Redis cluster syntax. Valid values: • 0: disables the compatibility mode. • 1: enables the compatibility mode. This is the default value.	Not supported	Supported	Supported
script_check_enable	Specifies whether to verify that the keys used in Lua scripts are mapped to the same slot. Valid values: • 0: does not check whether the keys are mapped to the same slot. • 1: checks whether the keys are mapped to the same slot. This is the default value.	Not supported	Supported	Supported
readonly_lua_route_rnode_enable	Specifies whether to enable Lua scripting on read replicas. Valid values: • 0: disables Lua scripting. Read replicas do not support Lua scripts. The master node processes Lua scripts. This is the default value. • 1: enables Lua scripting. Lua scripts that include only read requests are forwarded to read replicas.	Not supported	Not supported	Supported
transfer_subscrible_to_psubscrible_enable	Specifies whether to enable the feature of converting **SUBSCRIBE** to **PSUBSCRIBE**. Valid values: • 0: disables this feature. The two commands are not interchangeable. This is the default value. • 1: enables this feature. Redis Proxy can convert **SUBSCRIBE** to **PSUBSCRIBE**. You can enable this feature if you use the PUB/SUB commands in Lua scripts, and the channel to which you have subscribed cannot receive messages.	Not supported	Supported	Supported

For more information, see redis.conf for Redis 5.0.

Custom parameters of Redis 4.0

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
#no_loose_audit-read-enabled	Specifies whether to enable read request audit. After you enable this feature, logs of read requests are displayed in audit logs.	Not supported	Supported	Supported
#no_loose_check-whitelist-always	Specifies whether to verify that the client IP address is in a whitelist of the ApsaraDB for Redis instance if password-free access is enabled in a Virtual Private Cloud (VPC) network. If you set this parameter to yes, the whitelist still takes effect for password-free access over a VPC network. Valid values: • yes: enables the system to check whether a client IP address is in a whitelist. • no: disables the compatibility with the Sentinel mode. This is the default value.	Supported	Supported	Supported
#no_loose_disabled-commands	Specifies the commands that you want to disable, including **FLUSHALL**, **FLUSHDB**, **KEYS**, **HGETALL**, **EVAL**, **EVALSHA**, and **SCRIPT**. Separate multiple commands with commas (,).	Supported	Supported	Supported
#no_loose_ssl-enabled	Specifies whether to enable SSL encryption. Valid values: • yes: enables SSL encryption. • no: disables SSL encryption. This is the default value.	Not supported	Supported	Supported
#no_loose_sentinel-enabled	Specifies whether to enable the compatibility with the Sentinel mode. Valid values: • yes: enables the compatibility with the Sentinel mode. • no: disables the compatibility with the Sentinel mode. This is the default value.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
read_request_only_rnode_whenrwsplit_enable	Specifies whether to enable directional forwarding for requests from accounts that have read-only permissions. Valid values: • 1: enables directional forwarding. Requests from accounts that have read-only permissions are only forwarded to read replicas. • 0: disables directional forwarding. Requests from accounts that have read-only permissions are forwarded all nodes including the primary node based on weights. This is the default value.	Not supported	Not supported	Supported
appendonly	Specifies whether to enable append-only file (AOF) persistence. Valid values: • yes: enables AOF persistence. This is the default value. • no: disables AOF persistence. Note If you set this parameter to no, AOF persistence is disabled on the master node and remains functional on the replica node.	Supported	Supported	Supported
client-output-buffer-limit pubsub	Specifies output buffer limits of publisher and subscriber clients. The clients are disconnected when the specified limits are reached. Specify this parameter in this format: `<hard limit> <soft limit> <soft seconds>` . • hard limit: disconnects a client if the output buffer of the client is larger than or equal to the value specified by the hard limit field. Unit: bytes. • soft limit and soft seconds: disconnects a client if the output buffer of the client is larger than or equal to the value specified by the soft limit field and if this situation lasts for a period longer than or equal to the value specified by the soft seconds field. The soft limit value is measured in bytes, and the soft seconds value is measured in seconds.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
dynamic-hz	Specifies whether to enable a dynamic hz value. Valid values: • yes: enables a dynamic hz value. This is the default value. • no: disables a dynamic hz value.	Not supported	Not supported	Not supported
hash-max-ziplist-entries	The upper limit of the number of bytes in each key-value pair stored in a hash to support ziplist encoding. The ziplist data structure is used only if both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the hash is less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored in the hash is less than the value of the hash-max-ziplist-entries parameter.	Supported	Supported	Supported
hash-max-ziplist-value	The upper limit of the number of bytes in each key-value pair stored in a hash to support ziplist encoding. The ziplist data structure is used only if both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the hash is less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored in the hash is less than the value of the hash-max-ziplist-entries parameter.	Supported	Supported	Supported
hz	Specifies the frequency at which Redis background tasks are performed. For example, to evict expired keys. You can specify a value from 1 to 500 as the task frequency. The default value is 10. A higher value results in higher CPU consumption but allows the system to delete expired keys and terminate timeout connections more frequently. We recommend that you specify a value smaller than or equal to 100.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
lazyfree-lazy-eviction	Specifies whether to use Lazyfree to evict data. Valid values: • yes: enables Lazyfree to evict data. • no: disables Lazyfree to evict data. This is the default value.	Supported	Supported	Supported
lazyfree-lazy-expire	Specifies whether to enable Lazyfree to delete expired keys. Valid values: • yes: enables Lazyfree to delete expired keys. This is the default value. • no: disables Lazyfree to delete expired keys.	Supported	Supported	Supported
lazyfree-lazy-server-del	Specifies whether to run the **DEL** command and asynchronously delete data based on Lazyfree. • yes: runs Lazyfree. This is the default value. • no: does not run Lazyfree.	Supported	Supported	Supported
list-compress-depth	Specifies the number of entries that are not compressed at both ends of a list. Valid values: 0 to 65535. • 0: does not compress any node of the list. This is the default value. • 1: specifies that the first node from each end of the list is not compressed, but all nodes between these two nodes are compressed. • 2: specifies that the first two nodes from each end of the list are not compressed, but all nodes between these four nodes are compressed. • 3: specifies that the first three nodes from each end of the list are not compressed, but all nodes between these six nodes are compressed. • You can specify other values based on the same rule.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
list-max-ziplist-size	• The maximum length of a ziplist in a quicklist. A positive value specifies the maximum number of elements in each ziplist of a quicklist. For example, if you set this parameter to 5, each ziplist of a quicklist can contain a maximum of five elements. • A negative number indicates the maximum number of bytes in each ziplist of a quicklist. Valid values: [-5, -1], where: ◦ -5: specifies that each ziplist of a quicklist cannot exceed 64 KB (1 KB = 1,024 bytes). ◦ -4: specifies that each ziplist of a quicklist cannot exceed 32 KB. ◦ -3: specifies that each ziplist of a quicklist cannot exceed 16 KB. ◦ -2: specifies that each ziplist of a quicklist cannot exceed 8 KB. This is the default value. ◦ -1: specifies that each ziplist of a quicklist cannot exceed 4 KB.	Supported	Supported	Supported
maxmemory-policy	Specifies the policy used to evict data when the system runs out of memory. Valid values: • volatile-lru: evicts the approximated least recently used (LRU) keys among keys that have time-to-live (TTL) values configured. • allkeys-lru: evicts the approximated LRU keys. • volatile-lfu: evicts the approximated least frequently used (LFU) keys among keys that have TTL values configured. • allkeys-lfu: evicts the approximated LFU keys. • volatile-random: evicts random keys among keys that have TTL values configured. • allkeys-random: evicts random keys. • volatile-ttl: evicts keys with the minimum TTL. The LRU, LFU, and volatile-ttl policies use approximated randomized algorithms. • noeviction: specifies that the system does not evict any keys, but returns an error for write operations.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
notify-keyspace-events	The types of events of which the Redis server can notify clients. The value of this parameter is any combination of the following characters, • each of which specifies a type of events: • K: keyspace events, published with the __keyspace@<db>__ prefix. • E: key events, published with the __keyevent@<db>__ prefix. • g: generic events that are not related to any specific commands, such as DEL, EXPIRE, and RENAME. • l: events of list commands. • s: events of set commands. • h: events of hash commands. • z: event of sorted set commands. • x: events of expired keys. An expiration event is triggered when an expired key is deleted. • e: the eviction events. An eviction event is triggered when a key is deleted due to the policy specified by the maxmemory-policy parameter. • A: the alias for g$lshzxe.	Supported	Not supported	Not supported
set-max-intset-entries	The maximum number of data entries in a set to support inset encoding. A set uses inset encoding when both of the following conditions are met: 1. The number of entries in the set is less than or equal to the value of the set-max-intset-entries parameter. 2. The set only contains radix-10 integers in the range of 64-bit signed integers.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
slowlog-log-slower-than	Specifies whether to log slow queries. • Negative value: does not log slow queries. • 0: logs all queries. • Positive value: logs queries of which the duration exceeds the specified value. Valid values: 0 to 10000000. Default value: 10000. Unit: microseconds.	Supported	Supported	Supported
slowlog-max-len	The maximum number of slow query logs that can be stored. Valid values: 100 to 10000. Default value: 1024.	Supported	Supported	Supported
stream-node-max-bytes	The maximum amount of memory in bytes that each macro node in a stream can consume. Valid values: 0 to 999999999999999. Unit: bytes. A value of 0 specifies that the amount of consumed memory is not limited.	Not supported	Not supported	Not supported
stream-node-max-entries	The maximum number of entries stored on each macro node in a stream. Valid values: 0 to 999999999999999. A value of 0 specifies that the number of entries on each node is not limited.	Not supported	Not supported	Not supported
timeout	Specifies a timeout period. The system terminates a connection to a client if the client has been idle for the specified period of time. Valid values: 0 to 100000. Unit: seconds. A value of 0 specifies that no timeout period is specified for connections.	Supported	Not supported	Not supported
zset-max-ziplist-entries	The upper limit of the number of key-value pairs stored in a sorted set to support ziplist encoding. A sorted set uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the sorted set is less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored in the sorted set is less than the value of the zset-max-ziplist-entries parameter.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
zset-max-ziplist-value	The upper limit of the number of bytes in each key-value pair stored in a sorted set to support ziplist encoding. A sorted set uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the sorted set is less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored in the sorted set is less than the value of the zset-max-ziplist-entries parameter.	Supported	Supported	Supported
list-max-ziplist-entries	The upper limit of the number of bytes in each element stored in a list to support ziplist encoding. A list uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each element stored in the list is less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored in the list is less than the value of the list-max-ziplist-entries parameter.	Not supported	Not supported	Not supported
list-max-ziplist-value	The upper limit of the number of bytes in each element stored in a list to support ziplist encoding. A list uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each element stored in the list is less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored in the list is less than the value of the list-max-ziplist-entries parameter.	Not supported	Not supported	Not supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
cluster_compat_enable	Specifies whether to enable the compatibility with the native Redis cluster syntax. Valid values: • 0: disables the compatibility mode. • 1: enables the compatibility mode. This is the default value.	Not supported	Supported	Supported
script_check_enable	Specifies whether to verify that the keys used in Lua scripts are mapped to the same slot. Valid values: • 0: does not check whether the keys are mapped to the same slot. • 1: checks whether the keys are mapped to the same slot. This is the default value.	Not supported	Supported	Supported
readonly_lua_route_rnode_enable	Specifies whether to enable Lua scripting on read replicas. Valid values: • 0: disables Lua scripting. Read replicas do not support Lua scripts. The master node processes Lua scripts. This is the default value. • 1: enables Lua scripting. Lua scripts that include only read requests are forwarded to read replicas.	Not supported	Not supported	Supported
transfer_subscrible_to_psubscrible_enable	Specifies whether to enable the feature of converting **SUBSCRIBE** to **PSUBSCRIBE**. Valid values: • 0: disables this feature. The two commands are not interchangeable. This is the default value. • 1: enables this feature. Redis Proxy can convert **SUBSCRIBE** to **PSUBSCRIBE**. You can enable this feature if you use the PUB/SUB commands in Lua scripts, and the channel to which you have subscribed cannot receive messages.	Not supported	Supported	Supported

For more information, see redis.conf for Redis 4.0.

Custom parameters of Redis 2.8

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
#no_loose_audit-read-enabled	Specifies whether to enable read request audit. After you enable this feature, logs of read requests are displayed in audit logs.	Not supported	Supported	Supported
#no_loose_check-whitelist-always	Specifies whether to verify that the client IP address is in a whitelist of the ApsaraDB for Redis instance if password-free access is enabled in a Virtual Private Cloud (VPC) network. If you set this parameter to yes, the whitelist still takes effect for password-free access over a VPC network. Valid values: • yes: specifies that the system checks whether a client IP address is in a whitelist. • no: specifies that the system does not check whether a client IP address is in a whitelist.	Not supported	Not supported	Not supported
#no_loose_disabled-commands	Specifies the commands that you want to disable, including **FLUSHALL**, **FLUSHDB**, **KEYS**, **HGETALL**, **EVAL**, **EVALSHA**, and **SCRIPT**. Separate multiple commands with commas (,).	Supported	Supported	Supported
#no_loose_ssl-enabled	Specifies whether to enable SSL encryption. Valid values: • yes: enables SSL encryption. • no: disables SSL encryption. This is the default value.	Not supported	Supported	Supported
#no_loose_sentinel-enabled	Specifies whether to enable the compatibility with the Sentinel mode. Valid values: • yes: enables the compatibility with the Sentinel mode. • no: disables the compatibility with the Sentinel mode. This is the default value.	Not supported	Not supported	Not supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
read_request_only_ronode_whenrwsplit_enable	Specifies whether to enable directional forwarding for requests from accounts that have read-only permissions. Valid values: • 1: enables directional forwarding. Requests from accounts that have read-only permissions are only forwarded to read replicas. • 0: disables directional forwarding. Requests from accounts that have read-only permissions are forwarded all nodes including the primary node based on weights. This is the default value.	Not supported	Not supported	Supported
appendonly	Specifies whether to enable append-only file (AOF) persistence. Valid values: • yes: enables AOF persistence. This is the default value. • no: disables AOF persistence. Note If you set this parameter to no, AOF persistence is disabled on the master node and remains functional on the replica node.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
client-output-buffer-limit pubsub	Specifies output buffer limits of publisher and subscriber clients. The clients are disconnected when the specified limits are reached. Specify this parameter in the following format: `<hard limit> <soft limit> <soft seconds>`. • hard limit: disconnects a client if the output buffer of the client is larger than or equal to the value specified by the hard limit field. Unit: bytes. • soft limit and soft seconds: disconnects a client if the output buffer of the client is larger than or equal to the value specified by the soft limit field and if this situation lasts for a period longer than or equal to the value specified by the soft seconds field. The soft limit value is measured in bytes, and the soft seconds value is measured in seconds.	Supported	Supported	Supported
dynamic-hz	Specifies whether to enable a dynamic hz value. Valid values: • yes: enables a dynamic hz value. This is the default value. • no: disables a dynamic hz value.	Not supported	Not supported	Not supported
hash-max-ziplist-entries	The upper limit of the number of bytes in each key-value pair stored in a hash to support ziplist encoding. The ziplist data structure is used only if both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the hash is less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored in the hash is less than the value of the hash-max-ziplist-entries parameter.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
hash-max-ziplist-value	The upper limit of the number of bytes in each key-value pair stored in a hash to support ziplist encoding. The ziplist data structure is used only if both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the hash is less than the value of the hash-max-ziplist-value parameter. 2. The number of key-value pairs stored in the hash is less than the value of the hash-max-ziplist-entries parameter.	Supported	Supported	Supported
hz	Specifies the frequency at which Redis background tasks are performed. For example, to evict expired keys. You can specify a value from 1 to 500 as the task frequency. The default value is 10. A higher value results in higher CPU consumption but allows the system to delete expired keys and terminate timeout connections more frequently. We recommend that you specify a value smaller than or equal to 100.	Supported	Supported	Supported
lazyfree-lazy-eviction	Specifies whether to use Lazyfree to evict data. Valid values: • yes: enables Lazyfree to evict data. • no: disables Lazyfree to evict data. This is the default value.	Not supported	Not supported	Not supported
lazyfree-lazy-expire	Specifies whether to enable Lazyfree to delete expired keys. Valid values: • yes: enables Lazyfree to delete expired keys. This is the default value. • no: disables Lazyfree to delete expired keys.	Not supported	Not supported	Not supported
lazyfree-lazy-server-del	Specifies whether to run the **DEL** command and asynchronously delete data based on Lazyfree. • yes: enables asynchronous deletion. This is the default value. • no: disables asynchronous deletion.	Not supported	Not supported	Not supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
list-compress-depth	Specifies the number of entries that are not compressed at both ends of a list. Valid values: 0 to 65535. • 0: does not compress any node of the list. This is the default value. • 1: specifies that the first node from each end of the list is not compressed, but all nodes between these two nodes are compressed. • 2: specifies that the first two nodes from each end of the list are not compressed, but all nodes between these four nodes are compressed. • 3: specifies that the first three nodes from each end of the list are not compressed, but all nodes between these six nodes are compressed. • You can specify other values based on the same rule.	Not supported	Not supported	Not supported
list-max-ziplist-size	• The maximum length of a ziplist in a quicklist. A positive value specifies the maximum number of elements in each ziplist of a quicklist. For example, if you set this parameter to 5, each ziplist of a quicklist can contain a maximum of five elements. • A negative number indicates the maximum number of bytes in each ziplist of a quicklist. Valid values: [-5, -1], where: ◦ -5: specifies that each ziplist of a quicklist cannot exceed 64 KB (1 KB = 1,024 bytes). ◦ -4: specifies that each ziplist of a quicklist cannot exceed 32 KB. ◦ -3: specifies that each ziplist of a quicklist cannot exceed 16 KB. ◦ -2: specifies that each ziplist of a quicklist cannot exceed 8 KB. This is the default value. ◦ -1: specifies that each ziplist of a quicklist cannot exceed 4 KB.	Not supported	Not supported	Not supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
maxmemory-policy	Specifies the policy used to evict data when the system runs out of memory. Valid values: • volatile-lru: evicts the approximated least recently used (LRU) keys among keys that have time-to-live (TTL) values configured. • allkeys-lru: evicts the approximated LRU keys. • volatile-lfu: evicts the approximated least frequently used (LFU) keys among keys that have TTL values configured. • allkeys-lfu: evicts the approximated LFU keys. • volatile-random: evicts random keys among keys that have TTL values configured. • allkeys-random: evicts random keys. • volatile-ttl: evicts keys with the minimum TTL. The LRU, LFU, and volatile-ttl policies use approximated randomized algorithms. • noeviction: specifies that the system does not evict any keys, but returns an error for write operations.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
notify-keyspace-events	The types of events of which the Redis server can notify clients. The value of this parameter is any combination of the following characters, • each of which specifies a type of events: • K: keyspace events, published with the __keyspace@<db>__ prefix. • E: key events, published with the __keyevent@<db>__ prefix. • g: generic events that are not related to any specific commands, such as DEL, EXPIRE, and RENAME. • l: events of list commands. • s: events of set commands. • h: events of hash commands. • z: event of sorted set commands. • x: events of expired keys. An expiration event is triggered when an expired key is deleted. • e: the eviction events. An eviction event is triggered when a key is deleted due to the policy specified by the maxmemory-policy parameter. • A: the alias for g$lshzxe.	Supported	Not supported	Not supported
set-max-intset-entries	The maximum number of data entries in a set to support inset encoding. A set uses inset encoding when both of the following conditions are met: 1. The number of entries in the set is less than or equal to the value of the set-max-intset-entries parameter. 2. The set only contains radix-10 integers in the range of 64-bit signed integers.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
slowlog-log-slower-than	Specifies whether to log slow queries. • Negative value: does not log slow queries. • 0: logs all queries. • Positive value: logs queries of which the duration exceeds the specified value. Valid values: 0 to 10000000. Default value: 10000. Unit: microseconds.	Supported	Supported	Supported
slowlog-max-len	The maximum number of slow query logs that can be stored. Valid values: 100 to 10000. Default value: 1024.	Supported	Supported	Supported
stream-node-max-bytes	The maximum amount of memory in bytes that each macro node in a stream can consume. Valid values: 0 to 999999999999999. Unit: bytes. A value of 0 specifies that the amount of consumed memory is not limited.	Not supported	Not supported	Not supported
stream-node-max-entries	The maximum number of entries stored on each macro node in a stream. Valid values: 0 to 999999999999999. A value of 0 specifies that the number of entries on each node is not limited.	Not supported	Not supported	Not supported
timeout	Specifies a timeout period. The system terminates a connection to a client if the client has been idle for the specified period of time. Valid values: 0 to 100000. Unit: seconds. A value of 0 specifies that no timeout period is specified for connections.	Not supported	Not supported	Not supported
zset-max-ziplist-entries	The upper limit of the number of key-value pairs stored in a sorted set to support ziplist encoding. A sorted set uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the sorted set is less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored in the sorted set is less than the value of the zset-max-ziplist-entries parameter.	Supported	Supported	Supported

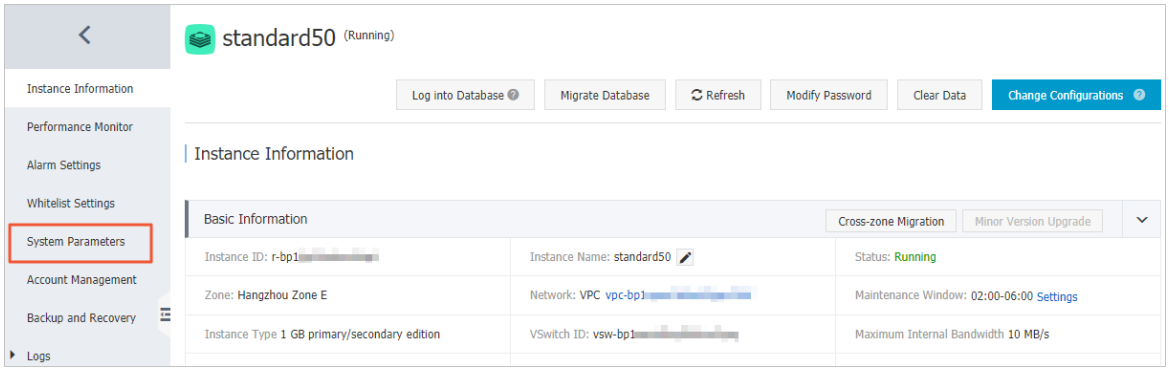
Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
zset-max-ziplist-value	The upper limit of the number of bytes in each key-value pair stored in a sorted set to support ziplist encoding. A sorted set uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each key-value pair stored in the sorted set is less than the value of the zset-max-ziplist-value parameter. 2. The number of key-value pairs stored in the sorted set is less than the value of the zset-max-ziplist-entries parameter.	Supported	Supported	Supported
list-max-ziplist-entries	The upper limit of the number of bytes in each element stored in a list to support ziplist encoding. A list uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each element stored in the list is less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored in the list is less than the value of the list-max-ziplist-entries parameter.	Supported	Supported	Supported
list-max-ziplist-value	The upper limit of the number of bytes in each element stored in a list to support ziplist encoding. A list uses ziplist encoding when both of the following conditions are met: 1. The number of bytes in each element stored in the list is less than the value of the list-max-ziplist-value parameter. 2. The number of elements stored in the list is less than the value of the list-max-ziplist-entries parameter.	Supported	Supported	Supported

Parameter	Description	Standard master-replica instance	Master-replica cluster instance	Read/write splitting instance
cluster_compat_enable	Specifies whether to enable the compatibility with the native Redis cluster syntax. Valid values: • 0: disables the compatibility mode. • 1: enables the compatibility mode. This is the default value.	Not supported	Supported	Supported
script_check_enable	Specifies whether to verify that the keys used in Lua scripts are mapped to the same slot. Valid values: • 0: does not check whether the keys are mapped to the same slot. • 1: checks whether the keys are mapped to the same slot. This is the default value.	Not supported	Supported	Supported
readonly_lua_route_rnode_enable	Specifies whether to enable Lua scripting on read replicas. Valid values: • 0: disables Lua scripting. Read replicas do not support Lua scripts. The master node processes Lua scripts. This is the default value. • 1: enables Lua scripting. Lua scripts that include only read requests are forwarded to read replicas.	Not supported	Not supported	Supported
transfer_subscrible_to_psubscrible_enable	Specifies whether to enable the feature of converting **SUBSCRIBE** to **PSUBSCRIBE**. Valid values: • 0: disables this feature. The two commands are not interchangeable. This is the default value. • 1: enables this feature. Redis Proxy can convert **SUBSCRIBE** to **PSUBSCRIBE**. You can enable this feature if you use the PUB/SUB commands in Lua scripts, and the channel to which you have subscribed cannot receive messages.	Not supported	Not supported	Not supported

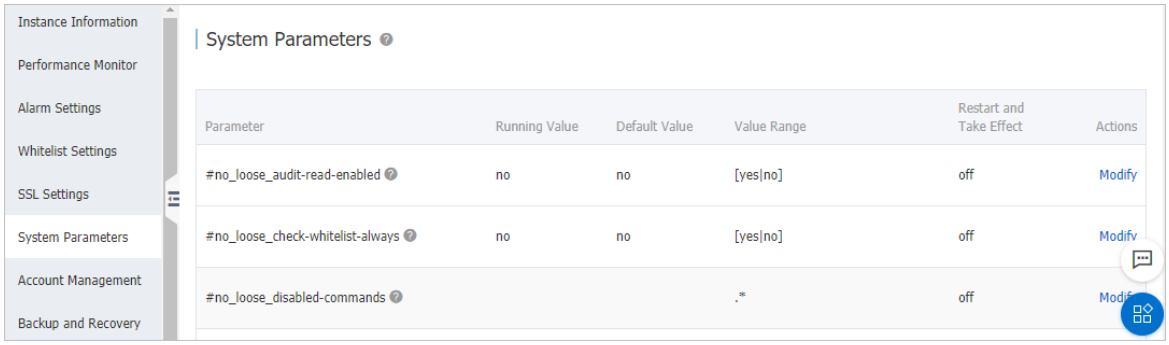
For more information, see redis.conf for Redis 2.8.

Modify parameters in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane of the **Instance Information** page, click **System Parameters**.



5. On the **System Parameters** page, find the parameter that you want to set and click **Modify** in the **Actions** column.



6. In the dialog box that appears, modify the parameter value and click **OK**.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

Related operations

API	Description
#no_loose_audit-read-enabled	Specifies whether to enable read request audit. After you enable this feature, logs of read requests are displayed in audit logs.
DescribeParameters	Queries configuration parameters and running parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies configuration parameters of an ApsaraDB for Redis instance.

FAQ

- Can I modify the maxclients parameter?

The value of maxclients is used to specify the maximum number of concurrent connections to an ApsaraDB for Redis data node. You cannot change the value of maxclients. Except for performance-enhanced instances of ApsaraDB for Redis Enhanced Edition, the default value of maxclients is 10000. To increase the maximum number of concurrent connections, you can scale out the instances.

- What is a zone-disaster recovery instance?

  When you create an ApsaraDB for Redis instance, if you select the zones where zone-disaster recovery is supported, for example, , **US (Virginia), Virginia Zone A and Virginia Zone B**, a zone-disaster recovery instance is created. For more information, see Zone-disaster recovery.

3.3.2. Disable AOF persistence

This topic describes how to set the appendonly parameter to disable AOF persistence. You can also set this parameter again to enable AOF. By default, append-only file (AOF) persistence is enabled for ApsaraDB for Redis.

Overview

ApsaraDB for Redis provides data persistence options: AOF persistence and Redis Database (RDB) persistence. AOF persistence logs every write operation received by the server, such as **SET**. After you restart ApsaraDB for Redis, the service runs the operations logged in the AOF files to restore data. If the files become too big, ApsaraDB for Redis can rewrite the files to optimize storage usage in the background.

AOF persistence follows the AOF_FSYNC_EVERYSEC policy of fsync every second write performances. The system records the received write commands to AOF every second. The policy has the least effect on the performance and can minimize data loss caused by user errors or a power outage. ApsaraDB for Redis can archive incremental AOF files, ensuring service performance when ApsaraDB for Redis rewrites the logs.

AOF persistence may affect the write performance. If an ApsaraDB for Redis instance is used in a cache-only scenario, you can follow the steps in this topic to set the appendonly parameter to disable AOF persistence for the instance.

Status and impacts of AOF persistence

- By default, AOF is enabled for an ApsaraDB for Redis instance.
- If you set the value of appendonly to no:
  - The system disables AOF persistence without restarting the instance.
  - After AOF persistence is disabled, you can no longer use the AOF files to restore data .
  - The existing AOF logs remain unchanged.
  - For a standard instance, AOF persistence is disabled for the master. The replica is not affected.
  - For a cluster instance, AOF persistence is disabled for masters of all shards. Replicas are not affected.
  - For a read/write splitting instance, AOF persistence is disabled for the master and all read replicas. The replica servers as a backup for the master is not affected.
- If you change the value of appendonly to yes: The system enables AOF persistence without restarting the instance.

Disable AOF persistence in the console

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select the region where the instance is deployed.
3. On the Instances page, click **Instance ID**, or click the More icon and select **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.
5. On the System Parameters page, click **Modify** in the **Actions** column for the appendonly parameter.
6. In the dialog box that appears, proceed with the following steps:

   > **Warning** After you disable AOF persistence, you can no longer use AOF files to restore data. You can only use RDB files to restore data. Proceed with caution.

   i. Set the appendonly parameter. Valid values:
      - yes: enables AOF persistence.
      - no: disables AOF persistence.
   ii. Click **OK**.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.3. Enable compatibility with the syntax of Redis Cluster

ApsaraDB for Redis cluster instances are compatible with the syntax of Redis Cluster and allow clients such as JedisCluster to use commands to access cluster nodes. After you enable compatibility with the syntax of Redis Cluster, you can seamlessly migrate your on-premises Redis clusters to Alibaba Cloud without modifying the code.

Prerequisites

- A cluster instance of ApsaraDB for Redis is used.
- The network type of the instance is Virtual Private Cloud (VPC).

Specify parameters in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the instance is deployed.
3. On the Instances page, find the target instance, click the **instance ID** or click **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.
5. On the System Parameters page, find the cluster_compat_enable parameter and click **Modify** in the **Actions** column.
6. In the dialog box that appears, set the parameter to *1* and click **OK**.

   A value of 0 specifies that this feature is disabled. A value of 1 specifies that this feature is enabled. The default value is 0.

   For more information, see Parameter overview and configuration guide.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.4. Limit the size of output buffers for Pub/Sub clients

ApsaraDB for Redis provides the client-output-buffer-limit pubsub parameter for you to specify a size limit for output buffers of Pub/Sub clients. If the data to be buffered for a Pub/Sub client exceeds the size limit, ApsaraDB for Redis closes the connection to the client. This prevents the buffered data from consuming too much memory and ensures the performance of ApsaraDB for Redis.

Limits on output buffers in ApsaraDB for Redis

ApsaraDB for Redis allocates an output buffer in the memory to each client. After processing the commands from clients, ApsaraDB for Redis temporarily stores the result data in output buffers and then sends the result data to the clients. If you do not limit the size of data in output buffers, a large amount of data may accumulate in output buffers. The data may eventually use up all the available memory and result in a service crash. This issue may occur in the following scenarios:

- A large amount of data needs to be returned for commands from clients.
- Message publishing outpaces message consumption.

By setting the client-output-buffer-limit pubsub parameter to a proper value, you can prevent the output buffers of Pub/Sub clients from consuming too much memory.

Parameter options

The client-output-buffer-limit pubsub parameter includes the following options: `hard limit`, `soft limit`, and `soft seconds`.

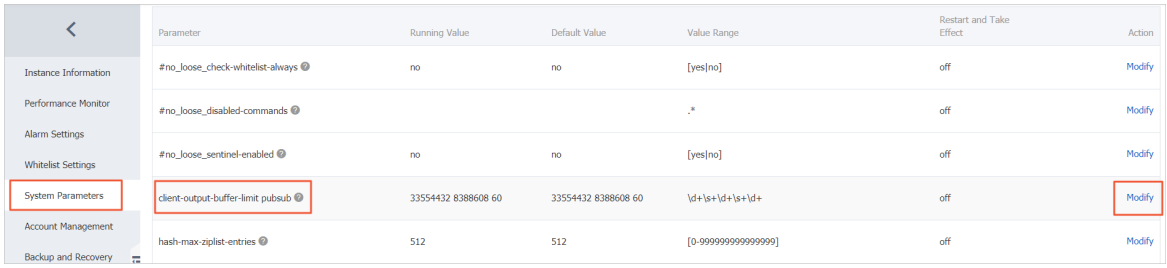
- `hard limit`: If the output buffer of a Pub/Sub client reaches or exceeds the number of bytes specified by `hard limit`, the client is immediately disconnected.

- `soft limit` and `soft seconds`: If the output buffer of a Pub/Sub client reaches or exceeds the upper limit specified by `soft limit` for a period of time in seconds that is specified by `soft seconds`, the client is disconnected.

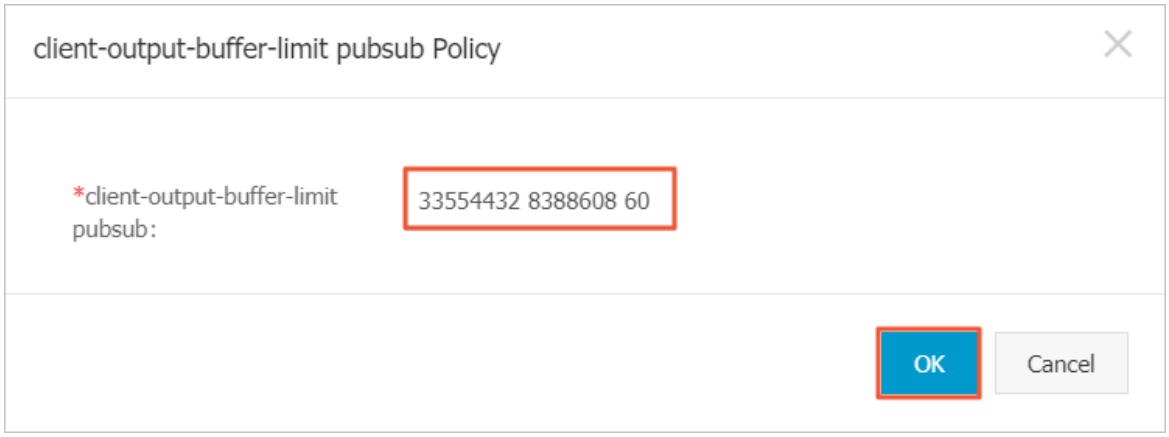
In ApsaraDB for Redis, the default values of `hard limit`, `soft limit`, and `soft seconds` are 33554432 bytes (32 MB), 8388608 bytes (8 MB), and 60 seconds, respectively. You can customize the values based on your business requirements and the capacities of clients.

Specify the parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. On the Instances page, find the target instance, click the **instance ID** or click **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.
5. On the page that appears, find the client-output-buffer-limit pubsub parameter and click **Modify** in the **Actions** column.



6. In the dialog box that appears, follow these steps to specify the parameter.
   i. Specify the client-output-buffer-limit pubsub parameter based on the description in Parameter options.
   ii. Click **OK**.



Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.5. Change the frequency of background tasks

You can modify the hz parameter to change the frequency how ApsaraDB for Redis runs background tasks to delete expired keys and close client connections that are timed out.

Correlation between background tasks and the hz parameter

To regularly check the status of resources and services and take actions based on specified policies, ApsaraDB for Redis calls an internal function to run background tasks, such as the following:

- Calculate the least recently used (LRU) information and delete expired keys.
- Close client connections that are timed out.
- Perform incremental rehashing on hash tables.
- Trigger BGSAVE/AOF rewrites.
- Update statistics.

ApsaraDB for Redis runs background tasks to ensure service availability. ApsaraDB for Redis uses the hz parameter to control the frequency how background tasks are executed. The default value of this parameter is 10, indicating that ApsaraDB for Redis runs background tasks 10 times per second.

Scenarios

ApsaraDB for Redis runs background tasks to delete expired keys. The process is as follows:

1. ApsaraDB for Redis randomly selects 20 keys for which a Time to Live (TTL) is specified, and checks whether the selected keys are expired.
2. Expired keys are deleted.
3. If more than 25% of the selected keys are expired, ApsaraDB for Redis runs the background task again.

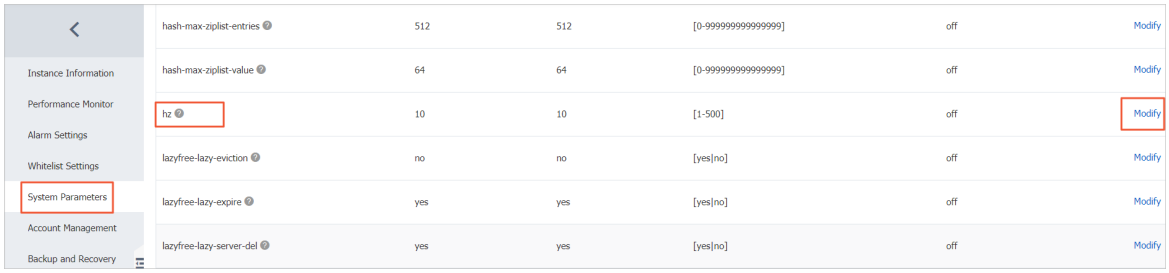
Assume that there are a large number of expired keys or a sharp increase in expired keys and the rate at which ApsaraDB for Redis deletes expired keys is slow. The remaining expired keys occupy a large amount of memory and may affect the performance of ApsaraDB for Redis. To resolve this issue, increase the value of the hz parameter, and the background task is executed more frequently.

Valid values and suggested settings for the hz parameter

Valid values of the hz parameter: 1 to 500. If you increase the value of the hz parameter, background tasks are executed more frequently, but the CPU usage of ApsaraDB for Redis also increases. You can use the default value 10 in most cases. If you want to run certain background tasks more frequently, set a value between 10 and 100. We recommend that you do not set the hz parameter to a value greater than 100, because this may cause a sharp increase in the CPU usage.

Set the hz parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select the region where the instance is deployed.
3. On the instance list page, click **Instance ID** or **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.
5. On the page that appears, find the hz parameter and click **Modify** in the **Action** column.



6. In the hz Policy dialog box that appears, follow these steps:
   i. Specify the value of the hz parameter based on your needs.
   ii. Click OK.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.6. Enable dynamic frequency control for background tasks

The dynamic-hz parameter is a new parameter added in Redis 5.0. You can set this parameter to enable or disable dynamic frequency control for background tasks. After dynamic frequency control is enabled, ApsaraDB for Redis can automatically change the frequency of background tasks based on the number of client connections.

Prerequisites

The engine version of the ApsaraDB for Redis instance is Redis 5.0 and later.

Relationship between the hz and dynamic-hz parameters

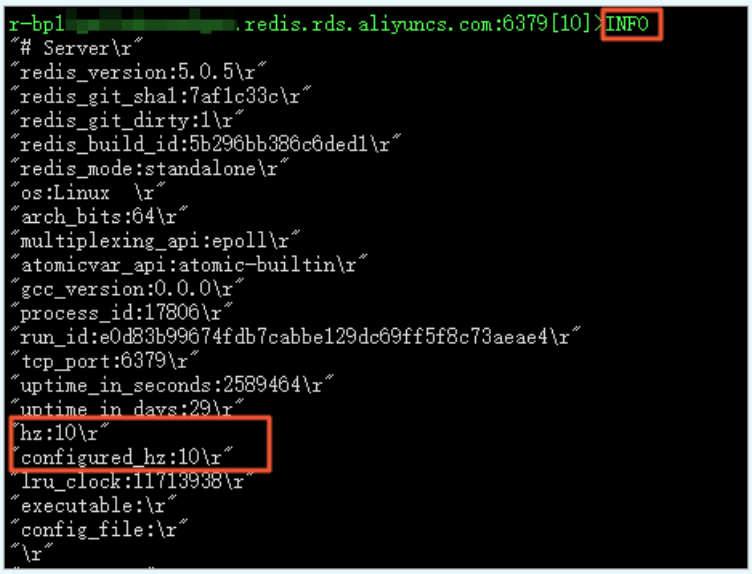
ApsaraDB for Redis supports various background tasks, such as closing client connections that have timed out and evicting expired keys. The hz parameter specifies the frequency of background tasks in ApsaraDB for Redis. For more information, see Change the frequency of background tasks. However, a fixed frequency may cause the following issues:

- If the frequency is too low, resources cannot be recycled in a timely manner when a large number of client connections have timed out or a large number of keys have expired. This may lead to poor performance or even crashes of ApsaraDB for Redis.
- If the frequency is too high, background tasks consume too many CPU resources. The

performance of ApsaraDB for Redis may also deteriorate.

To balance the CPU usage and efficiency of background tasks, Redis 5.0 provides the dynamic-hz parameter to enable or disable dynamic frequency control for background tasks. In addition, Redis 5.0 adds the configured_hz parameter to indicate the frequency that you set, and uses the original hz parameter to indicate the actual frequency.

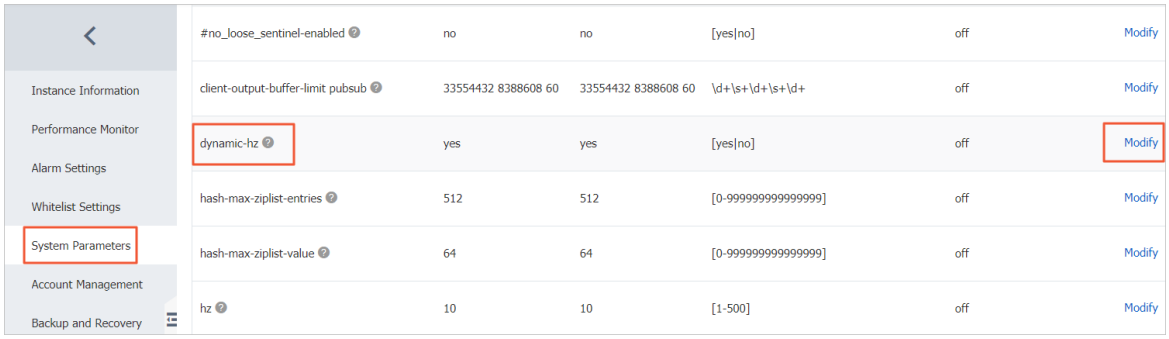
Note You can run the INFO command to query the values of the hz and configured_hz parameters.



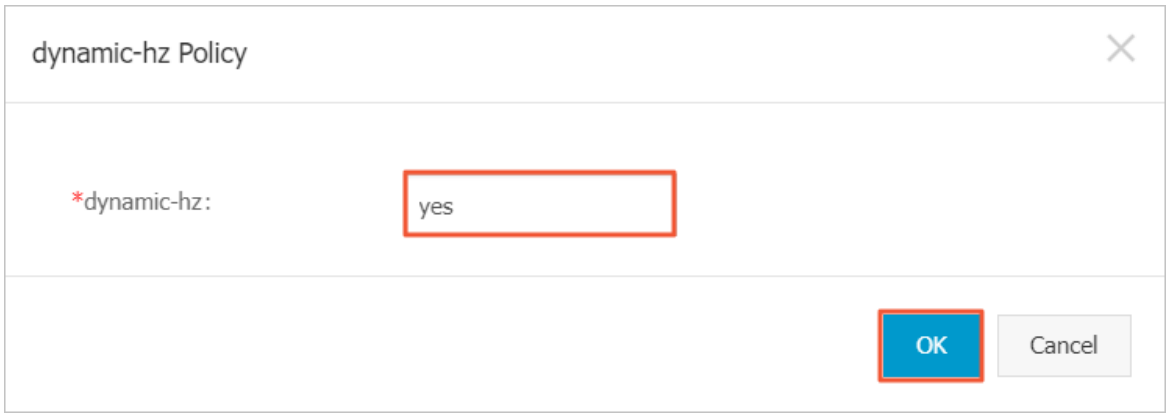
Valid values of the dynamic-hz parameter are `yes` and `no`. A value of yes enables dynamic frequency control and a value of no disables dynamic frequency control. Default value: `yes`. After dynamic frequency control is enabled, the value that you specify for the hz parameter is assigned to the configured_hz parameter as the baseline frequency. ApsaraDB for Redis automatically changes the value of the hz parameter based on the number of client connections. The value of the hz parameter increases with the number of client connections. Accordingly, background tasks are performed more frequently.

Specify the parameters in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. On the Instances page, find the target instance, click **Instance ID** or **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.
5. On the page that appears, find the dynamic-hz parameter and click **Modify** in the **Actions** column.



6. In the dialog box that appears, follow these steps to specify the parameter:
   i. Change the value of the dynamic-hz parameter based on your needs.
   ii. Click **OK**.



Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.7. Customize the size of macro nodes in streams

You can modify the stream-node-max-bytes parameter to specify the maximum memory that can be used by each macro node in streams. You can also modify the stream-node-max-entries parameter to specify the maximum number of stream entries that can be stored on each macro node.

Prerequisites

The engine version of the ApsaraDB for Redis instance is Redis 5.0 and later.

Relationship between Redis streams and macro nodes

Stream is a new data type introduced with Redis 5.0. Redis streams are represented as delta-compressed macro nodes that are linked together by the radix tree. Each macro node stores multiple stream entries. You can use this data structure to access random elements, obtain elements in a specified range, and create capped streams with high efficiency. This data structure also optimizes memory usage.

You can specify the stream-node-max-entries parameter to limit the maximum stream entries supported by each macro node. You can specify the stream-node-max-bytes parameter to limit the maximum memory consumed by each macro node.

- stream-node-max-entries: The default value is 100, which specifies that each macro node can store up to 100 stream entries. Valid values: 0 to 999,999,999,999,999. A value of 0 specifies that each macro node can store an unlimited number of stream entries. If the number of stream entries stored in a macro node reaches the upper limit, new stream entries are stored on a new macro node.
- stream-node-max-bytes: Unit: bytes. The default value is 4096, which specifies that each macro node can consume up to 4,096 bytes of the memory. Valid values: 0 to 999,999,999,999,999. A value of 0 specifies that each macro node can store an unlimited number of stream entries.

Scenarios

You can specify the stream-node-max-entries parameter to adjust the length deviation of a fixed-length message queue.

If your application does not need to store messages permanently, you can use the *MAXLEN* parameter to specify the maximum number of messages in a stream when you run the **XADD** command to add a message to the stream. Example:

```
XADD cappedstream MAXLEN 5000 * field value5001 // Add a value value5001 to field1 of cappedstream and set the maximum number of messages to 5,000.
```

When the number of messages in the stream reaches the upper limit, the earliest message is deleted each time a new message is added. No matter how many messages are added, the maximum length of the stream remains unchanged. In addition, the memory consumed by a message is released after the message is deleted.

> **Note** When you delete a message from a macro node, the message is marked as deleted, but the memory consumed by the message is not released immediately. ApsaraDB for Redis deletes a macro node and releases the consumed memory only when all messages on the macro node are marked as deleted.

However, if you set the maximum queue length to an exact value such as 5,000 messages, precise control is achieved at the sacrifice of performance. To optimize memory usage, Redis streams are represented as delta-compressed macro nodes that are linked together by a radix tree. Each time ApsaraDB for Redis deletes a message, it must search the macro node for the message and mark the message as deleted. This mechanism is not optimal for high-throughput ApsaraDB for Redis services where messages are deleted and added frequently. The performance of ApsaraDB for Redis deteriorates if it deletes messages frequently. Therefore, we recommend that you add a tilde (~) in the XADD command to specify an approximate maximum length. Example:

```
XADD cappedstream MAXLEN ~ 5000 * field value1 // Add a value value5001 to field1 of cappedstream and set the maximum number of messages to approximately 5,000.
```

The actual length of the stream can be an approximate value greater than or equal to the specified value. For example, the stream may contain 5000, 5050, or 5060 messages. The deviation from 5000 depends on the number of macro nodes in the stream and the maximum number of messages that can be stored in each macro node. ApsaraDB for Redis calculates the approximate value based on the stream-node-max-entries parameter. This parameter specifies the maximum number of messages that can be stored on each macro node. If the number of messages stored in the stream exceeds this approximate value, ApsaraDB for Redis deletes the macro node that stores the earliest messages, instead of deleting specific messages.

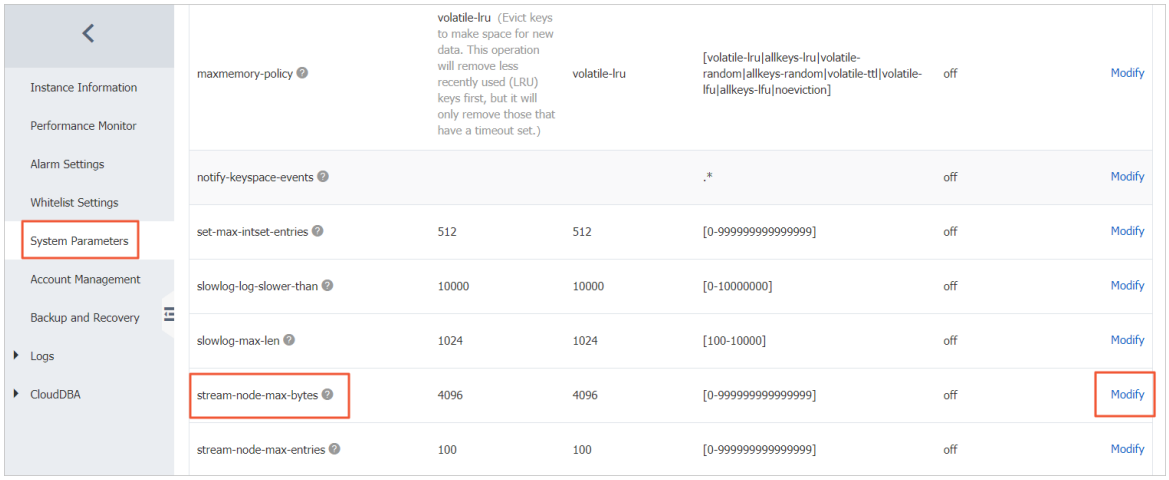
The value of the stream-node-max-entries parameter determines the length deviation of a fixed-length message queue. To reduce the deviation, you can set the parameter to a proper smaller value.

Specify the parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. On the Instances page, click the **instance ID** or click **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.

   > **Note** The following steps describe how to specify the stream-node-max-bytes parameter.

5. On the page that appears, find the stream-node-max-bytes parameter and click **Modify** in the **Actions** column.



6. In the dialog box that appears, follow these steps to specify the parameter.
   i. Change the value of the stream-node-max-bytes parameter based on your business requirements.
   ii. Click **OK**.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.8. Specify a timeout period for client connections

You can set the timeout parameter to specify a timeout period for client connections. Then, ApsaraDB for Redis can close client connections that have timed out to recycle resources.

Prerequisites

An ApsaraDB for Redis instance of the standard edition is created, and the engine version of the instance is Redis 4.0 and later.

> **Note** You cannot modify the timeout parameter for cluster instances or read/write splitting instances of ApsaraDB for Redis.

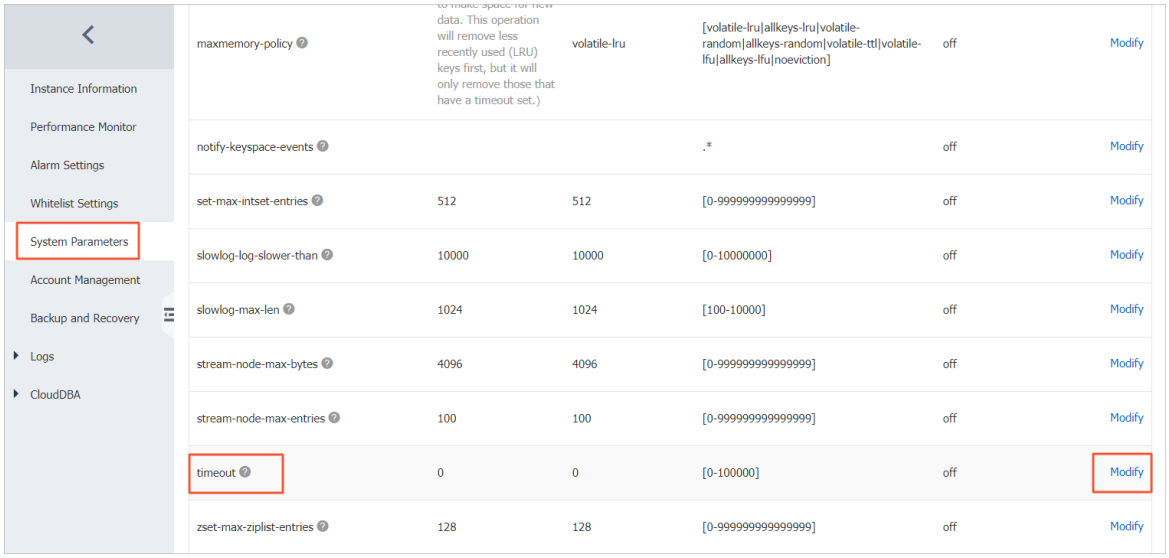
Manage client connections in ApsaraDB for Redis

In common scenarios, you can use clients to manage connections. For example, clients can allocate connections, monitor the status of connections, and recycle resources in the connection pool. By default, ApsaraDB for Redis does not close a client connection even if a client has been idle for a long period of time. However, we recommend that you specify the timeout parameter in core applications to allow ApsaraDB for Redis to recycle resources. If resources are not recycled in a timely manner after exceptions occur on clients, the connection pool may be full with idle client connections. This may result in a service crash. Such an issue in core applications may cause serious impact on your business.

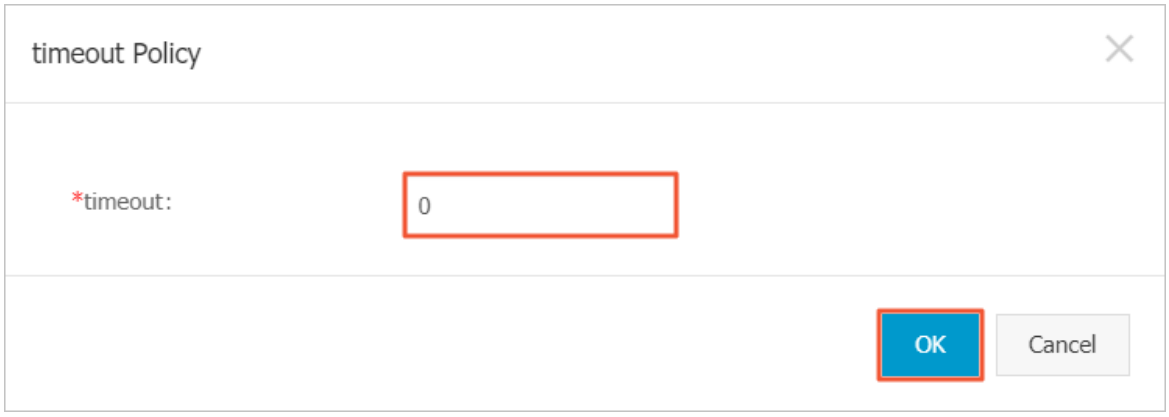
The timeout parameter is measured in seconds and the valid values is from 0 to 100000. The default value is 0, which specifies that client connections never time out. To improve performance, ApsaraDB for Redis does not immediately close a client connection when the client connection reaches the timeout period. For example, if the timeout parameter is set to 10 seconds, a client connection may be closed after it is idle for 12 seconds and after many client connections have been created on the shard server. To reduce the latency, you can set a larger value for the hz parameter to increase the frequency of the background task that closes idle connections. For more information, see Change the frequency of background tasks.

Specify the parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the page, select the region where the instance is deployed.
3. On the Instances page, find the target instance, click the **instance ID** or click **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **System Parameters** in the left-side navigation pane.
5. On the page that appears, find the timeout parameter and click **Modify** in the **Actions** column.



6. In the dialog box that appears, follow these steps to specify the parameter.
   i. Change the value of the timeout parameter based on your requirements.
   ii. Click **OK**.



Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.9. Enable the Redis Sentinel-compatible mode

Redis Sentinel provides high availability (HA) services for Redis. ApsaraDB for Redis is compatible with Sentinel and can be used for services that run Sentinel. This topic describes how to enable the Redis Sentinel-compatible mode in the ApsaraDB for Redis console.

Prerequisites

To use the Redis Sentinel-compatible mode, ApsaraDB for Redis instances must meet the following requirements:

Requirements of the Redis Sentinel-compatible mode

Item	Description
Engine version	Redis 4.0
Network type	VPC
Prerequisite	Password-free access is enabled.

Introduction to Redis Sentinel

Redis Sentinel provides Redis with services such as master and replica monitoring, fault alerting, and automatic failover. Redis Sentinel is used in many business scenarios that use local on-premises Redis databases and require high reliability. To facilitate the migration of Redis databases to the cloud in such scenarios, Alibaba Cloud provides the Redis Sentinel-compatible mode.

> **Note** ApsaraDB for Redis uses the HA component developed by Alibaba Cloud, without the need to use Redis Sentinel.

After you enable the Sentinel-compatible mode, you can use the following commands:

Command	Description
SENTINEL sentinels	Queries Sentinel instances for a specified master and the status of these Sentinel instances. Follow this syntax: `SENTINEL sentinels <the name of a master>`
SENTINEL get-master-addr-by-name	Queries the IP address and port number of a specified master. Follow this syntax: `SENTINEL get-master-addr-by-name <the name of a master>`

> **Note** Instances of ApsaraDB for Redis 2.8 do not support the preceding commands.

Specify the parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the Instances page, click the target **instance ID** or click **Manage** in the **Actions** column.
4. By default, the **Instance Information** page is displayed. In the left-side navigation pane, click

System Parameters.

5. On the System Parameters page, find the #no_loose_sentinel-enabled parameter and click **Modify** in the **Actions** column.

> **Note** If an instance of Redis 4.0 does not support this parameter, upgrade the minor version.

6. In the dialog box that appears, set the parameter to *yes*, and click **OK**.

   For more information about the parameters, see Parameter overview and configuration guide.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.10. Disable risky commands

You can set the #no_loose_disabled-commands parameter in the console to disable certain commands that may degrade service performance and cause data loss.

Background

Unlimited usage of commands may cause some issues in certain scenarios. Some Redis commands can delete most of or all data in a database, such as **FLUSHALL** and **FLUSHDB**. Improper uses of some commands such as **KEYS** and **HGETALL** cause process blocking in the single-threading Redis model and reduce service performance. To ensure stable and efficient management, you can disable these types of commands to minimize risks in your workloads.

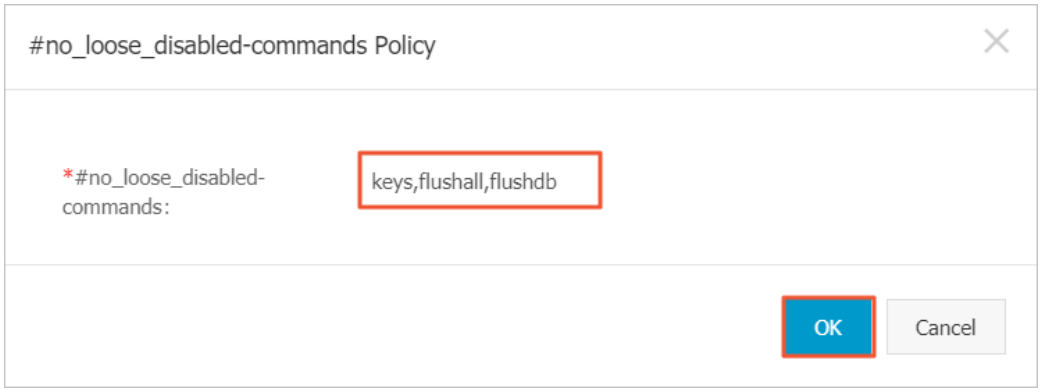
Set the parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is located.
3. In the left-side navigation pane, click Instance List to go to the **Instance List** page. Find the instance that you want to manage, and click the instance ID, or click the More icon and select **Manage** in the **Action** column for the instance.
4. The **Instance Information** page is displayed by default. In the left-side navigation pane, click **System Parameters**.
5. On the System Parameters page, find the #no_loose_disabled-commands parameter and click **Modify** in the **Action** column for the parameter.

   Disable specific Redis commands by setting the system parameter

6. In the dialog box that appears, specify the commands to be disabled and click **OK**.

Specify the Redis commands to be disabled



Note

- The parameter value can only contain lowercase letters. Separate multiple commands with commas (,).
- Commands that can be disabled include **FLUSHALL**, **FLUSHDB**, **KEYS**, **HGETALL**, **EVAL**, **EVALSHA**, and **SCRIPT**.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.3.11. Use the whitelist in password-free access mode

This topic describes how to modify the value of the #no_loose_check-whitelist-always parameter to use the whitelist even when the password-free access feature is enabled.

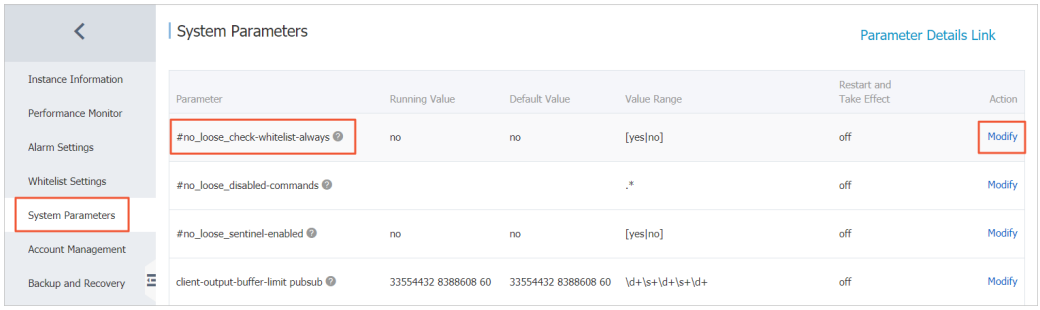
Background

After password-free access is enabled, ApsaraDB for Redis does not restrict access from other services in the same Virtual Private Cloud (VPC) network based on the whitelist. For more information, see Enable password-free access. If password-free access is enabled but you want to allow only specified resources, such as a specified Elastic Compute Service (ECS) or RDS instance, to access your ApsaraDB for Redis instance, you need to modify the value of the #no_loose_check-whitelist-always parameter.

Specify the parameter in the ApsaraDB for Redis console

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the Instances page, click the target **instance ID** or click **Manage** in the **Actions** column.
4. In the left-side navigation pane of the **Instance Information** page, click **System Parameters**.
5. On the System Parameters page, find the **#no_loose_check-whitelist-always** parameter and click **Modify** in the **Actions** column.

Specify the #no_loose_check-whitelist-always parameter.



6. In the dialog box that appears, set the parameter to *yes* and click **OK**.

   Set the #no_loose_check-whitelist-always parameter to yes.

> **Note** To disable the forcible whitelist authentication, set this parameter to *no*.

Call API operations to set parameters

You can also call the ModifyInstanceConfig operation to set parameters.

3.4. Tag management

3.4.1. Create a tag

If you have a large number of ApsaraDB for Redis instances, you can create tags, bind the tags to the instances for classification, and then filter the instances by tag.

Precautions

- A tag consists of a key-value pair. The key must be unique in the same region of the same account, but the values do not need to be unique.

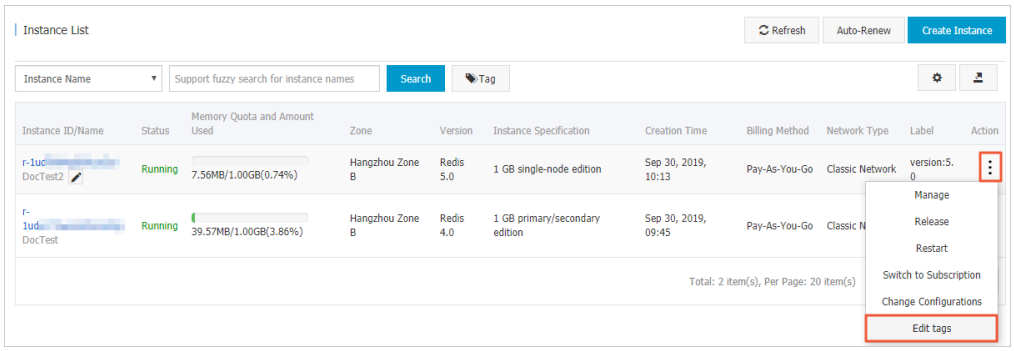
  > **Note** A key can multiple values or no value at all.

- You can edit tags for a maximum of 50 instances simultaneously.
- You can bind a maximum of 20 tags to an instance.
- You can bind or unbind a maximum of 20 tags at a time.

Procedure

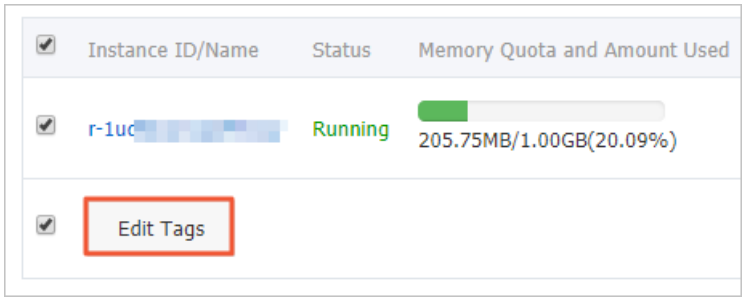
1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select the region where the instance resides.
3. On the **Instance List** page, create a tag for a single instance or multiple instances at the same time.
   - To create a tag for a single instance, choose > **Edit tags** in the **Action** column for the target instance.

     Create a tag for a single instance

   - To create tags for multiple instances, select the target instances and click **Edit Tags** below the instance list.

     Create a tag for multiple instances

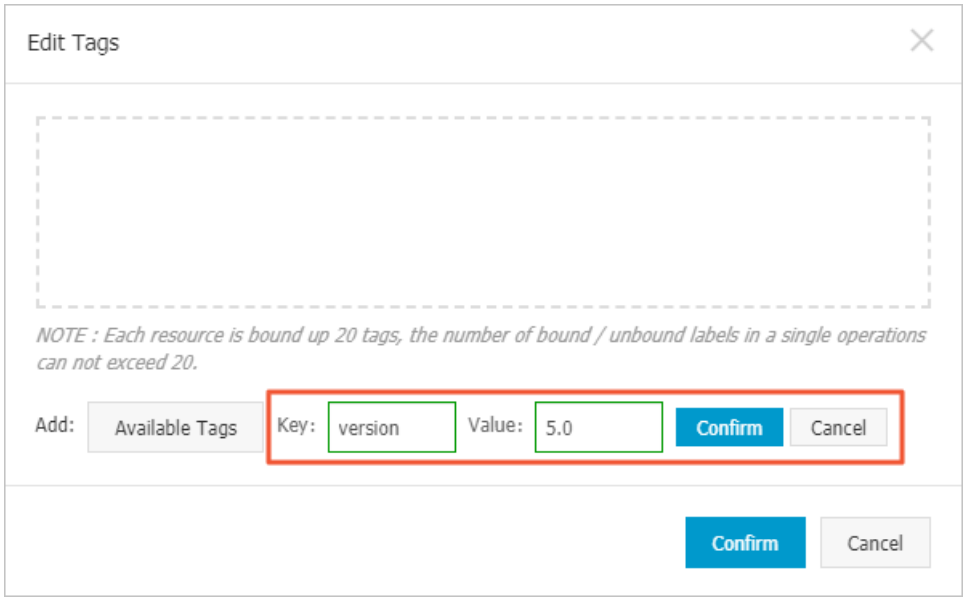
     

4. In the **Edit Tags** dialog box that appears, click **Create**.

   > **Note** If you have created a tag, you can select the tag from the Available Tags drop-down list to bind it to the selected instances.

5. Set **Key** and **Value**, and then click **Confirm**.

   Set the key and value of the tag



6. Repeat the preceding two steps to create all tags, and then click **Confirm**.

> **Note** After creating a tag, you can bind it to other instances.

Related operations

Operation	Description
TagResources	You can call this operation to bind tags to one or more ApsaraDB for Redis instances.

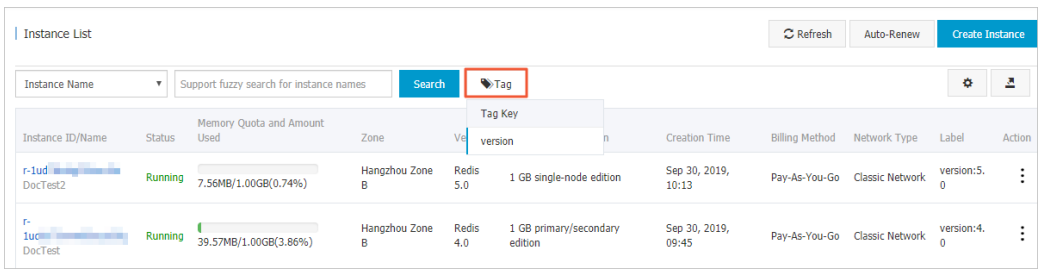
3.4.2. Filter ApsaraDB for Redis instances by tag

After binding tags to ApsaraDB for Redis instances, you can filter ApsaraDB for Redis instances by tag in the instance list to manage instances of a specific category.

Procedure

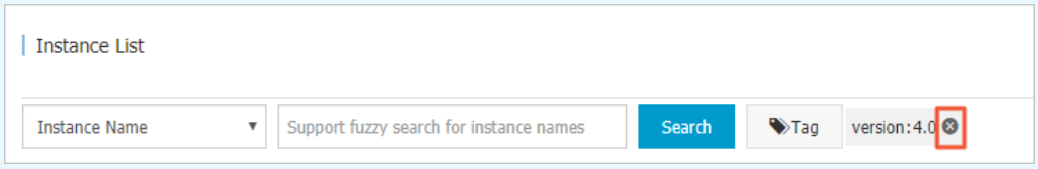
1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, click **Tag**.

   Open the tag list



4. Select the target tag key and value from **Tag Key** and **Tag Value**, respectively.

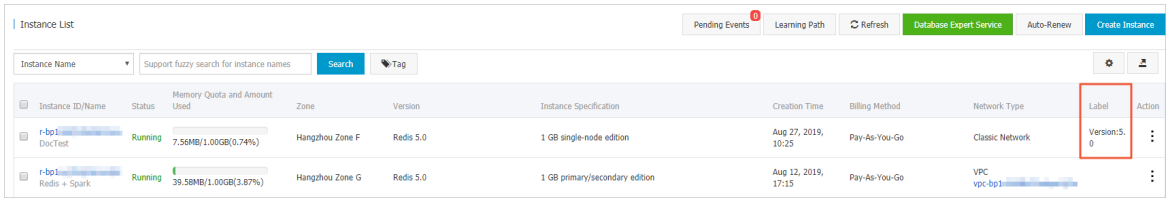
> **Note**
>
> - After creating a tag or updating an existing tag, you must refresh the page to check the updated tag in the tag list.
> - To clear a filter criterion, click the delete icon for the selected tag.



3.4.3. View tags bound to an instance

You can view the tags bound to an ApsaraDB for Redis instance on the Instance List page.

1. Log on to the ApsaraDB for Redis console.
2. In the top navigation bar, select the region where the instance resides.
3. On the **Instance List** page, find the target instance and view the tags in the **Label** column.



Related API operations

Operation	Description
ListTagResources	Queries the ApsaraDB for Redis instances that are bound to specified tags or the tags bound to specified ApsaraDB for Redis instances.

3.4.4. Unbind a tag

If an ApsaraDB for Redis instance no longer needs a tag, you can unbind the tag from the ApsaraDB for Redis instance.

> **Note** To delete a tag, see Delete a tag.

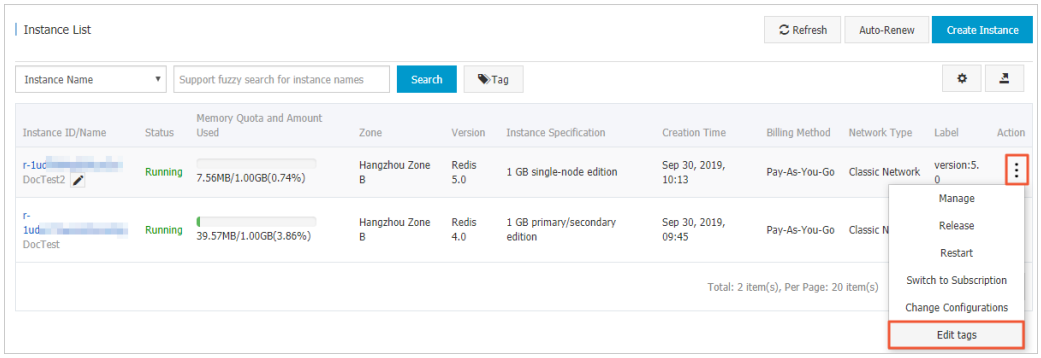
Precautions

- You can unbind a maximum of 20 tags at a time.
- After a tag is unbound from an ApsaraDB for Redis instance, the tag is automatically deleted if it is not bound to any ApsaraDB for Redis instances.

Procedure

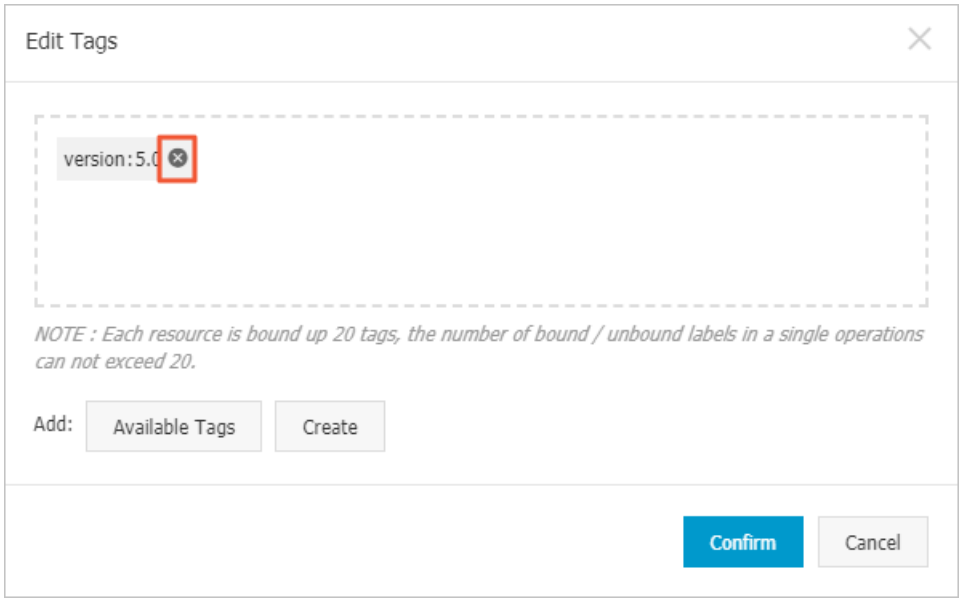
1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, choose > **Edit tags** in the **Action** column for the target instance.

   Edit tags



4. In the **Edit Tags** dialog box that appear, click the Delete icon for the target tag.

   Delete a tag



5. Click **Confirm**.

> **Note** Unbinding a tag from an ApsaraDB for Redis instance does not affect other ApsaraDB for Redis instances bound to this tag.

3.4.5. Delete a tag

You can unbind a tag from all ApsaraDB for Redis instances to delete the tag.

For more information about how to unbind a tag, see Unbind a tag.

> **Note** A tag cannot be retrieved after being deleted. If you need to use it again, you can create the same tag. For more information about how to create a tag, see Create a tag.

3.5. Set a maintenance window

You can modify the maintence window to prevent system maintence from happening in the peak period of your business.

Background

To ensure the stability of ApsaraDB for Redis instances on the Alibaba Cloud platform, ApsaraDB for Redis maintains instances and servers occasionally.

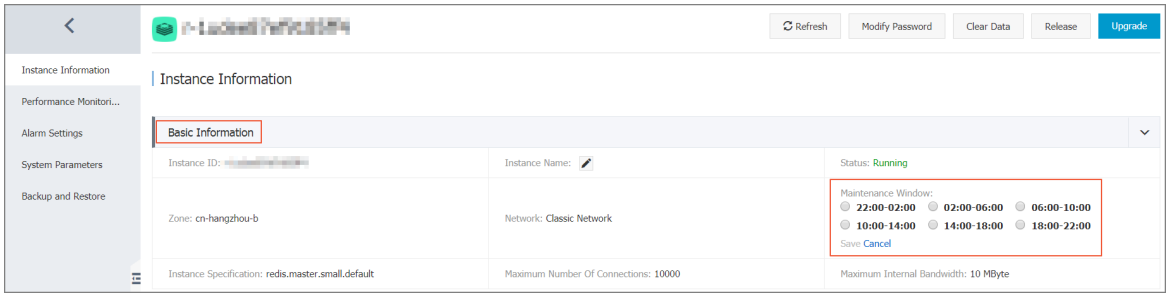
Before the maintenance, ApsaraDB for Redis sends short message service (SMS) messages and emails to contacts configured under your Alibaba Cloud account.

To guarantee the stability of the maintenance process, instances enter the **Maintaining** status before the preset maintenance window on the day of maintenance. When an instance is in this status, data in the database can still be accessed. However, change operations such as configuration change are temporarily unavailable for this instance in the console, whereas query operations such as performance monitoring are still available.

> **Notice**
>
> During the preset maintenance window, instances may be disconnected in the process of maintenance. We recommend that you set the maintenance window to a period during off-peak hours.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, click the target instance ID or **Manage** in the **Action** column for the target instance.
4. On the **Instance Information** page, click **Settings** to the right of the **Maintenance Window** field in the **Basic Information** section.



5. Select a period and click **Save**.

> **Note** The time is UTC+8.

Related API operations

ModifyInstanceMaintainTime

3.6. Temporarily adjust bandwidth

To handle burst or expected traffic peaks, you can temporarily increase the upper limit of bandwidth throttling on the Instance Information page in the console.

Context

If big keys are used or the traffic generated during peak hours exceeds the upper limit of bandwidth, Redis nodes may have network congested. This deteriorates the service performance. In this case, you can temporarily adjust the upper limit of bandwidth. This allows you to spend more time to optimize the memory architecture and fix big keys issues. You can also handle the peak traffic generated during time-limited promotions and focus on the business improvement.

Bandwidth adjustment rules

- The temporarily adjusted upper limit of bandwidth is valid for at least seven days and expires at 00:00 after seven days. For example, if you adjust the upper limit of bandwidth at 10:00 on January 1, the adjusted upper limit expires at 00:00 on January 9.

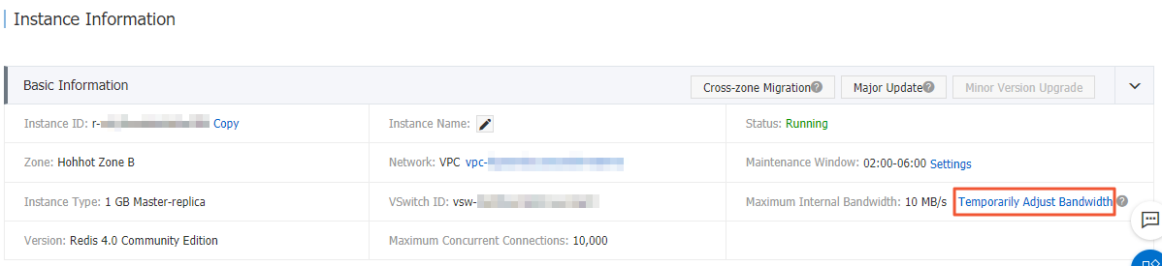
  > **Note** For more information about how to permanently adjust bandwidth, see Change specifications.

- After the adjustment, the maximum bandwidth is doubled. For example, if the upper limit of bandwidth for an instance is 10 MB/s, the upper limit is raised to 20 MB/s after the adjustment.

> **Notice** The following operations may restore the maximum internal bandwidth to the original value:
>
> - Upgrade the major version or the minor version of cluster instances.
> - Upgrade or downgrade specifications for all instances or migrate the instances across zones.

Procedure

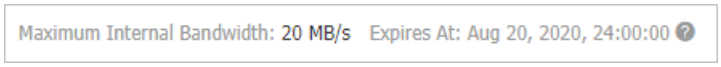
1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the section, click **Temporarily Adjust Bandwidth**.



Note For a cluster instance, the **Maximum Internal Bandwidth** shows the maximum internal bandwidth of a single node in the cluster.

Result

Refresh the page after the adjustment is completed. The temporary bandwidth and expiration time are displayed in the **Maximum Internal Bandwidth** section.



3.7. Migrate an instance across zones

This topic describes how to migrate an instance across zones in the ApsaraDB for Redis console. You may want to migrate an instance to another zone if the current zone does not have sufficient resources for instance upgrades.

Prerequisites

If an ApsaraDB for Redis instance is switched from a classic network to a Virtual Private Cloud (VPC) network and the classic network endpoint of the instance is retained, release the classic network endpoint first. For more information, see Release classic network endpoints.

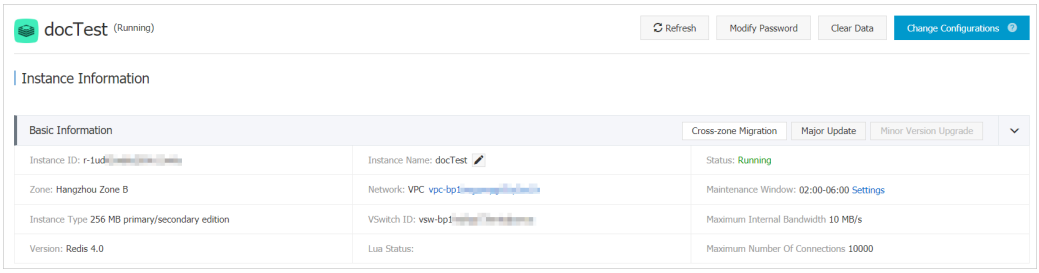
Limits

- You can only migrate an instance to zones in the region where the instance is deployed. For information about migrating an instance across regions, see Use redis-shake to migrate data.
- The migration duration depends on the data volume of the instance. During the migration, you can still connect to the instance. However, the instance may be disconnected for a few seconds. You must ensure that your applications support automatic reconnection.
- After an instance is migrated across zones, the virtual IP address (VIP) of the instance changes, but the endpoint of the instance remains unchanged. We recommend that you connect to the instance through an endpoint.

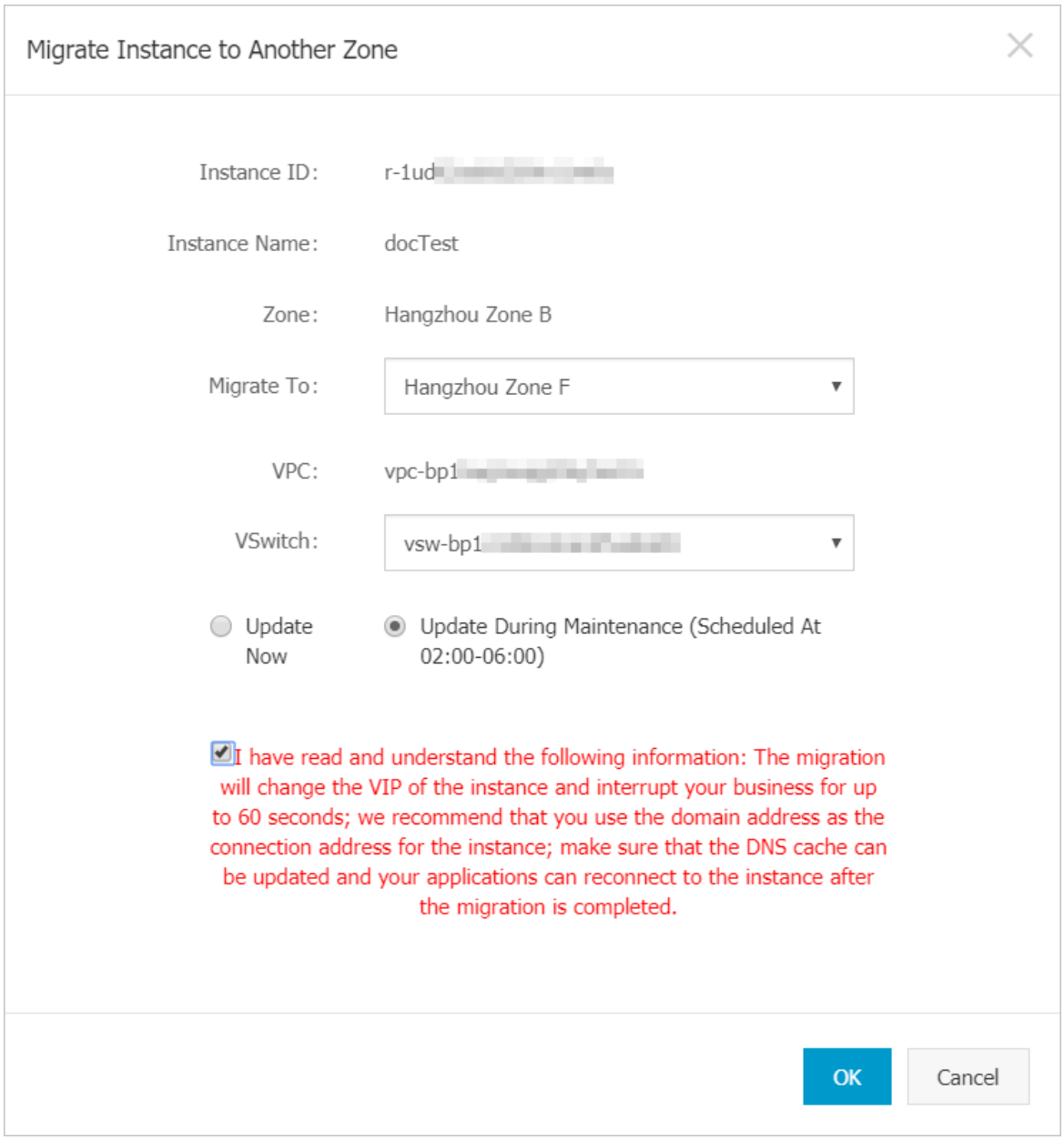
Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner, select the region where your instance is deployed.
3. On the **Instances** page, click the instance ID or click **Manage** in the **Actions** column.
4. On the **Instance Information** page, click **Cross-zone Migration** in the upper-right corner of the Basic Information section.

Cross-zone migration



5. In the **Migrate Instance to Another Zone** dialog box, select a zone and a migration time, and click **OK**.



> **Note**
>
> - If the network type of the instance is VPC, you must select a VSwitch. If no VSwitch is available in the specified zone, you must create a VSwitch first.
> - You can select **Update Now** or **Update During Maintenance**. During the migration, the instance may be disconnected for a few seconds. We recommend that you migrate an instance during off-peak hours. For more information, see Set a maintenance window.

Related API operations

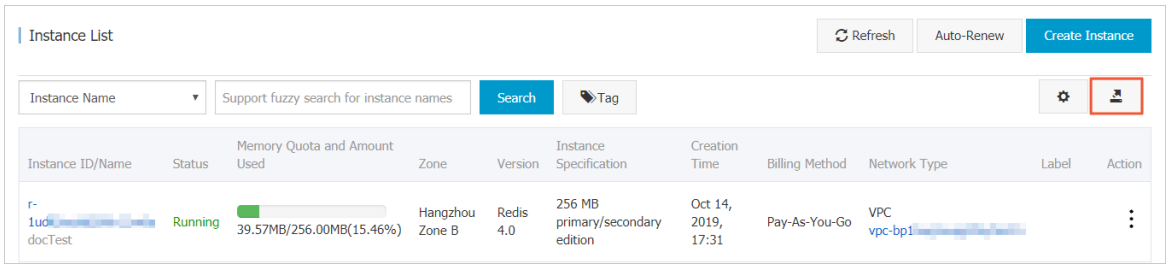
MigrateToOtherZone

3.8. Export information from the instance list

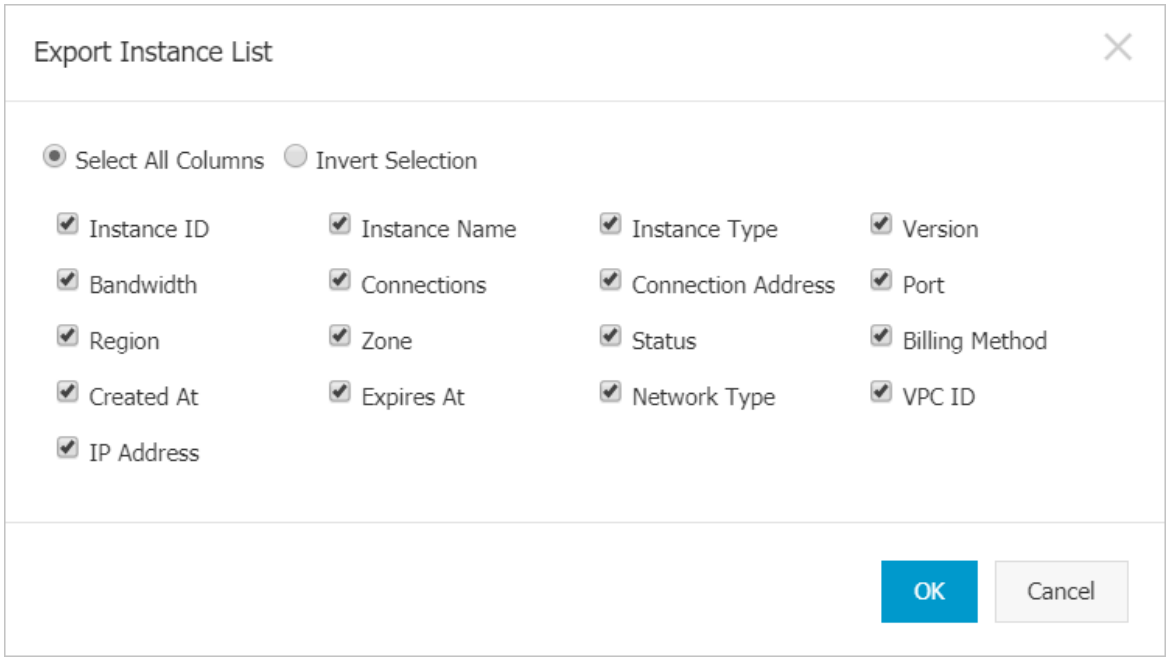
You can export an instance list from the ApsaraDB for Redis console to manage the instances offline.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, click the **Export Instances** icon in the upper-right corner.



4. In the **Export Instance List** dialog box, select the columns to be exported and click **OK**.



Note After you click **OK**, the browser starts to download an instance list in *CSV* format. You can use a text editor to open this file.

4.Security management

4.1. Manage database accounts

ApsaraDB for Redis allows you to create multiple database accounts for an instance. You can grant permissions to these accounts based on the actual usage to manage your instance and minimize user errors.

Prerequisites

The engine version of the instance is Redis 4.0 or later.

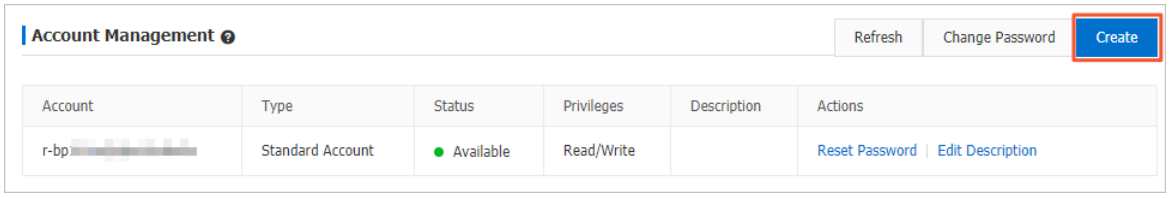
> **Note** If the engine version of an instance does not meet the requirements, you can upgrade it. For more information, see Upgrade the major version.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **Account Management**.

   > **Note** If the engine version of an instance is Redis 4.0 or later, but does not support the **Account Management** feature, you can upgrade the minor version. For more information, see Upgrade the minor version.

5. In the upper-right corner of the page, click **Create**.

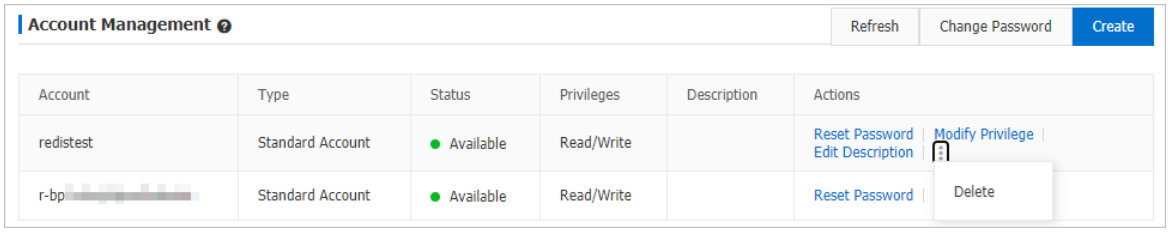


6. In the dialog box that appears, set the parameters listed in the following table.

Parameter	Description
Account	Your account must meet the following requirements: ○ The account name can contain lowercase letters, digits, underscores (_), and hyphens (-). ○ The name must be 1 to 16 characters in length. ○ The name cannot be the reserved words in the Reserved words for Redis account names section.

Parameter	Description
Privilege	The permissions granted to the account. ○ **Read-only**: The account has only the permission to read data but is not allowed to modify data. ○ **Read/Write**: The account has the permissions to read and write data. ○ **Copy**: The account has the permissions to read data, write data, and run the SYNC and PSYNC commands. **Note** Only standard instances allow you to create accounts that have the **Copy** permission.
Password Settings	The password of your account must meet the following requirements: ○ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! @#$%^&*()+-=_ ○ The password of the account must be 8 to 32 characters in length.
Confirm Password	Enter the password again.
Description	The description of an account must meet the following requirements: ○ It must start with a letter and cannot start with *http://* or *https://*. ○ The description can contain letters, Chinese characters, digits, underscores (_), and hyphens (-). ○ The description must be 2 to 256 characters in length.

7. Click **OK**.
   The newly created account is in the **Creating** state. After about one minute, the account is changed to the **Available** state.
8. (Optional)Perform the following steps to manage an account based on your business requirements:



   - Reset a password

     Click **Reset Password** in the **Actions** column of the account. In the pane that appears, reset the password and click **OK**.

   - Modify permissions

Click **Modify Privilege** in the **Actions** column of the account. In the pane that appears, select the required permission and click **OK**.

- Modify the description

  Click **Edit Description** in the **Actions** column of the account. In the pane that appears, modify the description and click **OK**.

- Delete an account

  Choose  > **Delete** in the **Actions** column of the account. In the pane that appears, click **OK**.

Reserved words for Redis account names

When you create an account, the account name cannot be one of the following reserved words. The reserved words are separated with commas (,) in the following table.

Initial	Reserved word
a~c	add,admin,all,alter,analyze,and,as,asc,asensitive,aurora,before,between,bigint,binary,blob,both,by,call,cascade,case,change,char,character,check,collate,column,condition,connection,constraint,continue,convert,create,cross,current_date,current_time,current_timestamp,current_user,cursor
d~f	database,databases,day_hour,day_microsecond,day_minute,day_second,dec,decimal,declare,default,delayed,delete,desc,describe,deterministic,distinct,distinctrow,div,double,drc_rds,drop,dual,each,eagleye,else,elseif,enclosed,escaped,exists,exit,explain,false,fetch,float,float4,float8,for,force,foreign,from,fulltext
g~l	goto,grant,group,guest,having,high_priority,hour_microsecond,hour_minute,hour_second,if,ignore,in,index,infile,information_schema,inner,inout,insensitive,insert,int,int1,int2,int3,int4,int8,integer,interval,into,is,iterate,join,key,keys,kill,label,leading,leave,left,like,limit,linear,lines,load,localtime,localtimestamp,lock,long,longblob,longtext,loop,low_priority
m~r	match,mediumblob,mediumint,mediumtext,middleint,minute_microsecond,minute_second,mod,modifies,mysql,natural,no_write_to_binlog,not,null,numeric,on,optimize,option,optionally,or,order,out,outer,outfile,precision,primary,procedure,purge,raid0,range,read,reads,real,references,regexp,release,rename,repeat,replace,replicator,require,restrict,return,revoke,right,rlike,root
s~z	schema,schemas,second_microsecond,select,sensitive,separator,set,show,smallint,spatial,specific,sql,sql_big_result,sql_calc_found_rows,sql_small_result,sqlexception,sqlstate,sqlwarning,ssl,starting,straight_join,table,terminated,test,then,tinyblob,tinyint,tinytext,to,trailing,trigger,true,undo,union,unique,unlock,unsigned,update,usage,use,using,utc_date,utc_time,utc_timestamp,values,varbinary,varchar,varcharacter,varying,when,where,while,with,write,x509,xor,xtrabak,year_month,zerofill

Related information

- Use redis-cli to connect to ApsaraDB for Redis
- CreateAccount
- DescribeAccounts
- ModifyAccountDescription

- ResetAccountPassword
- GrantAccountPrivilege
- DeleteAccount

4.2. Change the password

If you forget your password, need to change your password, or did not set a password for an instance, you can set a new password for the instance.

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the Instance List page, click the target instance ID or **Manage** in the Action column for the target instance to go to the **Instance Information** page.
4. Click Modify Password. In the **Change Password** dialog box, enter the old password and a new password, and then click **OK**.

> Notice
>
> - If you forget your password, you can click **Forgot password?** in the **Change Password** dialog box. In the **Reset Password** dialog box, you can set a new password.
> - The password must be 8 to 32 characters in length. It must consist of any three types of characters, including uppercase letters, lowercase letters, digits, and special characters. Special characters include `exclamation points (!), at signs (@), number signs (#), dollar signs ($), percent signs (%), carets (^), ampersands (&), asterisks (*), parentheses (()), underscores (_), plus signs (+), hyphens (-), and equal signs (=)`. For example, *Nm@lople0+981n-rasdoupo41xr*.

Related API operations

ModifyInstanceAttribute

4.3. Set IP address whitelists

Before you use an ApsaraDB for Redis instance, you must set one or more IP address whitelists or specify Elastic Compute Service (ECS) security groups as whitelists. This ensures data security and guarantee high performance of ApsaraDB for Redis. You can add client IP addresses or Classless Inter-Domain Routing (CIDR) blocks to a whitelist. We recommend that you update your whitelists on a regular basis to improve data security in ApsaraDB for Redis.

For more information about how to set IP address whitelists for ApsaraDB for Redis, see Step 2: Set IP address whitelists.

4.4. Add security groups

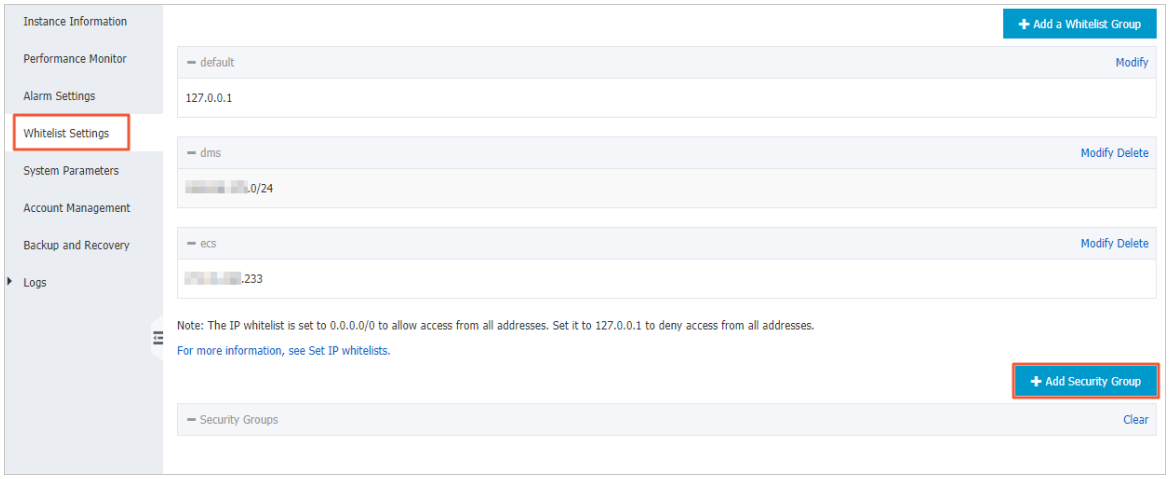
This topic describes how to add Elastic Compute Service (ECS) security groups as whitelists for an ApsaraDB for Redis instance. This facilitates the client management and enhances security of connections. A security group serves as a virtual firewall to limit the inbound and outbound network traffic of ECS instances that belong to this security group. After you specify a security group as a whitelist, all ECS instances in the security group are allowed to access the ApsaraDB for Redis instance.

Prerequisites

The ECS instances where ApsaraDB for Redis clients are deployed are added to the security group. For more information, see Add ECS instances to a security group.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the instance ID of the target instance or choose More > **Manage** in the **Actions** column for the instance.
4. In the left-side navigation pane of the **Instance Information** page, click **Whitelist Settings**.
5. On the **Whitelist Settings** page, click **Add Security Group** to add an ECS security group.



6. In the **Add Security Group** dialog box, perform the following steps:

> **Note** You can add up to 10 ECS security groups for each ApsaraDB for Redis instance.

   i. Select one or more security groups that you want to add.
   ii. Click **OK**.

4.5. Configure SSL encryption

This topic describes how to enable Secure Socket Layer (SSL) encryption to enhance security during data transmission. It also describes how to change the version of Transport Layer Security (TLS) based on business requirements.

Prerequisites

Supported instance types:

- Redis 2.8 standard master-replica instances

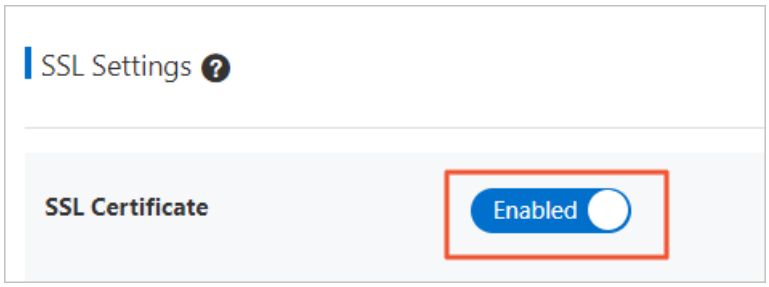
  Note You cannot change the TLS version for Redis 2.8 standard master-replica instances.

- Redis 2.8 cluster instances
- Redis 4.0 cluster instances
- Redis 5.0 cluster instances

Note SSL encryption may increase the network latency of instances. We recommend that you enable this feature only when required.

Enable SSL encryption

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **SSL Settings**.
5. On the **SSL Settings** page, turn on **SSL Certificate**.



Note

- After you enable SSL encryption, the port of ApsaraDB for Redis remains unchanged. You can establish encrypted SSL connections or connections that does not use SSL encryption.
- If you enable SSL encryption, the default TLS version is TLSv1. To change the TLS version, perform the following steps.

Change the TLS version

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **SSL Settings**.
5. On the **SSL Settings** page, click **Minimum TLS version** and select a supported TLS version from the drop-down list.

> **Note** If the **Minimum TLS version** drop-down list is disabled, upgrade the minor version of the instance. For more information, see Upgrade the minor version.

Update the SSL certificate validity

You can update the validity period of your certificate in the ApsaraDB for Redis console. The validity period is extended for one year from the date of the update.

> **Warning** If you update the validity period of an SSL certificate, the instance is restarted. During the restart process, the instance is disconnected for a few seconds. We recommend that you update the certificate during off-peak hours and make sure that your application supports automatic reconnection.

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **SSL Settings**.
5. On the **SSL Settings** page, click **Update Validity**.

> **Warning** The next step shows how to confirm whether certificate is updated. This operation immediately restarts the ApsaraDB for Redis instance. We recommend that you perform this operation during off-peak hours.

6. In the **Update SSL Certificate Validity** dialog box, click **OK**.

Download the CA certificate

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **SSL Settings**.
5. On the **SSL Settings** page, click **Download SSL Certificate**.

FAQ

What can I do if the error message "version not supported" appears?

You can upgrade the minor version of the instance. For more information, see Upgrade the minor version.

Related operations

API	Description
ModifyInstanceSSL	Enables or disables the SSL encryption of an ApsaraDB for Redis instance.

4.6. Enable password-free access

ApsaraDB for Redis supports password-free access in VPCs to achieve more convenient database connections while ensuring security. To enable password-free access, ensure that ECS instances and ApsaraDB for Redis instances are in the same VPC. This way, the ECS instances can access ApsaraDB for Redis instances without the need to use a password. At the same time, the username and password can also be used to connect to the ApsaraDB for Redis instances.

Prerequisites

- The network type of the instance is VPC.
- The client and instance are in the same VPC.
- The whitelist has been set for the instance. To secure access, you cannot add 0.0.0.0/0 to the whitelist to allow any access.

Limits

- Public endpoints can be used to access ApsaraDB for Redis 4.0 instances while VPC password-free access is enabled. In this case, you do not need to use a password to access ApsaraDB for Redis instances if an internal endpoint is used. However, you still need a password if a public endpoint is used.

  > **Note** If a public endpoint fails to access ApsaraDB for Redis 4.0 instances while VPC password-free access is enabled, upgrade the kernel version, see Upgrade the minor version.

- For ApsaraDB for Redis 2.8 or 5.0 instances, you cannot apply for public endpoints with the password-free access feature enabled. Please disable password-free access before applying for public endpoints.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. On the **Instance Information** page, find the Connection Information section and click **Enable Password-free Access**.
5. In the message that appears, click **OK**.

To disable password-free access, refresh the Instance Information page until **Disable Password-free Access** is displayed, then click it. However, applications that use the password-free access function lose connection to databases if this function is disabled.

> **Note** If your application is already connected to the instance before password-free access is enabled, reconnect it to the ApsaraDB for Redis instance for this function to take effect.

Related operations

Operation	Description
ModifyInstanceVpcAuthMode	Enable or disable password-free access.

5.Connection management

5.1. Connect to an ApsaraDB for Redis instance

You can use a Redis client, the redis-cli tool, or Data Management Service (DMS) to connect to an ApsaraDB for Redis instance.

Connection tools

Tool	Description
DMS	You can use Alibaba Cloud DMS to connect to an ApsaraDB for Redis instance and manage data that is stored on the instance.
Redis client	You can use a Redis client for the required programming language to connect to an ApsaraDB for Redis instance.
redis-cli	You can use the open source Redis tool redis-cli to connect to an ApsaraDB for Redis instance.

Connect to an instance over the internal network

You can connect to an ApsaraDB for Redis instance that does not have a public endpoint only over the internal network. Before an ApsaraDB for Redis instance is connected with an Alibaba Cloud service, such as Elastic Compute Service (ECS), make sure that the following requirements are met:

- The ECS instance and the ApsaraDB for Redis instance are in the same region. For more information, see .
- The ECS instance and the ApsaraDB for Redis instance are both in the classic network or in the same Virtual Private Cloud (VPC) network.
- The internal IP address of the ECS instance is added to the whitelist of the ApsaraDB for Redis instance.

Connect to an instance over the Internet

You can connect to an ApsaraDB for Redis instance that has a public endpoint from a local computer over the Internet.

Connect to an instance over the private endpoint

By default, ApsaraDB for Redis cluster instances are connected through a proxy. This method simplifies application code and minimizes development costs. However, the proxy-based connection may increase the response latency of ApsaraDB for Redis. To reduce the number of connections and response latency of ApsaraDB for Redis, you can enable the private endpoint and connect to the cluster instances through the private endpoint. For more information, see Enable a direct connection and Use a private endpoint to connect to an ApsaraDB for Redis instance.

Troubleshooting

- If the connection over the internal network fails, see Troubleshooting for connection issues in ApsaraDB for Redis.
- If the connection over the public network fails, see Resolve connection issues through the public network.

5.2. View endpoints

This topic describes how to view endpoints of an ApsaraDB for Redis instance in the console. The virtual IP address of an ApsaraDB for Redis instance may change during service upgrades or maintenance. We recommend that you connect to an ApsaraDB for Redis instance through an endpoint to avoid disconnection.

Connection types

ApsaraDB for Redis supports the following types of connections. An endpoint is generated for each connection type.

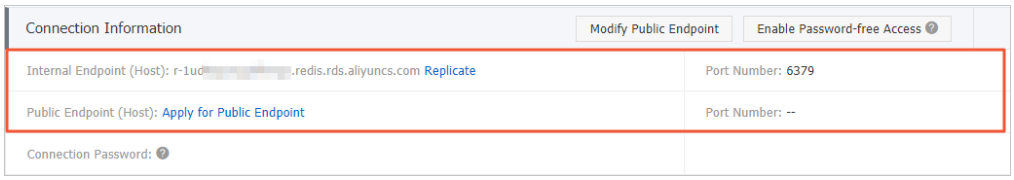
Connection type	Supported instance architecture	Description
Virtual Private Cloud (VPC)	All	You can connect to an ApsaraDB for Redis instance deployed in a VPC network if your client is also deployed in the VPC network. By default, the system generates a VPC endpoint for an ApsaraDB for Redis instance deployed in a VPC network. By default, the system generates a proxy endpoint for a cluster instance or a read/write splitting instance deployed in a VPC network. You can use this endpoint to connect to the instance through a proxy server. You can also establish a director connection if you want to connect to a cluster instance without using a proxy server.
Classic network	All	You can connect to an ApsaraDB for Redis instance deployed in a classic network if your client and the instance are in the same region. By default, the system generates a classic endpoint for an ApsaraDB for Redis instance deployed in a classic network. You cannot connect to a cluster instance deployed in a classic network by establishing a direct connection.
Proxy connection	Cluster instances and read/write splitting instances	By default, the system generates a proxy endpoint for a cluster instance or read/write splitting instance. You can use the proxy endpoint to connect to instance through a proxy server.

Connection type	Supported instance architecture	Description
Direct connection	Cluster instances	You can establish a direct connection to bypass the proxy server and access the shards of a cluster instance. The instance must be deployed in a VPC network. For more information about how to apply for a direct endpoint, see Enable a direct connection.
Public network	All	You can connect to an ApsaraDB for Redis instance over a public network. For more information about how to apply for a public endpoint, see Through the Internet.

View endpoints

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. On the **Instance Information** page or the **Connection** page, view endpoints.
   - On the **Instance Information** page, find the **Connection Information** section, and view **Internal Endpoint (Host)** and **Public Endpoint (Host)**.

     Endpoints of an ApsaraDB for Redis instance

   - On the **Connection** page, view endpoints:
     a. In the left-side navigation pane, click **Connection**.
     b. On the **Connection** page, view endpoints.

5.3. Use a private endpoint to connect to an ApsaraDB for Redis instance

After you obtain a private endpoint, you can bypass proxy servers and use the private endpoint to connect to cluster instances of ApsaraDB for Redis. This shortens the response time of ApsaraDB for Redis. This topic describes the precautions and method of using a private endpoint to connect to cluster instances of ApsaraDB for Redis. The Jedis and PhpRedis clients are used in the examples.

Prerequisites

- Direction connection to the ApsaraDB for Redis cluster instance is enabled. For more information, see Enable a direct connection.
- The client IP address is added to the whitelist of the ApsaraDB for Redis cluster instance. For more information, see Set IP address whitelists.

- A client that supports Redis Cluster is used, such as Jedis and PhpRedis.

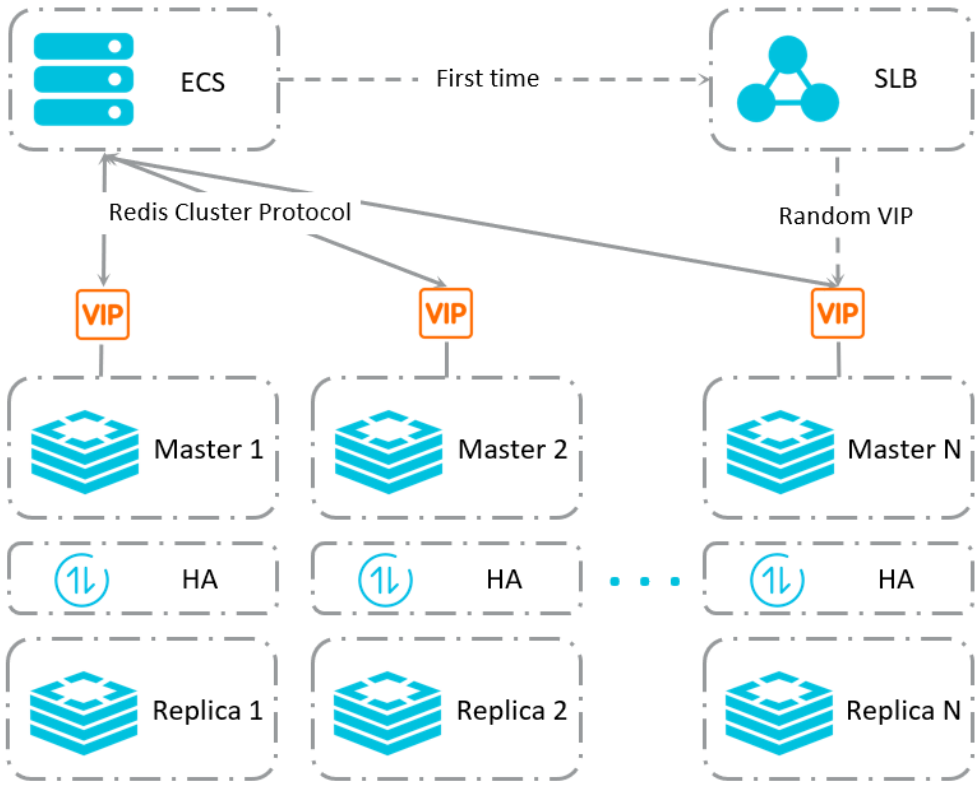
  > **Note**
  > - If you use a client that does not support Redis Cluster, you may fail to obtain data because the client cannot redirect your request to the correct shard.
  > - Jedis uses the JedisCluster class to support Redis Cluster. For more information, see Jedis documentation.
  > - You can obtain a list of clients that support Redis Cluster in the client list on the Redis official website.

- The Elastic Compute Service (ECS) instance on which the Redis client is deployed and the target ApsaraDB for Redis instance are connected to the same virtual private cloud (VPC).

Context

When you enable direct connection, ApsaraDB for Redis allocates a virtual IP (VIP) address to the master node of each data shard in the ApsaraDB for Redis cluster. Before a client sends the first request to a private endpoint, the client uses a domain name server (DNS) to resolve the private endpoint. The resolution result is the VIP address of a random data shard in the cluster. The client can use this VIP address to manage the data of the ApsaraDB for Redis cluster over the Redis Cluster protocol. The following figure shows the service architecture of an ApsaraDB for Redis cluster instance in the direct connection mode.

Service architecture of an ApsaraDB for Redis cluster instance in the direct connection mode



Precautions

- Private endpoints can be accessed through the Alibaba Cloud internal network only.
- In the direct connection mode, you can use the **SELECT** command to switch databases. However, some Redis Cluster clients, such as stackExchange.redis, do not support the **SELECT** command. If you are using these clients, you can only use database 0.
- If you use multi-key commands, transactions, or Lua scripts to modify multiple keys, ensure that these keys are stored in the same hash slot.

  > **Note** Multi-key commands include DEL, SORT, MGET, MSET, BITOP, EXISTS, MSETNX, RENAME, RENAMENX, BLPOP, BRPOP, RPOPLPUSH, BRPOPLPUSH, SMOVE, SUNION, SINTER, SDIFF, SUNIONSTORE, SINTERSTORE, SDIFFSTORE, ZUNIONSTORE, ZINTERSTORE, PFMERGE, and PFCOUNT.

- The direct connection mode supports password-free access in a VPC and username and password authentication.

Sample code for connecting Jedis to the instance

> **Note** For more information about how to use Jedis, see the Jedis documentation.

- Connections based on the default connection pool

```
import redis.clients.jedis.HostAndPort;
import redis.clients.jedis.JedisCluster;
import redis.clients.jedis.JedisPoolConfig;

import java.util.HashSet;
import java.util.Set;

public class DirectTest  {
   private static final int DEFAULT_TIMEOUT = 2000;
   private static final int DEFAULT_REDIRECTIONS = 5;
   private static final JedisPoolConfig DEFAULT_CONFIG = new JedisPoolConfig();

   public static void main(String args[]){

      // The private endpoint that is assigned to you when you Enable a direct connection.
      String host = "r-bp1xxxxxxxxxxx.redis.rds.aliyuncs.com";
      int port = 6379;
      String password = "xxxx";

      Set<HostAndPort> jedisClusterNode = new HashSet<HostAndPort>();
      jedisClusterNode.add(new HostAndPort(host, port));

      JedisCluster jc = new JedisCluster(jedisClusterNode, DEFAULT_TIMEOUT, DEFAULT_TIMEOUT,
            DEFAULT_REDIRECTIONS,password, "clientName", DEFAULT_CONFIG);

      jc.set("key","value");
      jc.get("key");

      jc.close();
   }
}
```

- Connections based on a custom connection pool

```
import redis.clients.jedis.*;

import java.util.HashSet;
import java.util.Set;
public class main {
   private static final int DEFAULT_TIMEOUT = 2000;
   private static final int DEFAULT_REDIRECTIONS = 5;
   private static final JedisPoolConfig DEFAULT_CONFIG = new JedisPoolConfig();
```

```
    private static final JedisPoolConfig DEFAULT_CONFIG = new JedisPoolConfig();

    public static void main(String args[]){
        JedisPoolConfig config = new JedisPoolConfig();
        // The maximum number of idle connections. Set this parameter as required. The value cannot e
xceed the maximum number of connections supported by the specific instance type.
        config.setMaxIdle(200);
        // The maximum number of connections. Set this parameter as required. The value cannot exce
ed the maximum number of connections supported by the specific instance type.
        config.setMaxTotal(300);
        config.setTestOnBorrow(false);
        config.setTestOnReturn(false);

        // The private endpoint that is assigned to you when you Enable a direct connection.
        String host = "r-bp1xxxxxxxxxxx.redis.rds.aliyuncs.com";
        int port = 6379;
        // The password of the instance.
        String password = "xxxxx";

        Set<HostAndPort> jedisClusterNode = new HashSet<HostAndPort>();
        jedisClusterNode.add(new HostAndPort(host, port));
        JedisCluster jc = new JedisCluster(jedisClusterNode, DEFAULT_TIMEOUT, DEFAULT_TIMEOUT,
            DEFAULT_REDIRECTIONS,password, "clientName", config);

        try {
            jc.set("foo", "bar");
            String foobar = jc.get("foo");
            jc.zadd("sose", 0, "car");
            jc.zadd("sose", 0, "bike");
            Set<String> sose = jc.zrange("sose", 0, -1);
        } finally {
            if (jc ! = null) {
                jc.close();
            }
        }
    }
}
```

Sample code for connecting PhpRedis to the instance

Note For more information about how to use PhpRedis, see the PhpRedis documentation.

```
<? php
// The private endpoint and the connection port.
$array = ['r-bp1xxxxxxxxxxx.redis.rds.aliyuncs.com:6379'];
// The password for the connection.
$pwd = "xxxx";

// Use the password to connect to the cluster instance.
$obj_cluster = new RedisCluster(NULL, $array, 1.5, 1.5, true, $pwd);

// The result of the connection.
var_dump($obj_cluster);

if ($obj_cluster->set("foo", "bar") == false) {
   die($obj_cluster->getLastError());
}
$value = $obj_cluster->get("foo");
echo $value;
? >
```

6.Data management

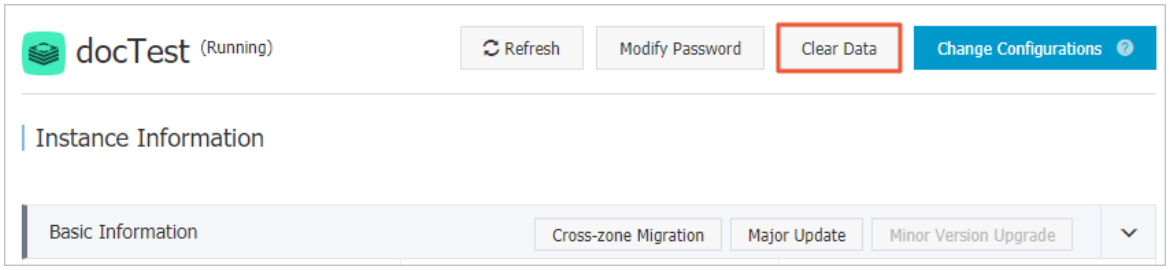
6.1. Clear data

You can clear the data of an ApsaraDB for Redis instance in the console.

> **Warning** This operation clears all the data of an instance. The data cannot be recovered. We recommend that you exercise caution when performing this operation.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. In the upper-left corner of the top navigation bar, select the region where the target instance is located.
3. On the **Instance List** page, click the target instance ID or **Manage** in the **Action** column for the target instance.
4. On the **Instance Information** page, click **Clear Data**.



5. In the **Clear Data** dialog box, select the data to be cleared:
   - **All Data**: All the data stored in the instance.
   - **Expired Data**: All the expired data stored in the instance. You can select the time to clear the expired data.

   > **Warning** Data is cleared once the operation is executed. It is fatal to online businesses if the operation is not executed properly. Please do it with caution. If you must use this feature, it is recommended to execute the clearing during off-peak hours.

6. Click **OK**.

Related API operations

API	Description
FlushInstance	Call FlushInstance to clear data of an ApsaraDB for Redis instance.
FlushExpireKeys	Call FlushExpireKeys to clear expired keys in an Apsaradb for Redis instance.

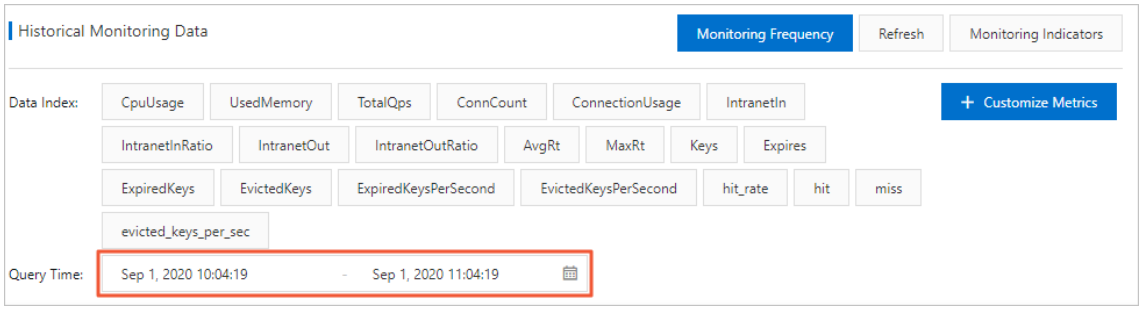
7.Performance monitoring

7.1. Query monitoring data

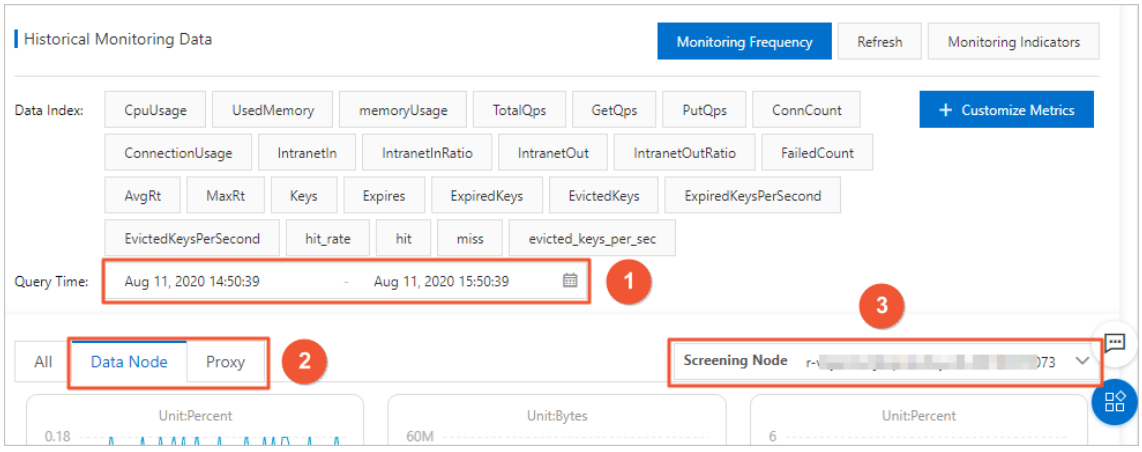
This topic describes how to query the monitoring data of an ApsaraDB for Redis instance within a specified period of the previous month.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **Performance Monitor**.
5. Perform the following steps based on the architecture type of the instance:
   - Standard master-replica instances: specify the query time.



   - Cluster or read/write splitting instances: specify the query time, node type (**Data Node** or **Proxy**), and the node ID.



> **Note**
>
> - By default, the **Performance Monitor** page displays the metrics of the basic monitoring group. You can click **Customize Metrics** to view the metrics of other monitoring groups. For more information, see Customize metrics.
> - For more information about the metrics, see Monitoring metrics.

Related operations

API	Description
DescribeMonitorItems	Queries the metrics of an ApsaraDB for Redis instance.
DescribeHistoryMonitorValues	Queries historical monitoring data of an ApsaraDB for Redis instance.

7.2. Customize metrics

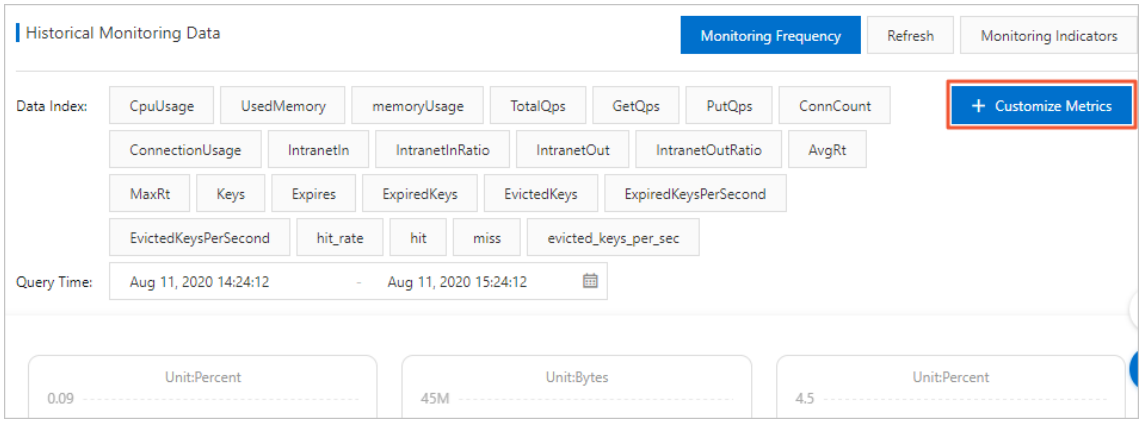
This topic describes how to select the metrics to be displayed on the Performance Monitor page of the ApsaraDB for Redis console as needed.

Context

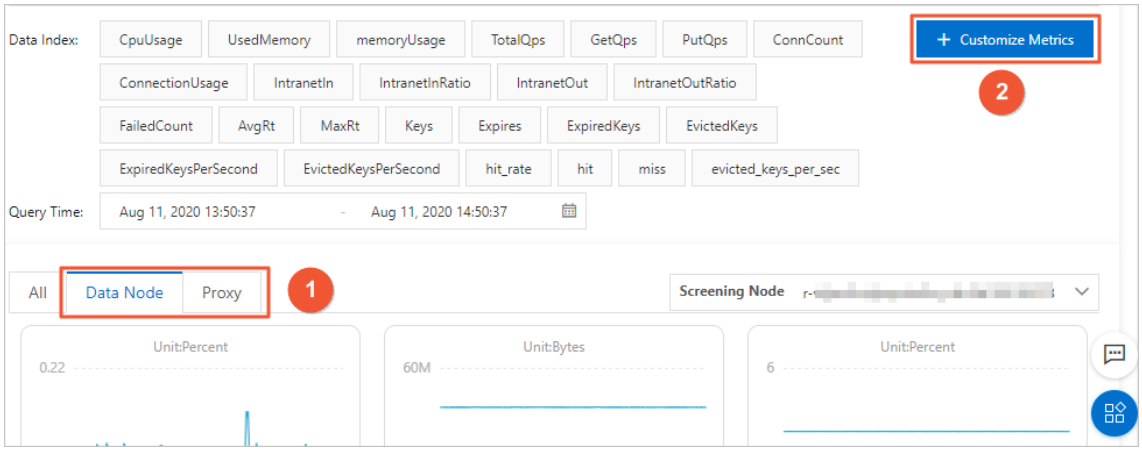
ApsaraDB for Redis supports more than 10 groups of monitoring metrics. By default, the Performance Monitor page displays the metrics of the basic monitoring group. You can click **Customize Metrics** to select other monitoring groups. For more information, see Monitoring metrics.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **Performance Monitor**.
5. Perform the following steps based on the architecture type of the instance:
   - Standard master-replica instances: on the right side of the page, click **Customize Metrics**.



   - Cluster or read/write splitting instances: click **Data Node** or **Proxy** based on the node type, and click **Customize Metrics** on the right side of the page.



6. In the dialog box that appears, select the target monitoring group.
7. Click **OK**.

> **Note**
> - For more information, see Monitoring metrics.
> - If you use read/write splitting or cluster instances, you can select the target node from the **Screening Node** drop-down list.

7.3. Monitoring metrics

ApsaraDB for Redis monitors more than 10 groups of metrics in real time. This allows you to monitor the status of ApsaraDB for Redis instances. This topic describes the purpose of each metric.

Monitoring frequency and monitoring cycle

The frequency that the performance monitoring system collects monitoring data is the monitoring frequency. If the system collects data every 5 seconds, the monitoring frequency is once every 5 seconds, and the monitoring cycle (or collection cycle) is 5 seconds. For more information about how to modify the monitoring frequency, see Modify the monitoring frequency.

> **Note** The default monitoring frequency is once every 60 seconds.

Aggregated metrics

By default, the aggregated metrics of read/write splitting or cluster instances are displayed on the **Performance Monitor** page. You can also click **Data Node** or **Proxy** to view the metrics in different monitoring groups of each node. For more information, see Custom monitoring groups and custom metrics.

> **Note** ApsaraDB for Redis will support more monitoring metrics in the future.

Metric	Unit	Description
CpuUsage	%	The average CPU usage of all data nodes.

Metric	Unit	Description
UsedMemory	Bytes	The total memory used by all data nodes. This includes the memory consumed by the data and cache.
Keys	Counts	The total number of keys on all data nodes.
Expires	Counts	The total number of keys for which the expiration time is set.
ExpiredKeys	Counts	The total number of expired keys on all data nodes.
EvictedKeys	Counts	The total number of keys that are evicted on all data nodes.
HitRate	%	The average hit ratio of keys for all data nodes. Hit ratio = (Total key hits)/(Total key hits + Total key misses).

Custom monitoring groups and custom metrics

You can view metrics in each monitoring group for data nodes or proxy nodes. For more information, see Customize metrics. The following table lists the monitoring groups.

Monitoring group	Description
Basic Monitoring Group	The basic monitoring metrics of an instance, such as queries per second (QPS), bandwidth, and memory usage. For more information, see Metrics in the basic monitoring group.
Key Monitoring Group	The metrics on the use of key-value related commands, such as the number of times DEL and EXITS are run.
String Monitoring Group	The metrics on the use of string-related commands, such as the number of times APPEND and MGET are run.
Hash Monitoring Group	The metrics on the use of hash-related commands, such as the number of times HGET and HDEL are run.
List Monitoring Group	The metrics on the use of list-related commands, such as the number of times BLPOP and BRPOP are run.
Set Monitoring Group	The metrics on the use of set-related commands, such as the number of times SADD and SCARD are run.
Zset Monitoring Group	The metrics on the use of zset-related commands, such as the number of times ZADD and ZCARD are run.
HyperLog Monitoring Group	The metrics on the use of HyperLogLog-related commands, such as the number of times PFADD and PFCOUNT are run.
Pub/Sub Monitoring Group	The metrics on the use of publication and subscription-related commands, such as the number of times PUBLISH and SUBSCRIBE are run.

Monitoring group	Description
Transaction Monitoring Group	The metrics on the use of transaction-related commands, such as the number of times WATCH, MULTI, and EXEC are run.
Lua Script Monitoring Group	The metrics on the use of Lua script-related commands, such as the number of times EVAL and SCRIPT are run.
TairDoc monitoring group (for Enhanced Edition only)	The metrics on the use of TairDoc commands, such as the number of times JSON.SET and JSON.GET are run.
TairHash monitoring group (for Enhanced Edition only)	The metrics on the use of TairHash commands, such as the number of times EXHSET and EXHMSET are run.
TairString monitoring group (for Enhanced Edition only)	The metrics on the use of TairString commands, such as the number of times EXSET and EXGET are run.
TairGis monitoring group (for Enhanced Edition only)	The metrics on the use of TairGis commands, such as the number of times GIS.ADD, GIS.GET, and GIS.DEL are run.
TairBloom monitoring group (for Enhanced Edition only)	The metrics on the use of TairBloom commands, such as the number of times BF.RESERVE and BF.ADD are run.

The preceding monitoring groups can be classified into **Basic Monitoring Group** and other monitoring groups.

- Metrics in the basic monitoring group

Metrics in the basic monitoring group

Type	Metric	Unit	Description
CPU	CpuUsage	%	The CPU usage.
Memory	UsedMemory	Bytes	The amount of used memory, which includes the memory consumed by the data and cache.
	MemoryUsage	%	The memory usage. **Note** Redis 2.8 instances are not supported. To display this metric, you must upgrade the engine version of the instance. For more information, see Upgrade the major version.
	Keys	Counts	The total number of keys.

Type	Metric	Unit	Description
	ExpiredKeysPerSecond	Counts/s	The number of expired keys per second.
	EvictedKeysPerSecond	Counts/s	The number of evicted keys per second.
Disk (only for hybrid-storage instances)	DataSize	MB	The size of data files.
	LogSize	MB	The size of logs.
Request	TotalQps	Counts/s	The total number of requests per second, which includes read and write requests.
	GetQps	Counts/s	The number of read requests per second. Note The engine version of the ApsaraDB for Redis instance must be Redis 4.0 or later with the latest minor version. For more information, see Upgrade the major version and Upgrade the minor version.
	PutQps	Counts/s	The number of write requests per second. Note The engine version of the ApsaraDB for Redis instance must be Redis 4.0 or later with the latest minor version. For more information, see Upgrade the major version and Upgrade the minor version.

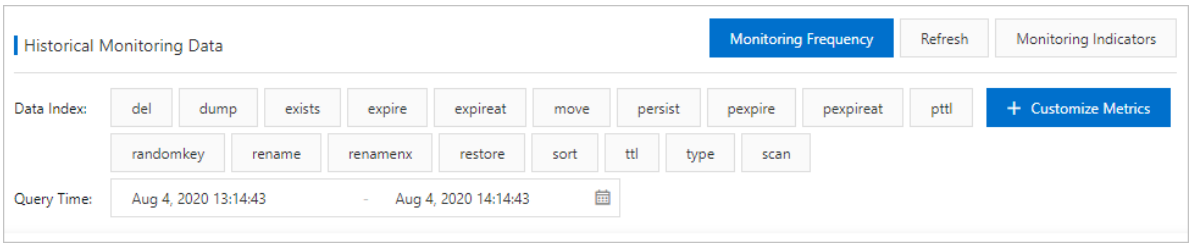
Type	Metric	Unit	Description
Network	UsedConnection	Counts	The number of used connections. This is also the number of Transmission Control Protocol (TCP) connections to the instance. Note If you use a cluster instance and connect to the instance by using the proxy endpoint, you need to pay attention to the number of connections on each proxy node.
	ConnectionUsage	%	The connections usage. Connections usage = Number of connections/Total number of connections supported by the instance.
	IntranetIn	KBps	The inbound traffic of the instance.
	IntranetInRatio	%	The usage rate of the inbound traffic.
	IntranetOut	KBps	The outbound traffic of the instance.
	IntranetOutRatio	%	The usage rate of the outbound traffic.
Latency	AvgRt	us	The average response time that is consumed by all commands. This is the average time period from the time when the data node receives the command to the time when the data node returns the result.
	MaxRt	us	The maximum response time of requests. This is the maximum response time at which a data node sends a response after it receives a command.
Response	hit	Counts	The number of keys that are hit per second.
	miss	Counts	The number of keys that are missed per second.

Type	Metric	Unit	Description
	HitRate	%	The hit ratio of keys. Hit ratio = (Total key hits)/(Total key hits + Total key misses).

- Other monitoring metrics

  You can also click **Customize Metrics** to view metrics of other monitoring groups. These metrics show the number of times the related commands are run. To understand what information a metric provides, you only need to identify the metric category based on the metric name.

  For example, the **del**, **dump**, and **exists** metrics in the **Key Monitoring Group** specify the number of times the DEL, DUMP, and EXISTS commands are run.



7.4. Alert settings

You can set alert rules for an ApsaraDB for Redis instance based on the performance monitoring data of the instance. When an alert is triggered by the performance status change of a Redis instance, alert notifications will be sent to the specified contacts.

Redis monitoring and CloudMonitor

CloudMonitor can monitor Apsaradb for Redis instances and generate alerts. CloudMonitor is a service that monitors Alibaba Cloud resources and Internet applications. It is an all-in-one, out-of-the-box, enterprise-ready monitoring solution. You can create alert rules and set metrics. When an alert rule is triggered by the change of monitored metrics, CloudMonitor notifies all the contacts in the alert contact group.

CloudMonitor help information

- For more information about Redis-related metrics, see ApsaraDB for Redis.
- For more information about how to create alert rules, see Create a threshold-triggered alert rule.
- For more information about how to create alert contacts and contact groups, see Create an alert contact or alert group.

8.Log management

8.1. Audit logs (new version)

8.1.1. Enable the new version of the audit log feature

ApsaraDB for Redis with Alibaba Cloud Log Service has released a new version of the audit log feature. This version can be used to query log data, analyze logs online, and export logs. This allows you to locate security and performance issues of your ApsaraDB for Redis instances at the earliest opportunity.

Prerequisites

- Either of the following instances is used: ApsaraDB for Redis instances of Community Edition and performance-enhanced instances of Enhanced Edition.
- The engine version of the ApsaraDB for Redis instance is Redis 4.0 or later.
- If you want to enable the new version of the audit log feature by using a RAM user, make sure that the RAM user is granted the AliyunLogFullAccess permission. For more information, see Grant permissions.

Context

Alibaba Cloud Log Service is an all-in-one service and has been used in big data analytics scenarios. Log Service can be used to collect, consume, ship, search, and analyze log data without the need for extra code resources. This service improves O&M efficiency. ApsaraDB for Redis integrates with the features of Log Service to provide an audit log feature that is stable and flexible, simple, and efficient.

You must follow the procedure in this topic to enable the new version of the audit log feature.

Pricing

The new version of the audit log feature is provided in the free trial and official versions.

> **Note** The free trial version is available. The official version will be available soon.

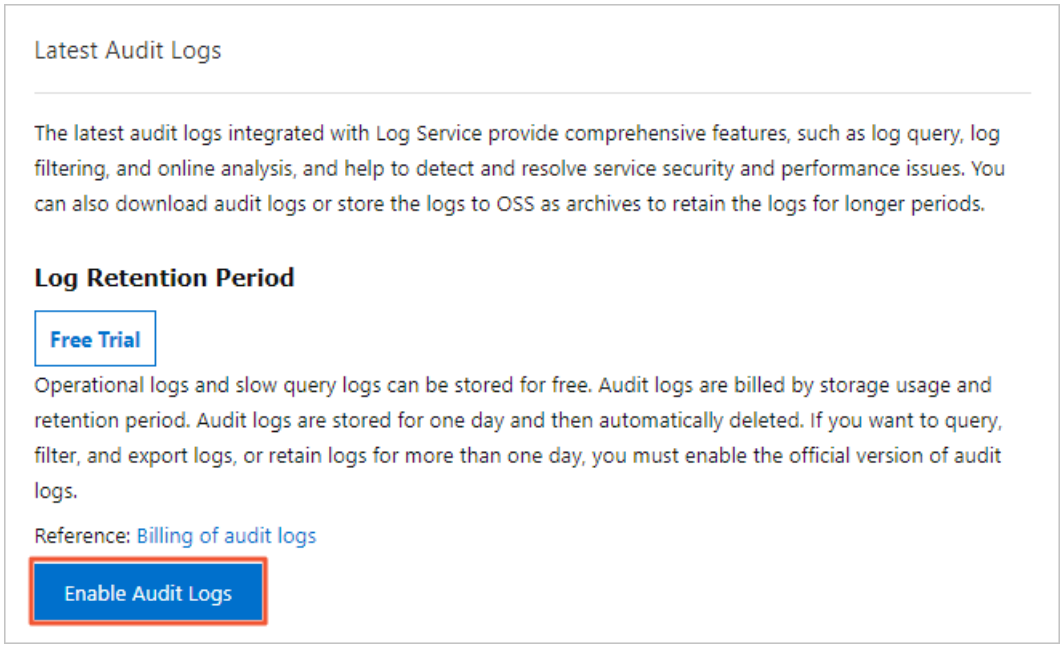
- In the free trial version, audit logs can be stored for one day. You can store up to 100 GB of audit logs.
- The official version is charged based on the size of audit logs and retention period. The official version supports more features than the free trial version.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The **Instance Information** page appears. In the left-side navigation pane, choose **Logs** >

Audit Log.

5. Click **Enable Audit Logs.**



6. In the **Enable Audit Logs** message, click **OK.**

FAQ

- Q: Where can I view the documentation for earlier versions of the audit log feature?

  A: For more information, see Enable an earlier version of the audit log feature.

- Q: Why do I fail to enable the new version of the audit log feature?

  A:

  - Make sure that you have met the prerequisites described in this topic.
  - Try to upgrade your instance to the latest minor version. For more information, see Upgrade the minor version.

What's next

- Query slow logs
- Query audit logs

8.1.2. Query audit logs

This topic describes how to query audit logs of an ApsaraDB for Redis instance. In the ApsaraDB for Redis console, you can view audit logs in a specified time range and filter audit logs that match specified conditions.

Prerequisites

The new version of the audit log feature is enabled. For more information, see Enable the new version of the audit log feature.

Background

Audit logs provide a detailed insight into the status of your ApsaraDB for Redis instance. You can use audit logs to view request records. This allows you to check records of modify and delete operations and find the cause of sudden increases in database resource consumption.

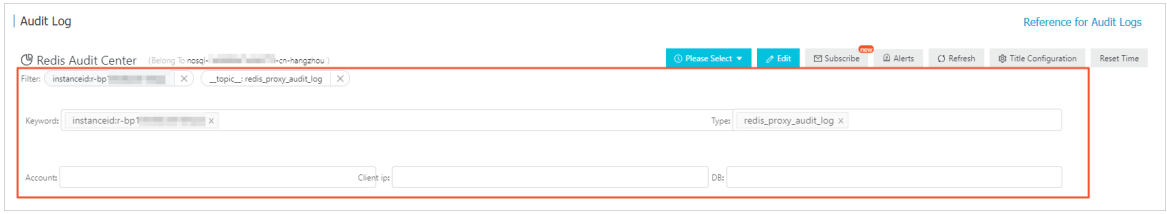
View audit logs

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. On the **Audit Log** page, you can check the audit log details for the instance.

Filter audit logs

You can define different conditions to filter audit logs.

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. On the **Audit Log** page, you can define conditions to filter audit logs.



Filter conditions

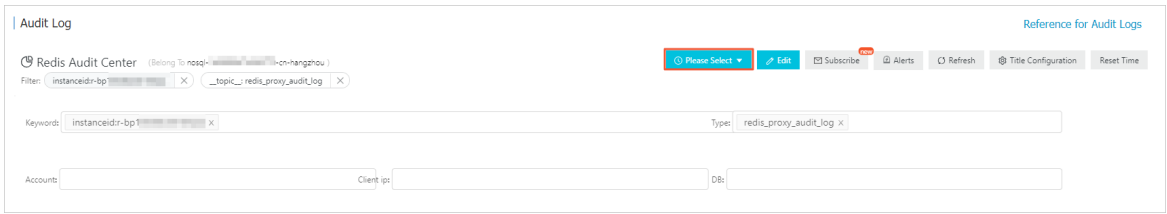
Filter condition	Description

Filter condition	Description
Keyword	Filters audit logs by keyword, such as the client IP address, executed commands, accounts, and extended information. **Note** ○ The Keyword filed supports exact match. You must enter complete information in this field. ■ For example, you must enter a complete IP address, such as 192.168.1.1, instead of 192.168 or 1.1. ■ You must enter a complete command, such as AUTH or auth, instead of au. ○ You must enclose keywords that contain colons within double quotation marks (""), for example, *"userId:1"*.
Type	The type of audit logs. Valid values: ○ redis_audit_log: the audit log of a data shard. ○ redis_proxy_audit_log: the audit log of a proxy server.
Account	The account used to connect to the ApsaraDB for Redis instance. Default value: null.
Client ip	The client IP address used to connect to the ApsaraDB for Redis instance.
DB	The database of which you want to query the audit logs.

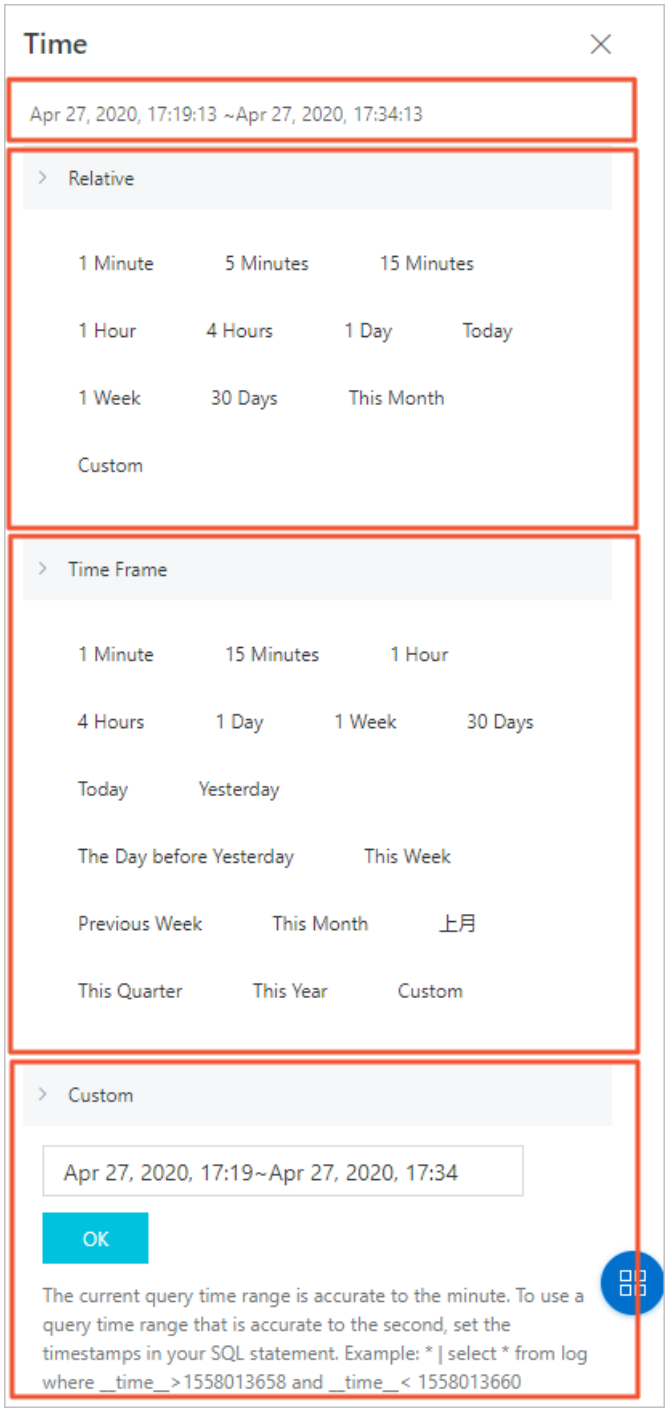
View audit logs within a specified time range

You can view audit logs within a specified time range by using the time picker.

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. On the **Audit Log** page, click **Please Select**.



6. In the Time pane, specify the time range to query audit logs.



Sections in the Time pane

No.	Section name	Description
1	Time Range	Information about a specified time range appears in this section when you place the pointer over a relative time range or a time frame.

No.	Section name	Description
2	Relative	A time range relative to the current point in time. Information about the specified time range appears in the Time Range section when you place the pointer over any element in this section.
2	Time Frame	A time range that is more than one minute. Information about the specified time range appears in the Time Range section when you place the pointer over any element in this section.
4	Custom	A custom time range. Specify a time range and click **OK** to confirm the time range.

FAQ

- Q: I can only view 2,000 audit log entries in total. How can I view the others?

  The Audit Log page in the ApsaraDB for Redis console displays up to 2,000 audit log entries. To view more audit log entries, log on to the Log Service console. For more information, see Query logs.

- Q: Where can I view the documentation for earlier versions of the audit log feature?

  A: For more information, see Query audit logs for earlier versions.

8.1.3. Query the history of hot keys

In an ApsaraDB for Redis database, the keys that are frequently accessed are called hot keys. The improper management of hot keys may cause Redis processes to be blocked and result in service interruption. You can use the Audit Log feature to query the history of hot keys and analyze the history. This allows you to further optimize the database.

Prerequisites

Enable the new version of the audit log feature

Context

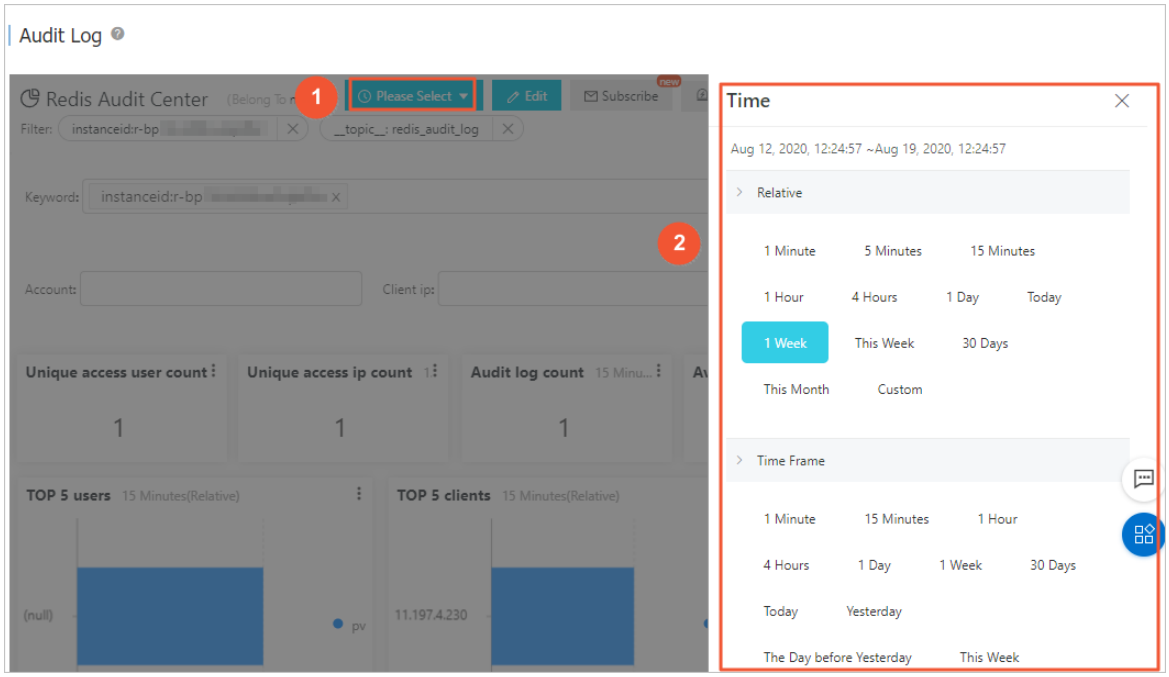
Based on the Least Frequently Used (LFU) algorithm, ApsaraDB for Redis uses efficient sorting and statistical algorithms to identify the hot keys on an instance.

> **Note** If the number of queries per second (QPS) of a key is greater than 3,000, the key is regarded as a hot key.

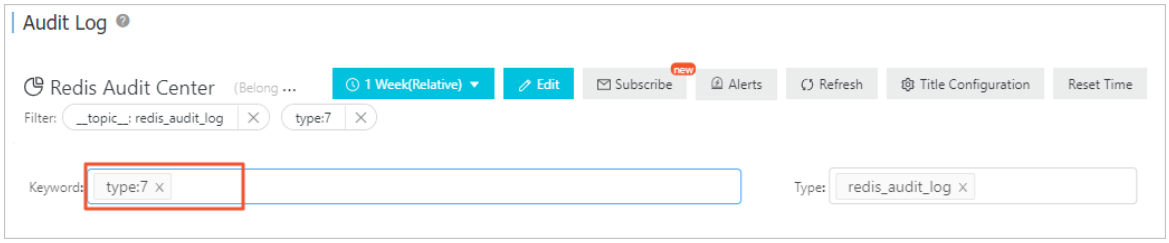
Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.

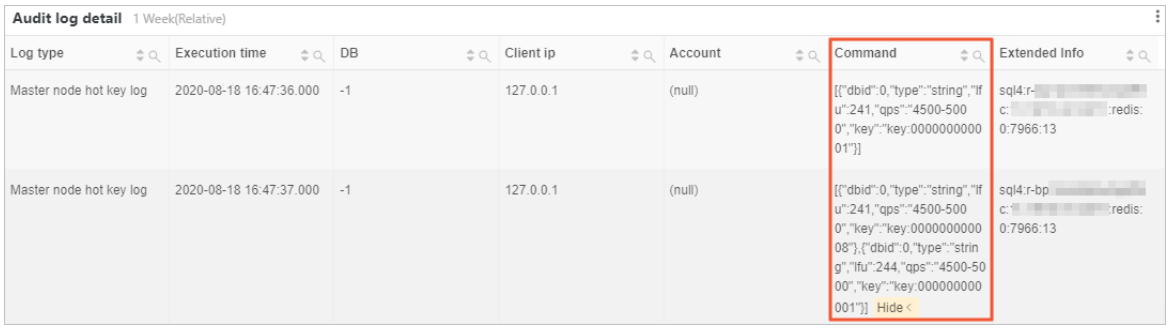
5. On the **Audit Log** page, click **Please Select** in the upper-right corner of the page. In the Time pane, specify a time range that you want to query. In this example, **1 Week** is selected to query the history that was generated in the last week.



6. Clear the **Keyword** field, enter *type:7* in the field, and then press Enter.



7. The **Audit log detail** section displays the history of hot keys that are returned.



> **Note** The **Client ip** column displays 127.0.0.1 to indicate the IP address of the local host of the ApsaraDB for Redis instance.

The **Command** column displays the details of the hot keys that are found. The following table describes the fields in each command.

Field	Example	Description
dbid	"dbid":0	The database in which the hot key is located.
type	"type":"string"	The type of data structure that the hot key uses.
lfu	"lfu":241	The LFU value of the hot key.
qps	"qps":"4500-5000"	The QPS of the hot key. The value is displayed as a range.
key	"key":"key:000000000008"}	The hot key that is returned.

8.1.4. Download audit logs

You can download the queried audit logs to your local device, and then archive, filter, or analyze the logs.

Prerequisites

You have activated the new version of the audit logs service. For more information, see Enable the new version of the audit log feature.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane of the **Instance Information** page, choose **Logs > Audit Log**.
5. In the log chart area, click the  > **Download Log** button in the upper-right corner of the target chart.

   Note You can filter logs by using the following methods. Then, you can download the content that meets your requirements.
   - Filter log data by keyword, type, account, or client IP address.
   - Filter logs by the time when logs are generated. Click **Select Time Range** above the **Download Log** button to select a time range.

   After you click **Download Log**, the selected log entries are saved to a *.csv* file on your local device through the web browser. Then, you can view the log data by using tools such as Excel.

	A	B	C
1	command	pv	
2	select 0	586	
3	PING	572	
4	INFO ALL	571	
5	AUTH ******	148	
6	FLUSHDB	4	
7			

8.1.5. Subscribe to audit log reports

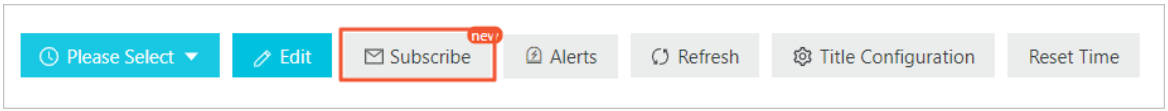
This topic describes how to subscribe to audit log reports of ApsaraDB for Redis by using emails or the DingTalk ChatBot. This allows you to periodically check the status of an ApsaraDB for Redis instance.

Prerequisites

You activate the new version of Log Service. For more information, see Enable the new version of the audit log feature.

Procedure

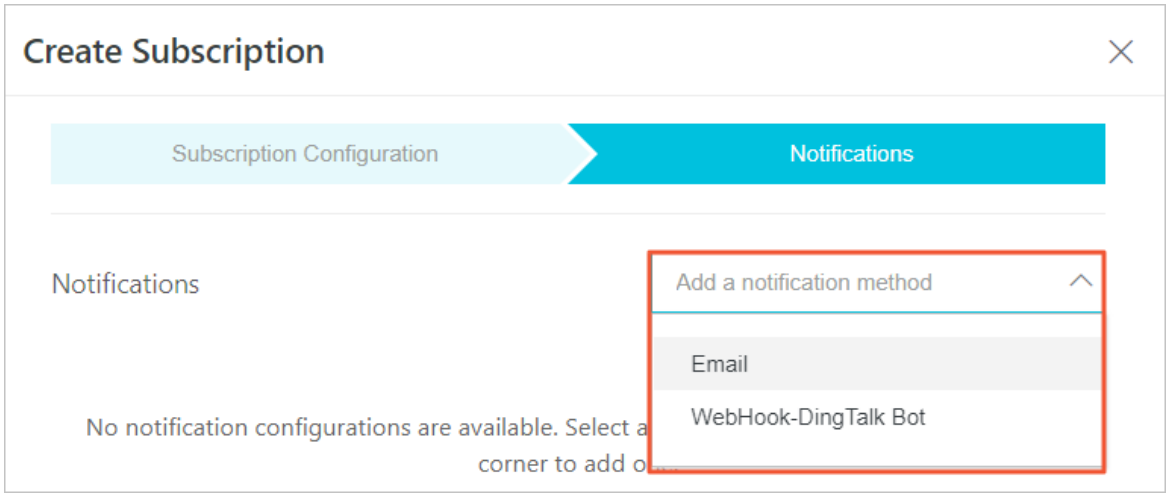
1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane of the **Instance Information** page, choose **Logs** > **Audit Log**.
5. In the upper-right corner of the **Audit Log** page, click **Subscribe**.



6. On the **Create Subscription** page, complete the settings and click **Next** at the bottom of the page.The following table describes the parameters on the Create Subscription page.

Parameter	Description
Subscription Name	The description of the subscription. You can customize the description.
Frequency	Specify the frequency at which ApsaraDB for Redis delivers the reports.
Add Watermark	After you enable this feature, the images in the reports are watermarked with the email address or the WebHook URL.

7. On the **Notifications** page, click the drop-down list on the right and select a notification method.



The available notification methods are **Email** and **WebHook-DingTalk Bot**. You can choose one or both of them.

> **Note** For more information about how to obtain the WebHook request URL, see Set up a DingTalk chatbot webhook.

8. Specify the **Recipients** of the **Email** or the **Request URL** of the **WebHook-DingTalk Bot**, and then click **Submit**.

8.2. Audit logs (previous version)

8.2.1. Enable an earlier version of the audit log feature

To view request records of ApsaraDB for Redis by using audit logs, you must enable the audit log feature.

Prerequisites

- Either of the following instances is used: ApsaraDB for Redis instances of Community Edition and performance-enhanced instances of Enhanced Edition.
- The engine version of the ApsaraDB for Redis instance is Redis 4.0 or later.
- The ApsaraDB for Redis instance uses the latest minor version.

> **Note** If your instance version does not allow you to query audit logs and must be upgraded before you can use the audit log feature, upgrade the major or minor version as needed. For more information, see Upgrade the minor version and Upgrade the major version.

Applicability

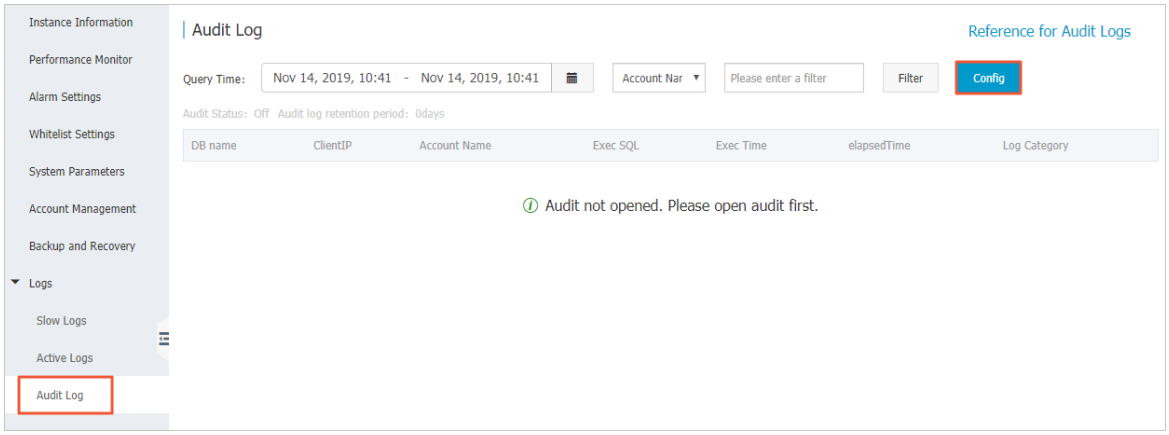
This topic is applicable to earlier versions of the audit log feature of ApsaraDB for Redis. For more information about the new version, see Enable the new version of the audit log feature.

Background

After the audit log feature is enabled, the system applies for disk space to save logs. By default, logs are retained for 7 days. You can set the logs to be retained for up to 30 days. The audit log feature is free of charge and may incur fees in later versions.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. On the **Audit Log** page, click **Config**.



6. In the dialog box that appears, click **On**, set **Retention Days**, and then click **Confirm**.

> **Note** To disable the audit log feature, click **Off**. If the audit log feature is disabled, existing audit logs are deleted. Proceed with caution.

7. In the **Open Audit Log** dialog box, read the prompt and click **Confirm, On**.

8.2.2. Query audit logs for earlier versions

You can query audit logs to view request records within a specified period, and filter the records by using required filter criteria.

Prerequisites

-
- The engine version of the ApsaraDB for Redis instance is Redis 4.0 or later.
- The ApsaraDB for Redis instance uses the latest minor version.
- The audit log feature is enabled.

> **Note** If your instance version does not allow you to query audit logs and must be upgraded before you can use the audit log feature, upgrade the major or minor version as needed. For more information, see Upgrade the minor version and Upgrade the major version.

Applicability

This topic is applicable to earlier versions of the audit log feature of ApsaraDB for Redis. For more information about the new version, see Query audit logs.

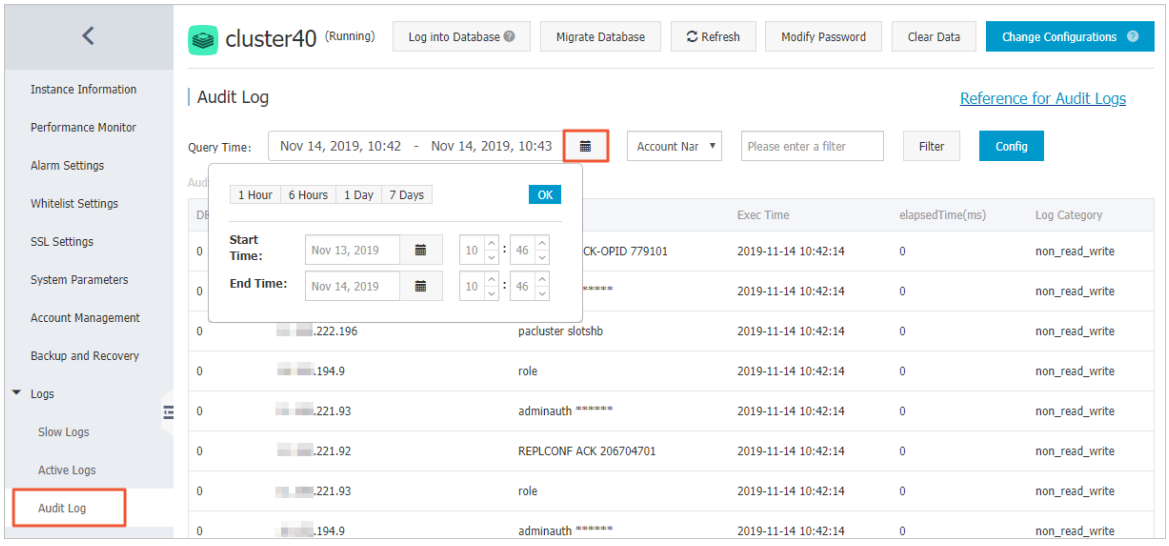
Background

Audit logs provide a detailed insight into the status of your ApsaraDB for Redis instance. You can use audit logs to view request records. This allows you to check records of modify and delete operations and find the cause of sudden increases in database resource consumption.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. On the **Audit Log** page, click the calendar icon in the **Query Time** field, select a preset period, or set **Start Time** and **End Time**, and then click **OK**.

> **Note** The end time cannot be earlier than the time when the audit log feature is enabled.



8.2.3. Filter audit logs of earlier versions

You can filter audit logs by account name, client IP, database name (database ID), command, or cluster node.

Prerequisites

-
- The engine version of the ApsaraDB for Redis instance is Redis 4.0 or later.
- The ApsaraDB for Redis instance uses the latest minor version.
- The audit log feature is enabled.
- Logs exist in the specified time range.

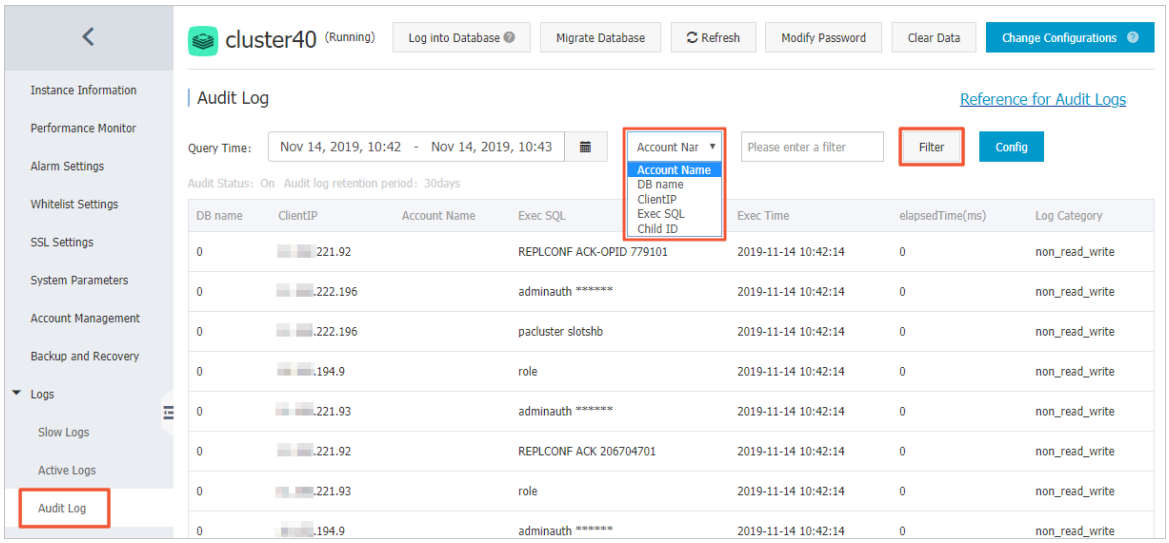
> **Note** If your instance version does not allow you to query audit logs and must be upgraded before you can use the audit log feature, upgrade the major or minor version as needed. For more information, see Upgrade the minor version and Upgrade the major version.

Applicability

This topic is applicable to earlier versions of the audit log feature of ApsaraDB for Redis. For more information about the new version, see Filter audit logs.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. Select or enter the required conditions to filter audit logs.



> **Note** For cluster instances, you can select **Child ID** to filter logs.

6. Click **Filter**.

8.2.4. Set the retention period for audit logs of earlier versions

This topic describes how to set the period for which audit logs of earlier versions are retained. You can log on to the ApsaraDB for Redis console and set the retention period. Audit logs can be stored for 7 to 30 days.

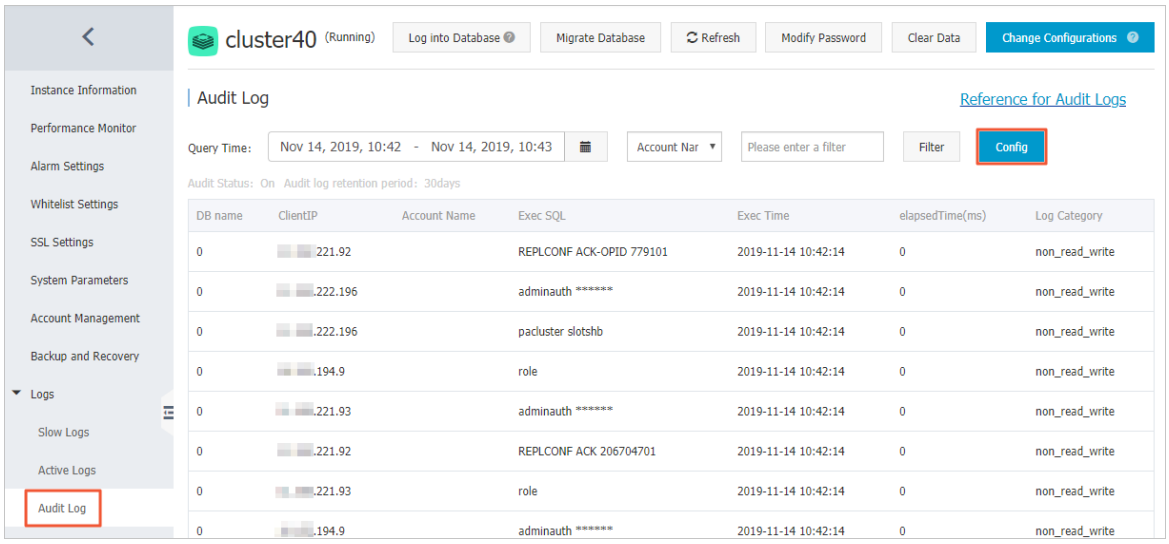
Prerequisites

- Either of the following instances is used: ApsaraDB for Redis instances of Community Edition and performance-enhanced instances of Enhanced Edition.
- The engine version of the ApsaraDB for Redis instance is Redis 4.0 or later.
- The ApsaraDB for Redis instance uses the latest minor version.
- The audit log feature is enabled.

> **Note** If your instance version does not allow you to query audit logs and must be upgraded before you can use the audit log feature, upgrade the major or minor version as needed. For more information, see Upgrade the minor version and Upgrade the major version.

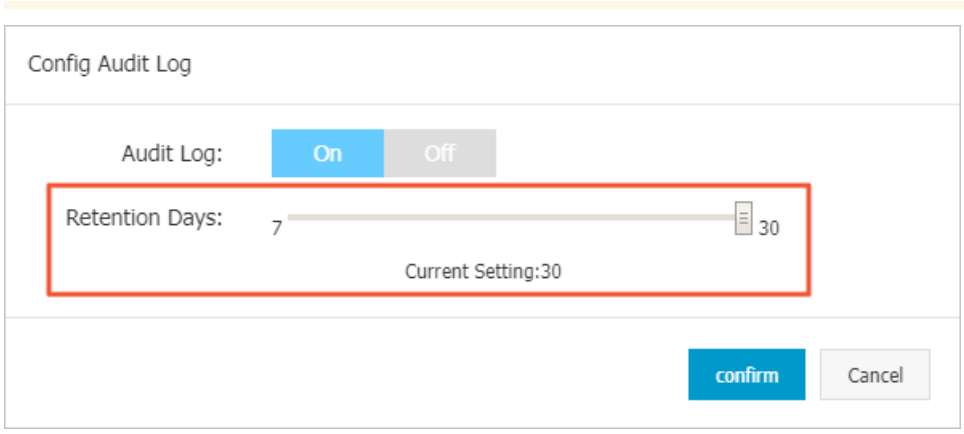
Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Audit Log**.
5. On the **Audit Log** page, click **Config**.



6. In the **Config Audit Log** dialog box, adjust the **Retention Days** slider, and click **Confirm**.

> **Warning** Expired log entries are not retained.



8.3. Query slow logs

This topic describes how to query slow logs in the ApsaraDB for Redis console. This allows you to troubleshoot performance issues and optimize requests.

Prerequisites

- The database version of the ApsaraDB for Redis instance is Redis 4.0 or later, and the latest minor version is used.

  > **Note** If the instance version does not meet the requirements, you can upgrade the instance version after assessing the compatibility with your business. For more information, see Upgrade the minor version and Upgrade the major version.

- ApsaraDB for Redis instances of Community Edition instances and performance-enhanced instances of Enhanced Edition instances are used.

Context

Slow logs are used to record requests whose execution time exceeds a specified threshold. The slow log is classified into the data node slow log and proxy slow log.

> **Note** For standard instances, only slow logs of data nodes are collected.

Slow log type	Description	Parameter

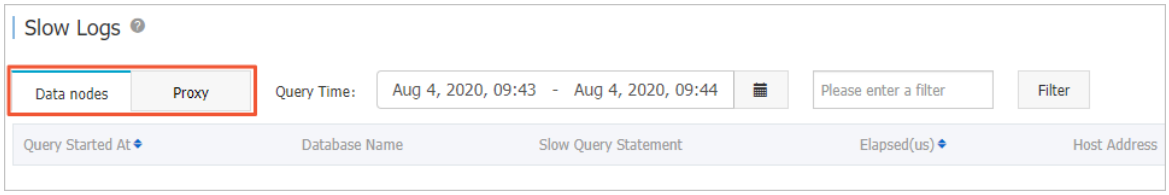
Slow log type	Description	Parameter
Slow logs from data nodes	• The command execution time collected in slow logs that are generated on a data node only includes the command execution time on the data node. The communication time between the data node and a proxy or client, and the latency of the command in the single-threading queue are not included. • The number of slow logs of data nodes is small due to the high performance of ApsaraDB for Redis.	• slowlog-log-slower-than: specifies the threshold of command execution time. If the time consumed to run a command exceeds this threshold, the command is recorded in a slow log. Default value: 10000 μs (10 ms). Note Typically, the latency seems to be higher than the specified value of this parameter, because the specified value does not include the time consumed to transmit and process data among clients, proxies, and data nodes. • slowlog-max-len: specifies the maximum number of slow logs that can be stored. Default value: 1024. For more information, see Parameter overview and configuration guide.
Slow logs from the proxy server	• The command execution time collected in proxy slow logs starts from the time when the proxy server sends a request to a data node and ends at the time when the proxy receives the response from the data node. This time includes the command execution time on the data node, the data transmission time over the network, and the queuing latency of the command. • Proxy slow logs are retained for 72 hours. However, the number of proxy slow logs is not limited. • The latency recorded in proxy slow logs is similar to the latency that you experience on your application. We recommend that you check this type of logs when you troubleshoot timeout issues.	rt_threshold_ms: specifies the threshold of slow logs from the proxy server. Default value: 500 ms. We recommend that you set the threshold to a value that is similar to the client timeout. The value can be 200 to 500 ms. For more information, see Parameter overview and configuration guide.

Procedure

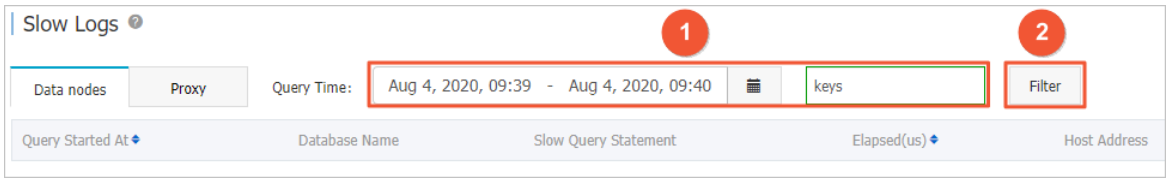
1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, choose **Logs > Slow Logs**.

5. (Optional)For cluster or read/write splitting instances, specify the node type from which you want to query slow logs.

> **Note** For standard instances, only slow logs of data nodes are collected. Skip this step.



6. Filter the results based on the specified query time range (within 72 hours) and keywords.



> **Note** When you connect to a cluster instance of Community Edition by using the default endpoint (the endpoint of the proxy server), the **Host Address** displayed on the Data nodes tab are the IP addresses of proxy nodes rather than the client IP addresses. You can use a private endpoint to connect to the instance to avoid this restriction. For more information, see Use a private endpoint to connect to an ApsaraDB for Redis instance. To use the endpoint of the proxy server to connect to an instance, you can use ApsaraDB for Redis Enhanced Edition (Tair). Tair provides client IP addresses in data node logs.

References

Use slow logs to troubleshoot timeout issues

8.4. Query active logs of an instance

This topic describes how to query active logs of an ApsaraDB for Redis instance. In the **Logs** module of the ApsaraDB for Redis console, you can check the active logs that are generated in the last 72 hours to troubleshoot operations and maintenance (O&M) issues.

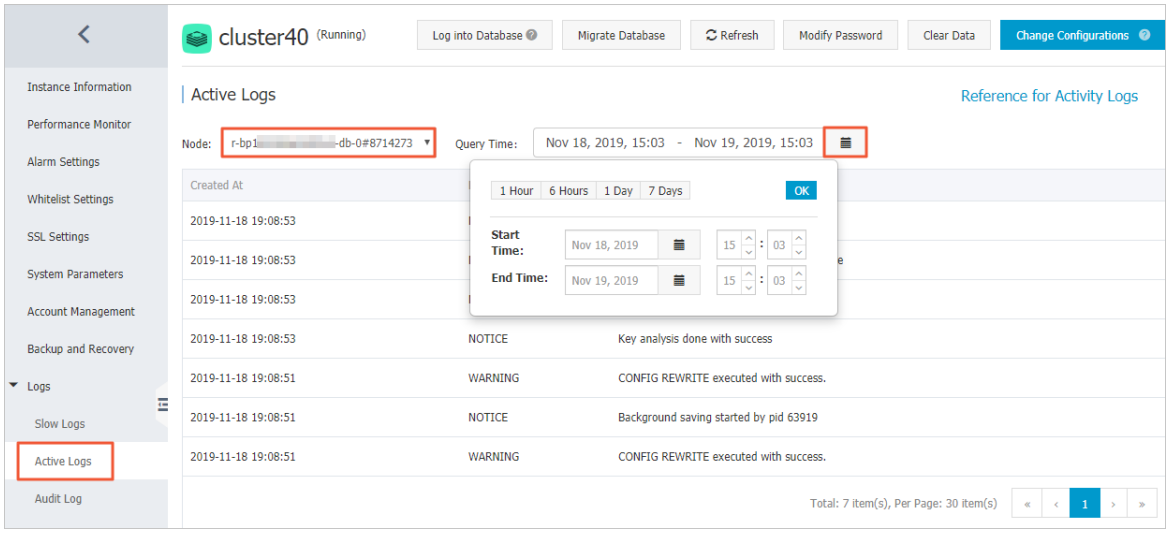
Prerequisites

-
- The engine version of the ApsaraDB for Redis instance is Redis 4.0 or later.
- The ApsaraDB for Redis instance uses the latest minor version.

> **Note** If your instance version does not allow you to query active logs and must be upgraded before you can use the active log feature, upgrade the major or minor version as needed. For more information, see Upgrade the minor version and Upgrade the major version.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. The Instance Information page appears. In the left-side navigation pane, choose **Logs** > **Active Logs**.
5. On the **Active Logs** page, click in the **Query Time** field, select a preset period, or set **Start Time** and **End Time**, and then click **OK**.



Note If you use a cluster instance of ApsaraDB for Redis, you can select the required node in the **Node** drop-down list next to **Query Time**.

8.5. Use slow logs to troubleshoot timeout issues

Connection timeouts caused by slow requests are the common issues that affect the service performance. The slow log feature of ApsaraDB for Redis allows you to find the IP address of the client that sends these requests and troubleshoot issues based on the details of slow logs.

Prerequisites

- The database version of the ApsaraDB for Redis instance is Redis 4.0 or later, and the latest minor version is used.

  Note If the instance version does not meet the requirements, you can upgrade the instance version after assessing the compatibility with your business. For more information, see Upgrade the minor version and Upgrade the major version.

- ApsaraDB for Redis instances of Community Edition instances and performance-enhanced instances of Enhanced Edition instances are used.

Context

Slow logs are used to record requests whose execution time exceeds a specified threshold. The slow log is classified into the data node slow log and proxy slow log.

> **Note** For standard instances, only slow logs of data nodes are collected.

Slow log type	Description	Parameter
Slow logs from data nodes	• The command execution time collected in slow logs that are generated on a data node only includes the command execution time on the data node. The communication time between the data node and a proxy or client, and the latency of the command in the single-threading queue are not included. • The number of slow logs of data nodes is small due to the high performance of ApsaraDB for Redis.	• slowlog-log-slower-than: specifies the threshold of command execution time. If the time consumed to run a command exceeds this threshold, the command is recorded in a slow log. Default value: 10000 μs (10 ms). **Note** Typically, the latency seems to be higher than the specified value of this parameter, because the specified value does not include the time consumed to transmit and process data among clients, proxies, and data nodes. • slowlog-max-len: specifies the maximum number of slow logs that can be stored. Default value: 1024. For more information, see Parameter overview and configuration guide.
Slow logs from the proxy server	• The command execution time collected in proxy slow logs starts from the time when the proxy server sends a request to a data node and ends at the time when the proxy receives the response from the data node. This time includes the command execution time on the data node, the data transmission time over the network, and the queuing latency of the command. • Proxy slow logs are retained for 72 hours. However, the number of proxy slow logs is not limited. • The latency recorded in proxy slow logs is similar to the latency that you experience on your application. We recommend that you check this type of logs when you troubleshoot timeout issues.	rt_threshold_ms: specifies the threshold of slow logs from the proxy server. Default value: 500 ms. We recommend that you set the threshold to a value that is similar to the client timeout. The value can be 200 to 500 ms. For more information, see Parameter overview and configuration guide.

Methods used to query slow logs

Slow log type	Method
Slow logs from data nodes	Connect to the ApsaraDB for Redis instance from a client and run the **SLOWLOG GET** command. For more information, see Redis Commands.
Slow logs from the proxy server	Log on to the ApsaraDB for Redis console or call an API operation: • Query slow logs • DescribeSlowLogRecords

Procedure

In most cases, service timeouts may be caused by slow requests. We recommend that you perform the following steps to troubleshoot the timeout issues.

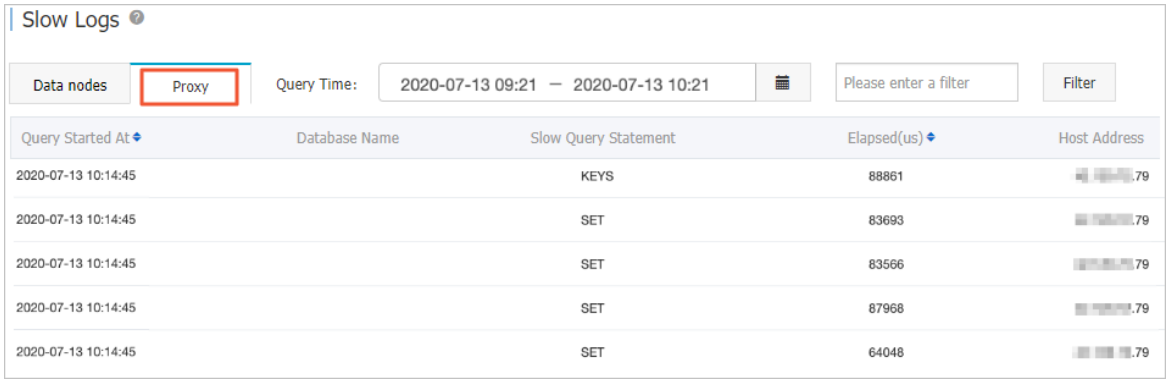
1. If a service timeout issue occurs, check the slow logs generated on the proxy server first. For more information, see Query slow logs.

   > **Note**
   > - For standard instances, go to Step 3 and analyze slow logs of data nodes.
   > - If no log exists, you can check the network between the client and the ApsaraDB for Redis instance.

2. Find the command that is recorded by the earliest proxy slow log.

   > **Note** If slow requests occur on data nodes and cause command accumulation, these requests are recorded in proxy slow logs.

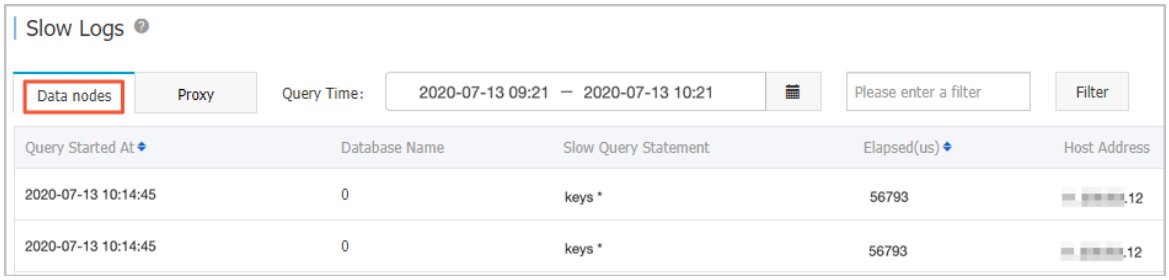
   In this example, the earliest recorded slow log is caused by the **KEYS** command. The IP address on the right of the log entry is the IP address of the client that sends the command.



3. Check the data node slow logs to find the proxy slow logs that cause the timeout issue.

> **Note**
>
> - When you connect to a cluster instance of Community Edition by using the default endpoint (the endpoint of the proxy server), the **Host Address** displayed on the Data nodes tab are the IP addresses of proxy nodes rather than the client IP addresses. To avoid this restriction, you can use a private endpoint to connect to the instance. For more information, see Use a private endpoint to connect to an ApsaraDB for Redis instance. To use the endpoint of the proxy server to connect to an instance, you can use ApsaraDB for Redis Enhanced Edition (Tair). Tair provides client IP addresses in data node logs.
> - Typically, the command that generates slow logs first in the proxy slow log can also generate data node slow logs. The number of slow logs for a data node is less than that of the proxy server. This is due to the different definitions of the execution time and different thresholds of slow logs.

In this example, after you view proxy slow logs, you can see that the slow log caused by the **KEYS** command also exists in data node slow logs. No other slow logs that are displayed on the Proxy tab exist on the Data nodes tab. This shows that the **KEYS** command causes the timeout.



4. In proxy slow logs, you can search for the client IP address based on the command that is found in Step 2.

9.Backup and recovery

9.1. Back up and restore data in the console

More and more users choose to use Redis for persistent storage. Consequently, users raise higher requirements for data reliability. The data backup and restore solution provided by ApsaraDB for Redis has significantly improves data reliability.

Automatic backup by using backup policies

Background

An increasing number of applications use Redis for persistent storage. In this case, an automatic backup mechanism is required to back up data on a regular basis so that you can quickly restore data in the event of user errors. ApsaraDB for Redis uses RDB snapshots to back up data on replica nodes. The backup process has no negative impacts on the performance of your instance. You can customize a backup policy in the console.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the instance ID or **Manage** in the **Actions** column for the instance that you want to manage.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. Click the **Backup Settings** tab.
6. Click **Edit** to customize the automatic backup cycle and backup time.

   > **Notice** By default, backup data is retained for seven days. You cannot modify this configuration.

7. Click **OK**.

Manual backup (instant backup)

In addition to automatic backup, you can also manually back up data at any time in the console.

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the instance ID or **Manage** in the **Actions** column for the instance that you want to manage.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. Click **Create Backup** in the upper-right corner.
6. In the dialog box that appears, click OK to back up data for the instance.

> **Notice** On the **Data Backup** tab, you can select a time range to query historical backup data. By default, backup data is retained for seven days. You can query the historical backup data within the last seven days.

Backup archiving

Background

Due to industry regulations or corporate policy requirements, you may need to archive backup data for Redis periodically. ApsaraDB for Redis provides a backup archiving feature, which is free of charge. You can use this feature to store automatically and manually created backup files in Object Storage Service (OSS). You can store backup files in Alibaba Cloud OSS free of charge for up to seven days. After seven days, the backup files are automatically deleted.

To retain backup files for a longer time period, you can copy their URLs of the backup files from the console and download the backup files to a local host.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the instance ID or **Manage** in the **Actions** column for the instance that you want to manage.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. On the Data Backup tab, select the backup dataset to be archived and click **Download Backups** in the Actions column.

Data restore

ApsaraDB for Redis provides the data restore feature to minimize the possible loss caused by user errors. You can restore data from backup files in the console or use redis-shake on an Elastic Compute Service (ECS) instance to restore data from local backup files.

Restore data in the console

> **Notice** You cannot restore data for cluster instances in the console. To restore data for cluster instances, use redis-shake.

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the instance ID or **Manage** in the **Actions** column for the instance that you want to manage.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. On the **Backup and Recovery** page, click the **Data Backup** tab.
6. Select the time range of the data to be restored and click **Search**. Select the backup dataset that you want to restore and click **Restore Data**.
7. In the Data Recovery dialog box, click **OK** to restore data to the instance. You can click **Clone Instance** to create an instance and restore the backup data to the created instance. Make sure that only the expected data is to be restored before you confirm the operation.

> **Notice** Proceed with caution when you restore data. We recommend that you clone an instance to restore data if time permits. This solution allows you to create a pay-as-you-go instance and restore data from a backup dataset to the created instance. Make sure that only the expected data is to be restored before you confirm the operation.

Restore data on an ECS instance

You can use redis-shake on an ECS instance to restore data from local backup files to an ApsaraDB for Redis instance. For more information, see Use the redis-shake tool to back up data.

Clone an instance

O&M administrators may need to deploy an application during their daily maintenance work. It is simple for O&M administrators to deploy an application in an ECS instance that is created based on an image file. Deploying a database is complex. O&M administrators must purchase or install a database, and then initialize relevant database scripts, for example, to create tables, triggers, and views. In this scenario, they must perform many trivial operations, which are more likely to involve user errors. In the gaming industry that requires efficient service deployment, administrators have to quickly deploy an application multiple times each day.

To resolve this issue, ApsaraDB for Redis develops an instance cloning feature. This feature enables you to create a subscription or pay-as-you-go instance based on backup files. You can develop and deploy a database on the web interface, which improves productivity. For more information, see Clone an instance.

Related API operations

- CreateBackup
- ModifyBackupPolicy
- DescribeBackupPolicy
- DescribeBackups
- RestoreInstance

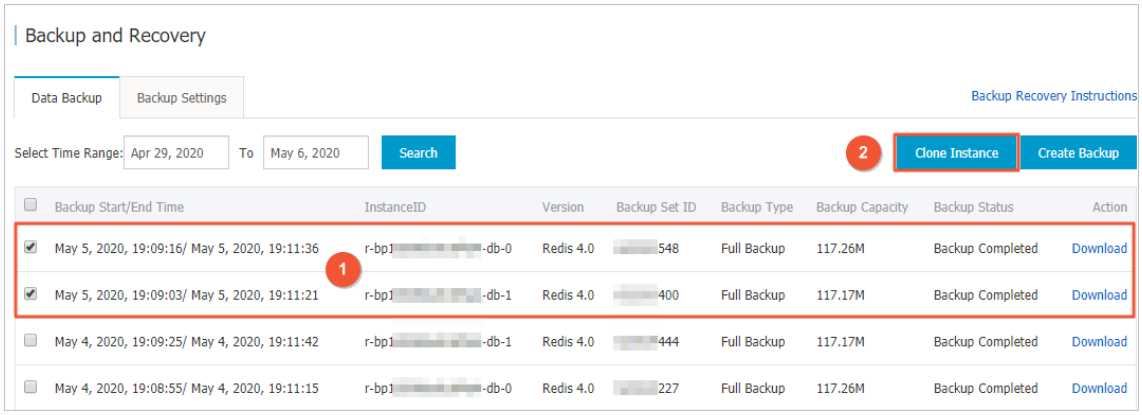
9.2. Clone an instance

You can clone a new instance from a backup file in the ApsaraDB for Redis console to quickly deploy applications.

Procedure

1. Log on to the ApsaraDB for Redis console.
2. On the top of the page, select the region where the instance is deployed.
3. On the **Instances** page, click the Instance ID of the instance.
4. In the left-side navigation pane, click **Backup and Recovery**.
5. On the **Backup and Recovery** page, click the **Data Backup** tab, select a data source, and complete the following operations based on the instance type:
   - If the ApsaraDB for Redis instance is a standard instance or a read/write splitting instance, click **Clone Instance** in the **Action** column.
   - If the ApsaraDB for Redis instance is a cluster instance, select a data source, and click **Clone Instance**.

> **Note** You must select a backup data record for each data shard. For example, a cluster contains two data shards: db-0 and db-1. Before you clone the instance, select a backup data record for each data shard, as shown in the following figure. If you select multiple backup data records for a shard, the system uses the latest record for cloning.



6. In the **Clone Instance** dialog box that appears, click **OK**.
7. On the **Clone Instance** page, configure the following parameters.

Parameter	Description
Region	The region of the instance. The value is the same as that of the source instance. You cannot modify this parameter.
Clone Source Type	Select **Clone Instance**.

Parameter	Description
Network Type	○ Classic Network: a traditional type of network. ○ VPC: the recommended network type. A virtual private cloud (VPC) is an isolated virtual network that provides higher security and better performance than a classic network. **Notice** ○ The ApsaraDB for Redis instance must be of the same network type as the ECS instance to be connected. Otherwise, these instances cannot communicate over an internal network. ○ If both instances use the VPC network type, to enable connections over an internal network, the instances must be connected to the same VPC. ○ You can change the network type of an ApsaraDB for Redis instance from classic network to VPC. However, you cannot change the network type from VPC to classic network. For more information, see Switch to VPC network.
VSwitch	A VSwitch is a basic network module of a VPC. If no VSwitch is available in the VPC, create a VSwitch first. For more information, see Create a VSwitch.
Edition Type	○ Community Edition: This edition is compatible with the open source Redis protocol and provides database solutions that support hybrid storage in memory and disks. ○ (Tair): This edition is developed based on ApsaraDB for Redis Community Edition and is optimized in terms of performance, storage, and data schemas. For more information, see Overview.

Parameter	Description
Series Type	provides the following series types: ○ Performance-enhanced. For more information, see Enhanced multi-threading performance and Integration of multiple Redis modules. ○ Hybrid Storage. For more information, see Hybrid-storage instances. Note If you select Community Edition, this parameter is not supported.
Version	ApsaraDB for Redis supports the following engine versions: ○ 2.8 ○ 4.0 ○ 5.0 By default, the version is the same as that of the source instance. You cannot modify this parameter.
Architecture Type	○ Cluster ○ Standard For more information, see Overview.
Shards	The number of shards on a cluster instance. Note The number of shards can be specified for cluster instances only.
Node Type	Standard instances and cluster instances only support the master-replica type. This type provides a master node and a replica node that are deployed in the hot standby mode to support data persistence.
Instance Class	Each instance type contains a group of configurations, such as the memory, maximum number of concurrent connections, and maximum bandwidth. For more information, see Overview. Note After you create an instance, database metadata is automatically generated and consumes a small amount of storage space. ○ A standard instance has about 50 MB of metadata. ○ On each shard of an ApsaraDB for Redis cluster instance, the size of metadata is about 50 MB. In a cluster, the total amount of storage space that metadata uses is the total size of metadata on each shard.

Parameter	Description
Password Setting	Specify a time to set the password. ○ Now: Set the instance password now. ○ Later: Set the password after the instance is created. For more information, see Change the password.
Instance Name	The name of the ApsaraDB for Redis instance. If this parameter is not specified, the instance ID is used as the instance name.
Duration	Set the subscription duration. This option is not displayed if the billing method is pay-as-you-go.

8. Click ApsaraDB for KVStore Agreement of Service, and read the agreement.
9. If you accept the service agreement, click the check box before the service agreement.
10. Click Activate.

9.3. Use the redis-shake tool to back up data

You can use the dump mode of the redis-shake tool to back up the data of an ApsaraDB for Redis instance to an RDB file.

Prerequisites

- A database account that has the Replicate permission is created for an ApsaraDB for Redis instance. For more information about how to create a database account, see Manage database accounts.
- The architecture of the ApsaraDB for Redis instance is the standard edition or single-node read/write splitting edition.
- The version of the ApsaraDB for Redis instance is Redis 4.0.
- An Elastic Compute Service (ECS) instance is created for running the redis-shake tool.
  - The IP address of the ECS instance is added to the whitelist of the source ApsaraDB for Redis instance.
  - The ECS instance is running the Linux operating system.
  - The remaining disk space in the ECS instance is larger than the space required by the RDB file to be generated.

Background

The redis-shake tool is an open-source tool developed by Alibaba Cloud. You can use it to parse (decode mode), recover (restore mode), back up (dump mode), and synchronize (sync/rump mode) Redis data. In dump mode, the redis-shake tool can back up the data of a Redis database to an RDB file, which can be used to recover or migrate data. This topic describes how to use the dump mode of the redis-shake tool to back up the data of an ApsaraDB for Redis instance to an RDB file.

> **Note**
>
> - The redis-shake tool can use an RDB file to recover or migrate data. For more information, see Use the redis-shake tool to migrate data from an RDB file.
> - For more information about the redis-shake tool, see redis-shake on GitHub or FAQ.

Procedure

1. Log on to the ECS instance that can access the source ApsaraDB for Redis instance.
2. Download the redis-shake tool in the ECS instance.

   > **Note** We recommend that you download the latest version.

3. Run the following command to decompress the downloaded `redis-shake.tar.gz` package:

   ```
   tar -xvf redis-shake.tar.gz
   ```

   > **Note** In the decompressed folder, the redis-shake file is a binary file that can be run in the 64-bit Linux operating system. The redis-shake.conf file is the configuration file of the redis-shake tool. You need to modify this configuration file in the next step.

4. Modify the redis-shake.conf file. The following table describes the parameters for the dump mode of the redis-shake tool.

Parameters for the dump mode of the redis-shake tool

Parameter	Description	Example
source.address	The connection address and service port of the source ApsaraDB for Redis instance.	xxxxxxxxxxxx.redis.rds.aliyuncs.com:6379
source.password_raw	The password of the source ApsaraDB for Redis instance.	account:password
rdb.output	The name of the RDB file to be generated.	local_dump

5. Run the following command to back up data:

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

   > **Note** You must run this command in the same directory as the redis-shake and redis-shake.conf files. Otherwise, you need to specify the correct file path in the command.

```
[root@                              ]# ./redis-shake.linux64 -type=dump -conf=redis-s
hake.conf
2019/05/23 15:38:07 [WARN]
_______________________________
\                               \            _          ______   |
 \                               \        /   \___-=O'/|O'/__|
  \  redis-shake, here we go !! \_______\          / | /    )
  /                               /         '/-==__ _/__|/__=-|  -GM
 /                               /          *             \ | |
/                               /                         (o)
------------------------------
if you have any problem, please visit https://github.com/alibaba/RedisShake/wiki
/FAQ

2019/05/23 15:38:07 [INFO] redis-shake configuration: {"Id":"redis-shake","LogFi
le":"","LogLevel":"info","SystemProfile":9310,"HttpProfile":9320,"NCpu":0,"Paral
lel":32,"SourceType":"standalone","SourceAddress":"                    .redis.rd
s.aliyuncs.com:6379","SourcePasswordRaw":"a1:        ","SourcePasswordEncoding":
"","SourceVersion":0,"SourceAuthType":"auth","SourceParallel":1,"TargetAddress":
"127.0.0.1:20551","TargetPasswordRaw":"","TargetPasswordEncoding":"","TargetVers
ion":0,"TargetDB":-1,"TargetAuthType":"auth","TargetType":"standalone","RdbInput
":["local"],"RdbOutput":"local_dump","RdbParallel":1,"FakeTime":"","Rewrite":tru
e,"FilterDB":"","FilterKey":[],"FilterSlot":[],"BigKeyThreshold":524288000,"Psyn
c":false,"Metric":true,"MetricPrintLog":false,"HeartbeatUrl":"","HeartbeatInterv
al":3,"HeartbeatExternal":"test external","HeartbeatNetworkInterface":"","Sender
Size":104857600,"SenderCount":5000,"SenderDelayChannelSize":65535,"KeepAlive":0,
"PidPath":"","ScanKeyNumber":50,"ScanSpecialCloud":"","ScanKeyFile":"","ReplaceH
ashTag":false,"ExtraInfo":false,"SockFileName":"","SockFileSize":0,"SourceAddres
sList":["r-bp1                    .redis.rds.aliyuncs.com:6379"],"TargetAddressList":
null,"HeartbeatIp":"127.0.0.1","ShiftTime":0,"TargetRedisVersion":"","TargetRepl
ace":false}
2019/05/23 15:38:07 [INFO] start routine[0]
2019/05/23 15:38:07 [INFO] routine[0] dump from 'r-bp1                    .redis.rds.
aliyuncs.com:6379' to 'local_dump.0'
2019/05/23 15:38:07 [INFO] try to auth address[r-bp1                    .redis.rds.al
iyuncs.com:6379] with type[auth]
2019/05/23 15:38:07 [INFO] auth OK!
2019/05/23 15:38:07 [INFO] routine[0] source db[r-bp1                    .redis.rds.a
liyuncs.com:6379] dump rdb file-size[262]
2019/05/23 15:38:07 [INFO] routine[0] total = 262 -          262 [100%]
2019/05/23 15:38:07 [INFO] routine[%v] dump: rdb done0
2019/05/23 15:38:07 [INFO] execute runner[*run.CmdDump] finished!
2019/05/23 15:38:07 [WARN]
                #####  |  #####
Oh we finish ? #  _  _  #|#  _  _  #
               #        |        #
         |       ############
                     # #
  |                  # #
                    #   #
         |     |    #   #      |        |
  |  |              #     #               |
         | |   |    # .-. #          |
                    #( O )#     |     |      |
  |  ################.  .################  |
   ##   _ _|____|      ###      |_ __| _   ##
  #  |                                 |  #
  #  |     |     |     |    |     |    |  #
   ####################################
                    #     #
                     #####
```

> Note
>
> - When `execute runner[*run.CmdDump] finished!` appears in redis-shake logs, the data is backed up to the RDB file.
> - The name of the RDB file is `local_dump.0` by default. You can run the `cat local_dump.0` command to check whether Redis data is backed up.

(Optional) Next step

Use the RDB file to recover data to the destination ApsaraDB for Redis instance. For more information, see Use the redis-shake tool to migrate data from an RDB file.

9.4. Use the redis-shake tool to recover data

You can use the redis-shake tool to recover data from an RDB file to ApsaraDB for Redis. For more information, see Use the redis-shake tool to migrate data from an RDB file.

> Note For more information about how to use the redis-shake tool to back up data to an RDB file, see Use the redis-shake tool to back up data.

10.FAQ

10.1. Data persistence

ApsaraDB for Redis supports the following data persistence methods: RDB persistence and append-only file (AOF) persistence. This topic describes the features of these persistence methods and provides guidelines about how to manage data persistence in the console. You can modify persistence settings to serve your workloads.

RDB persistence

Based on RDB persistence, ApsaraDB for Redis periodically creates snapshots for the data stored in the engine, generates RDB files, and then saves the files to disks. RDB files consume minimal space to simplify data migration. You can use RDB files to back up or migrate ApsaraDB for Redis data at a specified point in time.

By default, ApsaraDB for Redis generates RDB snapshots on a daily basis and retains the snapshots for up to seven days. You can manage RDB persistence in the following ways:

- Configure the point in time for automatic backup. For more information, see Automatic backup.
- Create backups with a few mouse clicks in the console. For more information, see Manual backup.
- Restore data based on RDB files. For more information, see Data recovery.
- Download RDB files. For more information, see Backup archiving.
- Clone an instance based on RDB files. For more information, see Back up and restore data in the console.

AOF persistence

Based on AOF persistence, ApsaraDB for Redis logs all commands executed to write data, such as **SET**. When you restart ApsaraDB for Redis, the service reruns the commands in the AOF files to restore data. If AOF files are too large, Redis runs the AOF Rewrite task to recreate AOF files and optimize their storage usage.

You can specify the AOF_FSYNC_EVERYSEC policy to enable AOF persistence in ApsaraDB for Redis. Based on this policy, the system logs the commands in write requests by using AOF files on a per-second basis and saves the AOF files to disks. AOF persistence does not decrease the service performance and can minimize data loss caused by user mistakes. ApsaraDB for Redis allows you to archive incremental backups based on AOF files and guarantees the service performance when the system runs the AOF Rewrite task. You can manage AOF persistence in the following ways:

- Enable or disable AOF persistence.
- To restore data on an ApsaraDB for Redis instance within several seconds, submit a ticket and request technical support.

Data persistence tools

You can run the SYNC command of redis-shake to synchronize Redis data in real time and create backups on a destination instance. For more information, see Use redis-shake to migrate data.

10.2. What is the size of each database on an ApsaraDB for Redis instance and how can I choose databases?

By default, each ApsaraDB for Redis instance has 256 databases named from DB 0 to DB 255.

The size of each database is not restricted. But the available database space is limited by the overall space of the ApsaraDB for Redis instance.

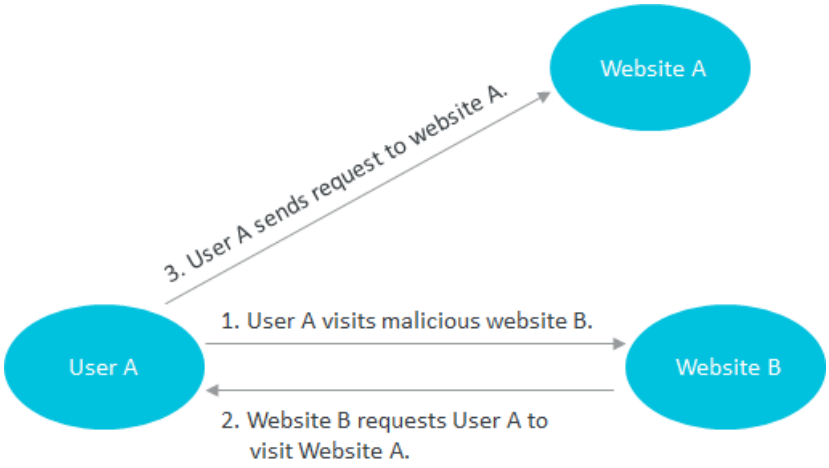
To switch between different databases, run the **SELECT** command. For example, to select DB 10, run the following command.

```
SELECT 10
```

10.3. Analysis of the Redis CSRF vulnerability and the corresponding security measure in ApsaraDB for Redis

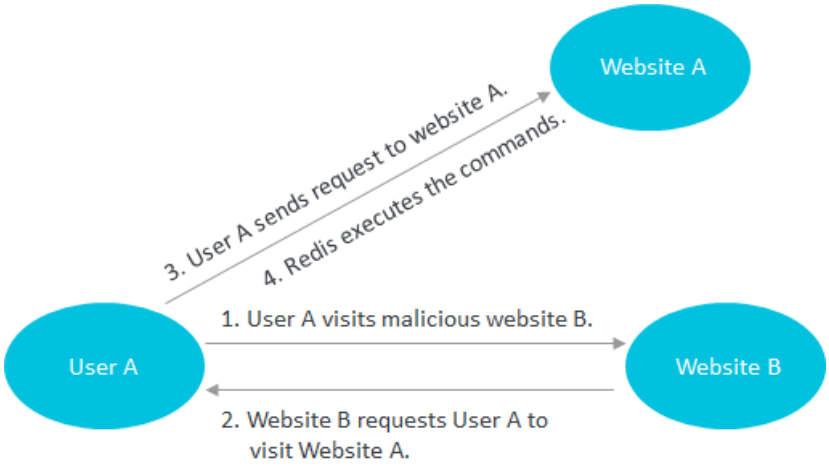
What is CSRF?

Cross-site request forgery (CSRF), also known as XSRF, one-click attack, or session riding, is a type of malicious exploitation of websites.



The preceding figure shows a simple model of a CSRF attack. A user visits the malicious website B, and the malicious website B replies to the user with an HTTP message that requires the user to visit the website A. If the user has maintained a trust relationship with the website A, the system processes the request as if the user personally sent the request to visit the website A.

Redis CSRF attack model



Based on the preceding principle of CSRF, a malicious website can require a user to send an HTTP request to Redis. Redis supports text protocols, and does not break off the connection in the case of illegal protocols during protocol parsing. An attacker can attach a Redis command to a normal HTTP request to run the command in Redis. If the user and Redis do not require password verification, Redis runs the command normally. Consequently, the attacker can encrypt data to extort money, similar to the earlier MongoDB ransom attacks.

Repair the kernel

Redis 3.2.7 provides a fix for this issue. The system processes the POST method and HOST keywords in a special way, keeps a log of events, and disconnects from the service to prevent Redis from running subsequent legitimate requests.

Redis security risks

Earlier Redis versions have exposed a security vulnerability where an attacker can obtain the root permissions of the Redis service in a certain condition. Similar security vulnerabilities occur because some users know less about security mechanisms of Redis and have little experience of operations and maintenance for Redis. In addition, Redis lacks sufficient security protection mechanisms. However, the ApsaraDB for Redis service can provide more security mechanisms. We recommend that you use ApsaraDB for Redis as the Redis service in the cloud.

Security rules of ApsaraDB for Redis

Connections over an internal network instead of a public network

By default, ApsaraDB for Redis only supports trusted connections over the Alibaba Cloud intranet. Without applying for a public endpoint, your instance is not open to the Internet.

Physical network isolation

ApsaraDB for Redis provides a physical isolation between the physical server network and the virtual server network. Your virtual servers cannot directly connect to the backend physical server network.

VPC network isolation

If you use a virtual private cloud (VPC) of Alibaba Cloud, only the services in the same VPC can interconnect with each other.

Whitelist

ApsaraDB for Redis supports whitelists. You can set a whitelist of IP addresses in the console to allow connections based on these IP addresses.

Password verification

ApsaraDB for Redis enforces password verification for instances in a classic network. You can set a complex password to prevent password cracking.

Access permission isolation

ApsaraDB for Redis isolates permissions and accessible directories for each backend instance. The instances can only access resources by using their own path to avoid mutual interference.

Dangerous commands forbidden

ApsaraDB for Redis forbids some dangerous system management commands such as CONFIG and SAVE. If you want to modify parameters, you must pass the secondary authentication in the console. This can also avoid direct operations of the backend configuration files and management commands.

Security monitoring

ApsaraDB for Redis provides comprehensive security monitoring for physical servers. The system performs regular scans and updates security monitoring policies to locate security risks in advance.

Redis cluster password

The original Redis 3.0 cluster does not support password verification. ApsaraDB for Redis clusters support password verification to improve system security.

10.4. How can I use the Redis command line interface (redis-cli) to import on-premises Redis data to ApsaraDB for Redis?

For more information, see Use the AOF to migrate on-premises Redis data to ApsaraDB Redis .

10.5. Which version of Redis is compatible with ApsaraDB for Redis?

This topic describes the Redis engine versions that are supported by ApsaraDB for Redis.

ApsaraDB for Redis supports multiple Redis engine versions, including 2.8, 4.0, and 5.0. ApsaraDB for Redis instances of all versions support the following native Redis databases:

- ApsaraDB for Redis instances that use Redis engine 2.8 are compatible with native Redis versions 2.8 to 3.2.
- ApsaraDB for Redis instances that use Redis engine 4.0 are compatible with native Redis versions 4.0 and earlier (for example, 2.8).
- ApsaraDB for Redis instances that use Redis engine 5.0 are compatible with native Redis versions 5.0 and earlier, such as 4.0 and 2.8.

For more information, see Overview.

10.6. What is the relationship between ApsaraDB for Redis and Redis?

Alibaba Cloud ApsaraDB for Redis is a key-value cloud storage service that is compatible with Redis protocols. ApsaraDB for Redis supports most Redis commands. All clients compatible with Redis can connect to the ApsaraDB for Redis service to complete data storage and related operations.

10.7. What Redis commands and operations are compatible with ApsaraDB for Redis?

ApsaraDB for Redis is compatible with most open-source Redis commands and operations, and disables only a few commands. An ApsaraDB for Redis cluster instance does not support some Redis commands and operations. For more information, see Supported Redis commands.

10.8. Does ApsaraDB for Redis support distributed cluster instances?

ApsaraDB for Redis supports distributed cluster instances. Cluster instances provide a larger storage capacity and higher processing performance.

For more information about Redis commands that ApsaraDB for Redis cluster instances support, see Supported Redis commands.

10.9. Does a master node of an ApsaraDB for Redis instance work with multiple replica nodes?

One master node of each ApsaraDB for Redis instance can work with only one replica node instead of multiple replica nodes.

10.10. How can I use the redis-cli tool to connect to ApsaraDB for Redis?

To connect to an ApsaraDB for Redis instance with redis-cli, see Use redis-cli to connect to ApsaraDB for Redis.

10.11. Does ApsaraDB for Redis support data persistence?

This topic describes data persistence of ApsaraDB for Redis.

ApsaraDB for Redis provides a hybrid storage of memory and hard disks, and uses Redis Database (RDB) and Append Only File (AOF) to meet data persistence requirements. For more information about how to back up and restore data, see Back up and recover data in the console.

Enterprise Edition instances store full data in disks and hot data in memory. This allows you to keep a balance between performance and costs, and breaks the performance bottleneck of Redis instances caused by memory constraints. For more information, see Hybrid-storage instances.

10.12. Can I upgrade or downgrade a subscription instance?

You can upgrade or downgrade an ApsaraDB for Redis instance that has enabled the subscription billing method.

> **Notice** During the scaling process, the instance may be disconnected for several seconds. We recommend that you upgrade or downgrade the instance during off-peak hours.

For more information, see Change specifications.

10.13. Does an ApsaraDB for Redis instance have restrictions on the number of connections, CPU processing capability, and data transmission bandwidth?

All ApsaraDB for Redis instances have restrictions on the number of connections, CPU processing capability, and data transmission bandwidth. The performance parameters vary according to different types of instances. For more information, see Types and performance or the description on the page for creating an instance in the console.

10.14. Why do I receive an SMS message or email indicating that a failover is triggered?

If an exception is detected on a master node of an ApsaraDB for Redis instance, the high availability (HA) module triggers a failover. The replica node corresponding to the master node takes over services, and the original master node becomes a replica node. Then, the HA module reconstructs the new replica node. If an exception is detected on a replica node of an ApsaraDB for Redis instance, the HA module reconstructs the replica node.

Content of the SMS message or email

A failover is triggered for your ApsaraDB for Redis instance r-bp1xxxxxxxxxxxx (name: xxxxxx). Check whether your applications still connect to the ApsaraDB for Redis instance. We recommend that you enable automatic reconnection in your applications so that they can reconnect to the ApsaraDB for Redis instance after a failover.

Failover modes and impacts

Trigger	Failover mode	Impact on business
A master node fails.	A failover is triggered to switch services to the corresponding replica node. The original master node becomes a replica node and is reconstructed.	During the failover, the ApsaraDB for Redis instance may be disconnected within seconds. Reconstructing the replica node does not affect your business. Note Make sure that your applications support automatic reconnection so that they can reconnect to the ApsaraDB for Redis instance after a failover.
A replica node fails.	No failover is triggered. The replica node is reconstructed.	None.

10.15. Does ApsaraDB for Redis support master-replica replication?

Yes. ApsaraDB for Redis automatically manages the synchronization and failover operations between the master and replica nodes.

10.16. How does ApsaraDB for Redis evict data by default?

By default, an ApsaraDB for Redis instance evicts data by using the volatile-lru policy. To modify the eviction policy, log on to the ApsaraDB for Redis console, click the target instance ID on the Instance List page to go to the Instance Information page, and click System Parameters in the left-side navigation pane.

- **volatile-lru**

  The system only evicts data that has Time To Live (TTL) configured according to the Least Recently Used (LRU) algorithm.

- **volatile-ttl**

  The system only evicts data that has TTL configured, and evicts the data in ascending order of TTL.

- **allkeys-lru**

  The system evicts data according to the LRU algorithm.

- **volatile-random**

  The system only randomly evicts data that has TTL configured.

- **allkeys-random**

  The system randomly evicts data.

- **noeviction**

  The system does not evict any data, but returns an error when you write new data to the system.

- **volatile-lfu**

  The system only evicts least frequently used keys that have TTL configured according to the Least Frequently Used (LFU) algorithm.

- **allkeys-lfu**

  The system evicts least frequently used keys according to the LFU algorithm.

10.17. How many databases does each ApsaraDB for Redis instance support?

Each instance supports 256 databases.

10.18. Can I restore the deleted data of ApsaraDB for Redis?

After you delete data of ApsaraDB for Redis, Alibaba Cloud does not retain the deleted data. So you cannot restore the data.

10.19. How can I monitor ApsaraDB for Redis? Does the system automatically generate alerts when the capacity is fully used?

ApsaraDB for Redis does not provide any alerts related to the capacity. You can configure the CloudMonitor service to enable this feature. CloudMonitor provides a range of metrics for ApsaraDB for Redis. You can specify the metrics as needed.

For more information about how to configure CloudMonitor for ApsaraDB for Redis, see ApsaraDB for Redis.

10.20. Do I need a password to connect to an ApsaraDB for Redis instance? How can I obtain the password?

You must pass the password verification when connecting to an ApsaraDB for Redis instance from a client. You can set a password when creating an ApsaraDB for Redis instance. To reset the password, log on to the ApsaraDB for Redis console, and choose **Modify Password** > **Forgot password**.

10.21. Can I modify configuration parameters for ApsaraDB for Redis?

You can modify configuration parameters for ApsaraDB for Redis in the console. To view the list of parameters, log on to the ApsaraDB for Redis console.

10.22. Does each ApsaraDB for Redis instance such as a cluster instance have one master node and one replica node running simultaneously?

All ApsaraDB for Redis instances including cluster instances have one master node and one replica node running at the backend. Each shard server in a cluster instance also has one master and one replica.

10.23. Do I need to install Redis on an ECS instance to use ApsaraDB for Redis?

No. Redis clients can connect to ApsaraDB for Redis from an ECS instance.

10.24. Why is the storage usage on my newly created ApsaraDB for Redis instance greater than zero?

ApsaraDB for Redis is consistent with Redis in terms of product behavior. A newly created instance generates database metadata that occupies a fraction of the storage space on this instance. The ApsaraDB for Redis console displays the occupied space.

Notes:

- For master-replica instances and single-node instances, the occupied space is approximately 32 MB.

- For cluster instances, the occupied space is approximately 32 MB multiplied by the number of nodes in a cluster.

10.25. Does ApsaraDB for Redis support common Redis clients such as Jedis?

Yes. Clients compatible with Redis protocols can connect to Alibaba Cloud ApsaraDB for Redis. You can choose any Redis client based on the features of your application. For more information about Redis clients, see Clients.

10.26. Can I modify the REDIS_SHARED_INTEGERS parameter in ApsaraDB for Redis?

In ApsaraDB for Redis, the **REDIS_SHARED_INTEGERS** parameter cannot be modified. Its default value is `10000`.

> **Note** For more information about the instance parameters that can be customized, see Parameter overview and configuration guide.

10.27. Does any ApsaraDB for Redis instance have a read-only replica node?

An ApsaraDB for Redis instance runs in a master-replica structure. But no replica node works as a read-only node.

If you require a read-only node, use a read/write splitting instance.