ALIBABA CLOUD

Alibaba Cloud

ApsaraDB for Redis
User Guide

Document Version: 20220711

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.

2.Limits

This topic describes the limits on data types and some features of ApsaraDB for Redis.

Item	Description
List data type	The numbers of lists and elements in a list are unlimited. The maximum size of each element is 512 MB. We recommend that the number of elements in a list is less than 8,192. The maximum value length is 1 MB.
Set data type	The numbers of sets and elements in a set are unlimited. The maximum size of each element is 512 MB. We recommend that the number of elements in a set is less than 8,192. The maximum value length is 1 MB.
Sorted set data type	The numbers of sorted sets and elements in a sorted set are unlimited. The maximum size of each element is 512 MB. We recommend that the number of elements in a sorted set is less than 8,192. The maximum value length is 1 MB.
Hash data type	The numbers of hash tables and elements in a hash table are not limited. The maximum size of each element is 512 MB. We recommend that the number of elements in a hash table is less than 8,192. The maximum value length is 1 MB.
Number of databases (DBs)	Each instance supports 256 databases. Note ● The total size of data stored in all databases depends on the memory size of an instance. ● The system automatically assigns memory to a single DB based on the usage. The upper limit of assigned memory is the instance memory. For example, if DB 0 occupies all memory, other databases have no data.
Supported Redis commands	For more information, see Overview.
Monitoring and alerting	ApsaraDB for Redis does not provide capacity alerts. You have to configure this feature in Cloud Monitor. For more information, see Alert settings. We recommend that you set alerts for the following metrics: instance faults, instance failover, connection usage, failed operations, capacity usage, write bandwidth usage, and read bandwidth usage.
Policies to delete expired data	● Active expiration: the system periodically detects and deletes expired keys in the background. ● Passive expiration: the system deletes expired keys when you access these keys.
Idle connection recycling mechanism	ApsaraDB for Redis does not automatically recycle idle connections to ApsaraDB for Redis. You can manage the connections.

Item	Description
Data persistence policy	ApsaraDB for Redis uses the `AOF_FSYNC_EVERYSEC` policy and runs the fsync command at a one-second interval.

3.RAM-based access control

3.1. Service linked roles in ApsaraDB for Redis

To use the log management feature of ApsaraDB for Redis, you must assign the AliyunServiceRoleForKvstore role to the ApsaraDB for Redis instance. Then, ApsaraDB for Redis can access Log Service (SLS) resources under your current Alibaba Cloud account.

Background

A service linked role is a Resource Access Management (RAM) role that is associated with a specific cloud service. In most cases, the cloud service automatically creates or deletes a service linked role as needed. You do not need to manually create or delete the service linked role. The service linked role simplifies the process to authorize a service to access other services and avoids the risks that may be caused by user errors. For more information, see Service-linked roles.

> ? **Note** The policy that is attached to a service linked role is predefined by the linked service. You cannot modify or delete the policy. In addition, you cannot add permissions to or remove permissions from a service linked role.

Scenarios

In this topic, the log management feature of ApsaraDB for Redis requires the resources of Log Service. To use the log management feature of ApsaraDB for Redis, you must assign the AliyunServiceRoleForKvstore role to the ApsaraDB for Redis instance.

Introduction to the AliyunServiceRoleForKvstore role

> ? **Note** Log on to the RAM console, click **RAM roles** in the left-side navigation pane, and then enter *AliyunServiceRoleForKvstore* in the search box to search for and view the role.

- Role name: AliyunServiceRoleForKvstore.
- Policy name of the role: AliyunServiceRolePolicyForKvstore.
- Description: ApsaraDB for Redis can use this role to access resources of Log Service and delete service linked roles. The following sample code shows the details of the policy:

  > ? **Note** For more information about the policy syntax, see Policy structure and syntax.

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "acs:log:*:*:project/nosql-*",
            "Action": [
                "log:GetLogstoreLogs",
                "log:ListLogStores",
                "log:GetLogStore",
                "log:GetIndex",
                "log:GetLogstoreHistogram",
                "log:GetConfig",
                "log:ListConfig",
                "log:GetDashboard",
                "log:ListDashboard"
            ]
        },
        {
            "Action": "ram:DeleteServiceLinkedRole",
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ram:ServiceName": "r-kvstore.aliyuncs.com"
                }
            }
        }
    ]
}
```

Permissions required for a RAM user to create a service linked role

The permission to create a service linked role is included in the administrative permission policy of the linked service (for example, AliyunESSFullAccess of ECS). Therefore, after you grant the administrator permissions of a cloud service to a RAM user, the RAM user is allowed to create the service linked role for the cloud service.

If the RAM user does not have the required permissions, you must grant the following permission to the RAM user before you authorize the service linked role. For more information about how to grant permissions, see Create a custom policy and Grant permissions to a RAM user.

```
{
    "Action": "ram:CreateServiceLinkedRole",
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "ram:ServiceName": "r-kvstore.aliyuncs.com"
        }
     }
}
```

Delete a service linked role

To delete the AliyunServiceRoleForKvstore service linked role, you must first release the ApsaraDB for Redis instance that is associated with the role. For more information, see Release 或退订instances and Delete a service-linked role.

Related information

- View slow logs
- View active logs
- Enable the new audit log feature

3.2. Authorize RAM users to manage ApsaraDB for Redis instances by using system policies

To implement fine-grained access control and improve account security, you can use Resource Access Management (RAM) to grant management permissions on ApsaraDB for Redis instances to RAM users. The authorized RAM users can then access ApsaraDB for Redis instances.

Context

RAM is an identity and access control service that is provided by Alibaba Cloud. RAM allows you to create and manage RAM users for employees, systems, applications, and other identities. You can manage the permissions of RAM users to control their access to Alibaba Cloud resources.

If multiple users in your enterprise need to access the same resources, you can use RAM to grant the minimum permissions to these users. This eliminates the need to share the AccessKey pair of your Alibaba Cloud account with these users and reduces security risks. For more information, see What is RAM?.

Scenarios

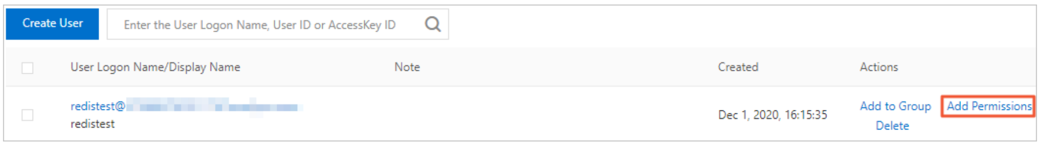
- Authorize a RAM user to manage ApsaraDB for Redis instances in the specified Resource Group.
- Authorize a RAM user to manage all ApsaraDB for Redis instances within your Alibaba Cloud account.

You can create a custom policy to provide finer-grained access control if the default system policies provided by RAM cannot meet your requirements. For more information, see Authorize RAM users to manage ApsaraDB for Redis instances by using custom policies.

Procedure

1. Log on to the RAM console.
2. Create a RAM user.
3. In the left-side navigation pane, click **Users** under **Identities**.
4. On the Users page, find the specific RAM user, and click **Add Permissions** in the **Actions** column.

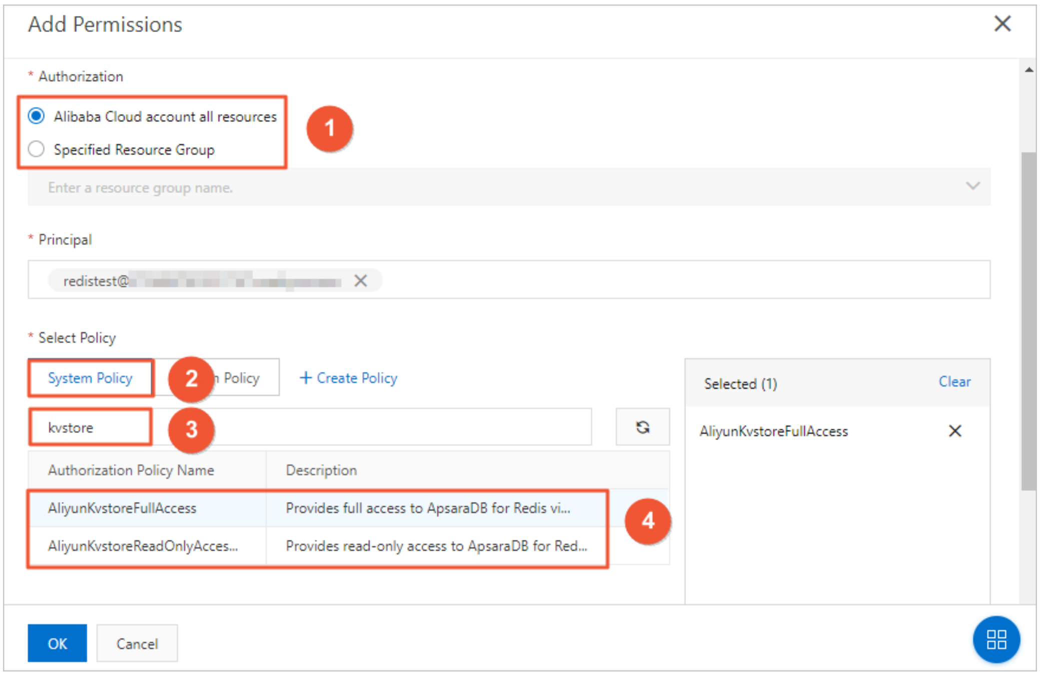
   Add Permissions



□	User Logon Name/Display Name	Note	Created	Actions
□	redistest@[redacted] redistest		Dec 1, 2020, 16:15:35	Add to Group Delete Add Permissions

5. In the Add Permissions dialog box, configure the parameters.

   Add a system policy



Authorization Policy Name	Description
AliyunKvstoreFullAccess	Provides full access to ApsaraDB for Redis vi...
AliyunKvstoreReadOnlyAcces...	Provides read-only access to ApsaraDB for Red...

   i. Select a type of authorization.

   > **Note** If you select **Specified Resource Group**, you must select the specified resource group from the drop-down list. For more information about resource groups, see Resource Group.

   ii. Set Select Policy to **System Policy**.

   iii. Enter *kvstore* in the search box and the system automatically displays the system permission policies related to ApsaraDB for Redis.

   iv. Click a policy name to add the policy to the **Selected** section.

   - **AliyunKvstoreFullAccess**

     This policy has full control permissions on ApsaraDB for Redis instances. The RAM users that are granted with this policy can perform purchase, configuration, and management operations on ApsaraDB for Redis instances.

   - **AliyunKvstoreReadOnlyAccess**

     This policy has read permissions on ApsaraDB for Redis instances. RAM users that are granted with this policy can view information about an ApsaraDB for Redis instance, for example, basic information and performance monitoring metrics. However, they cannot modify the instance configuration, for example, purchasing an instance or configuring a whitelist.

6. Click **OK**.

7. Click **Complete**.

What's next

Log on to the Alibaba Cloud Management Console as a RAM user

3.3. Authorize RAM users to manage ApsaraDB for Redis instances by using custom policies

This topic describes how to create custom policies. Custom policies provide more fine-grained permission control than system policies. You can create custom policies to control the permissions on specific instances or operations.

Context

Resource Access Management (RAM) is an identity and access control service that is provided by Alibaba Cloud. RAM allows you to create and manage RAM users for employees, systems, applications, and other identities. You can manage the permissions of RAM users to control their access to Alibaba Cloud resources.

Scenarios

- Authorize a RAM user to manage specified or all ApsaraDB for Redis instances
- Authorize a RAM user to manage specified ApsaraDB for Redis instances and perform specific operations only. For example, a RAM user is authorized only to configure whitelists.

> ⓘ **Note** In addition to the preceding scenarios, RAM also supports conditions for authorization to take effect. For example, Access Alibaba Cloud resources by using a specific IP address or CIDR block.

If fine-grained permission management is not required, you can grant system policies to RAM users. For more information, see Authorize RAM users to manage ApsaraDB for Redis instances by using system policies.

Step 1: Create a custom policy

1. Log on to the RAM console.
2. In the left-side navigation pane, choose **Permissions > Policies**.
3. On the Policies page, click **Create Policy**.
4. Click the **JSON** tab.

   > ⓘ **Note** **JSON** is used in this example to introduce the configuration method. If you select **Visual Editor Beta**, you must follow the instructions that appear to specify permissions, actions, and resources.

5. Configure the policy and click **Next**.

   The following code provides common custom permission policies. You must replace the `Redis instance ID` in the following code with the instance ID of your ApsaraDB for Redis instance.

Note

- The policy content must be expressed in a specific syntax structure to describe the authorized resource sets, operation sets, and authorization conditions. For more information, see Policy elements and Policy structure and syntax.
- You can grant permissions on specific resources and actions. For more information about the actions that you can grant RAM users to perform, see API operations that can be authorized in RAM.

○Manage all permissions on a single ApsaraDB for Redis instance ○Manage all permissions on multiple ApsaraDB for Redis instances ○Modify whitelist permissions on a single ApsaraDB for Redis instance ○Modify whitelist permissions on multiple ApsaraDB for Redis instances

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kvstore:*",
            "Resource": "acs:kvstore:*:*:*/the ID of your ApsaraDB for Redis instance",
            "Condition": {}
        },
        {
            "Action": "kvstore:Describe*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kvstore:*",
            "Resource": [
                "acs:kvstore:*:*:*/the ID of your ApsaraDB for Redis instance",
                "acs:kvstore:*:*:*/the ID of your ApsaraDB for Redis instance"
            ],
            "Condition": {}
        },
        {
            "Action": "kvstore:Describe*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kvstore:ModifySecurityIps",
            "Resource": "acs:kvstore:*:*:*/the ID of your ApsaraDB for Redis instance",
            "Condition": {}
        },
        {
            "Action": "kvstore:Describe*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

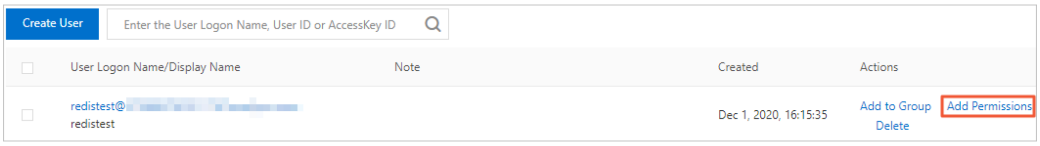
```
{
    "Version": "1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kvstore:ModifySecurityIps",
            "Resource": [
                "acs:kvstore:*:*:*/the ID of your ApsaraDB for Redis instance",
                "acs:kvstore:*:*:*/the ID of your ApsaraDB for Redis instance"
            ],
            "Condition": {}
        },
        {
            "Action": "kvstore:Describe*",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

6. Set **Name** and **Note** (optional) for the policy.
7. Click **OK**.

Step 2: Grant custom permission policies to RAM users

1. Log on to the RAM console.
2. Create a RAM user.
3. In the left-side navigation pane, choose **Identities > Users**.
4. On the Users page, find the specific RAM user, and click **Add Permissions** in the **Actions** column.

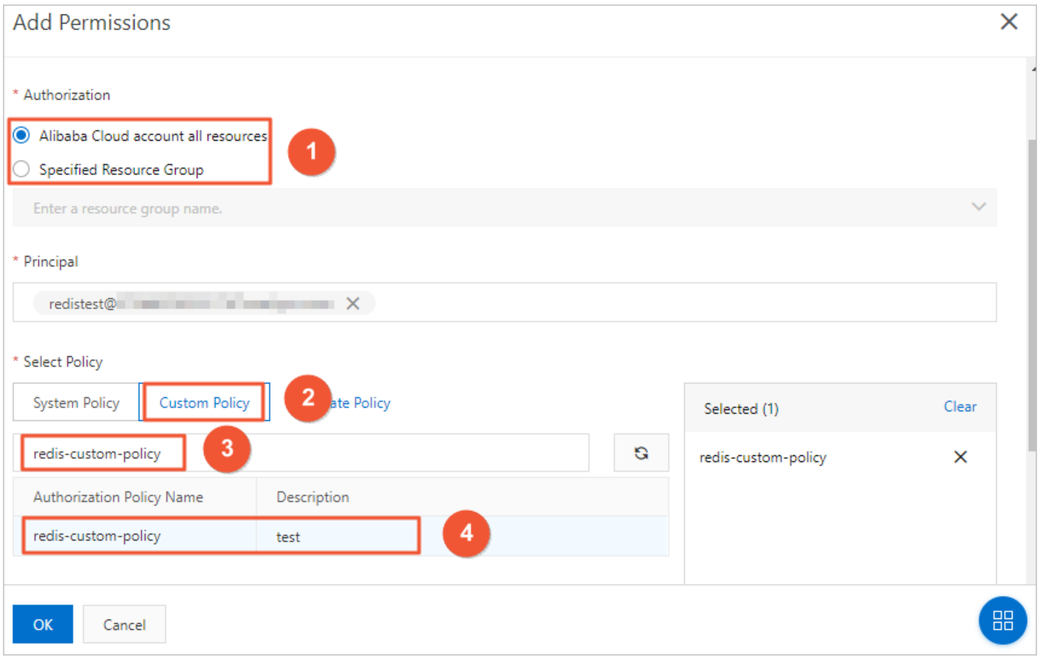
   Add Permissions



□	User Logon Name/Display Name	Note	Created	Actions
□	redistest@[redacted] redistest		Dec 1, 2020, 16:15:35	Add to Group Add Permissions Delete

5. In the Create User dialog box, set the parameters.

   Add permissions



Add Permissions

* Authorization

● Alibaba Cloud account all resources

○ Specified Resource Group

Enter a resource group name.

* Principal

redistest@[redacted]

* Select Policy

redis-custom-policy

Authorization Policy Name	Description
redis-custom-policy	test

Selected (1)

redis-custom-policy

   i. Select a type of authorization.

   > Note If you select **Specified Resource Group**, you must select the specified resource group from the drop-down list. For more information about resource groups, see Resource Group.

   ii. Select **Custom Policy**.

   iii. Enter the name of the permission policy created in Step 1. In this example, enter *redis-custom-policy*.

   iv. Click the name of a custom policy to add the policy to the **Selected** section.

6. Click **OK**.

7. Click **Complete**.

What's next

Log on to the Alibaba Cloud Management Console as a RAM user

4.Manage events

4.1. Query history events

ApsaraDB for Redis allows you to query history events. You can check whether the events that are recorded in the operation logs meet security and regulation compliance requirements.

Procedure

1.
2.
3. In the left-side navigation pane, choose **Event Center > History Events**.

   The system displays details about history events, such as event types, event notifications, and regions in which events occurred.

   > ? Note
   >
   > - On the History Events page, you can find the details of an event, such as the notification, instance ID, and scheduled time. The information about the scheduled time of an event includes the Start At, Scheduled Disconnection Time, and Set Before values. You can click the instance ID to go to the **Instances** page.
   > - ApsaraDB for Redis supports a variety of event types, such as minor version update, instance migration, and master-replica switchover. For more information, see Causes and impacts of events.

Related API operations

Operation	Description
DescribeActiveOperationTask	Queries the detailed information of an O&M task of an ApsaraDB for Redis instance.

References

You can configure alert rules for pending events in the CloudMonitor console. This allows you to handle pending events at the earliest opportunity. For more information, see Alert settings.

4.2. Query and manage pending events

You can receive notifications for ApsaraDB for Redis events such as instance migrations and version upgrades by email, internal message, or by using the ApsaraDB for Redis console. For pending events, you can view event types, regions, procedures, considerations, affected instances, and the default switchover time. You can also adjust the scheduled switchover time.

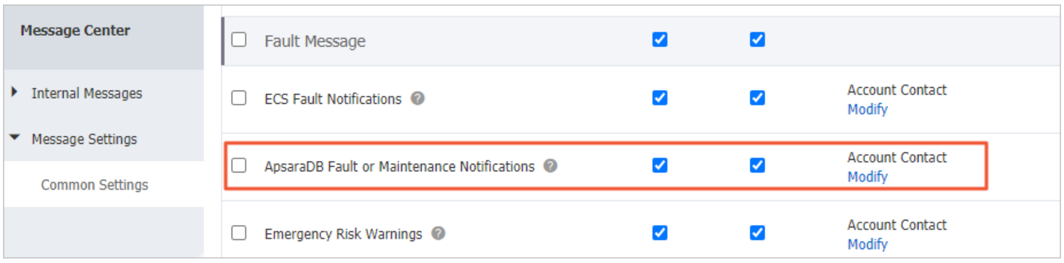
Precautions

- In the upper part of the **Instances** page in the ApsaraDB for Redis console, you can view the number of **pending events**. A value of **0** indicates no pending events.
- In most cases, notifications for system maintenance events such as instance migrations and version upgrades are sent at least three days before execution. Notifications for high-risk vulnerability fixes are sent three or fewer days before execution due to the urgency of these events.

  You can receive notifications by email, internal message, or by using the ApsaraDB for Redis console. To use this feature, log on to the Message Center, select **ApsaraDB Fault or Maintenance Notifications**, and then specify a contact. We recommend that you specify an O&M engineer as the contact.

  > **Note** To be notified of pending event updates such as new pending events and task progress at the earliest opportunity, you can configure alert rules for pending events in the CloudMonitor console. For more information, see Subscribe to event notifications.

  Message Center settings



Message Center

- Internal Messages
- Message Settings
  - Common Settings

☐ Fault Message	☑	☑	
☐ ECS Fault Notifications	☑	☑	Account Contact Modify
☐ ApsaraDB Fault or Maintenance Notifications	☑	☑	Account Contact Modify
☐ Emergency Risk Warnings	☑	☑	Account Contact Modify

Procedure

1.
2. In the left-side navigation pane, choose **Event Center > Scheduled Event**.

   > **Note** You can also click **Pending Events** in the upper part of the Instances page.

3. On the **Scheduled Event** page, view the event details.

   The details about an event include **Instance ID**, **Event Type**, **Cause**, **Business Impact**, and **Scheduled Disconnection Time**. For more information about the causes and impacts on business, see Causes and impacts of events.

   The system automatically performs a switchover on the related instance at the specified **Scheduled Disconnection Time**. If the scheduled switchover time is not during off-peak hours, perform the following steps to modify the **Scheduled Disconnection Time**.

4. (Optional) Modify the **Scheduled Disconnection Time**.

   You can use the default switchover time, modify the scheduled switchover time, or configure the periodic switchover time. The system performs the switchover based on the following priority: scheduled switchover time > periodic switchover time > default switchover time.

Type and description of the switchover time	Procedure
Scheduled switchover time You can adjust the scheduled switchover time of an event based on actual needs.	i. On the **Scheduled Event** page, select one or more events. ii. In the upper-left corner of the **Scheduled Event** page, click **Add Scheduled Event**. iii. In the **Add Scheduled Event** panel, set the scheduled switchover time. If you need to immediately perform the switchover, select **Earliest Execution Time**. After you confirm the settings, the system starts the preparation and immediately performs a switchover after the preparation is complete. **Note** The time specified by **Scheduled Disconnection Time** cannot be later than the time specified by **Set Before**. iv. Check whether the correct events are selected. ▪ If yes, click **Save** to save the settings. ▪ If no, select the correct events on the **Scheduled Event** page and set the scheduled switchover time again.
Periodic switchover time After you specify the periodic switchover time, the system performs switchovers for pending events that have not been executed at the scheduled switchover time and events that have no scheduled switchover time specified at the periodic switchover time. For example, if you set the periodic switchover time to 00:00 on Tuesday, the system performs switchovers on multiple instances related to the pending events at the same time. This reduces the impacts caused by the switchovers.	i. In the upper-left corner of the **Scheduled Event** page, click **Global Scheduler**. ii. In the **Global Scheduler** panel, specify the periodic switchover time in the **Week** or **Month** dimension based on your business requirements. **Note** You can view the specified periodic switchover time in the upper-right corner of the **Scheduled Event** page. If you select **None**, the periodic switchover time is not displayed. iii. Click **Save** to save the settings.

5. (Optional) Cancel the scheduled switchover for an event.

   You can cancel the scheduled switchover for a **Minor Version Update** event. You can select the **Minor Version Update** event and click **Cancel Scheduled Event** in the upper-left corner.

Causes and impacts of events

Cause	Impact	Description
Instance migration	Transient connections	When the switchover is performed at the you may experience the following impacts. We recommend that you perform the switchover during off-peak hours and make sure that your applications are configured to automatically re-establish a connection. • Single-zone instance: One switchover is performed. Your instance or data shards in your instance that are involved in the switchover experience transient connections and stay in the read-only state for up to 30 seconds until all the data is synchronized. • Multi-zone instance: Two switchovers are performed. During the switchovers, your instance or data shards in your instance that are involved in the switchovers experience transient connections and stay in the read-only state for up to 30 seconds until all the data is synchronized. The second switchover switches the master node to the primary zone. This prevents an increase in access latency that may occur when the master node is in the secondary zone. **Note** During the switchover, you cannot manage your instance by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary. scheduled switchover time
Switchover between master and replica nodes		
Zone migration		
Minor version update	Transient connections	When the switchover is performed at the you may experience the following impacts. We recommend that you perform the switchover during off-peak hours and make sure that your applications are configured to automatically re-establish a connection. • Single-zone instance: One switchover is performed. Your instance or data shards in your instance that are involved in the switchover experience transient connections and stay in the read-only state for up to 30 seconds until all the data is synchronized. • Multi-zone instance: Two switchovers are performed. During the switchovers, your instance or data shards in your instance that are involved in the switchovers experience transient connections and stay in the read-only state for up to 30 seconds until all the data is synchronized. The second switchover switches the master node to the primary zone. This prevents an increase in access latency that may occur when the master node is in the secondary zone. **Note** During the switchover, you cannot manage your instance by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary.
	Differences between minor versions	Different minor versions (kernel versions) have different updates. You must take note of the differences between the current minor version and the minor version to which your instance is updated.

Cause	Impact	Description
Minor version update for proxy nodes	Transient connections	When the switchover is performed at the you may experience the following impacts. We recommend that you perform the switchover during off-peak hours and make sure that your applications are configured to automatically re-establish a connection. - Single-zone instance: One switchover is performed. Your instance or data shards in your instance that are involved in the switchover experience transient connections and stay in the read-only state for up to 30 seconds until all the data is synchronized. - Multi-zone instance: Two switchovers are performed. During the switchovers, your instance or data shards in your instance that are involved in the switchovers experience transient connections and stay in the read-only state for up to 30 seconds until all the data is synchronized. The second switchover switches the master node to the primary zone. This prevents an increase in access latency that may occur when the master node is in the secondary zone. **Note** During the switchover, you cannot manage your instance by using Data Management (DMS) or Data Transmission Service (DTS). This impact is temporary.
	Differences between minor versions	Different minor versions have different updates. You must take note of the differences between the current minor version and the minor version to which your proxy nodes are updated.

FAQ

For more information, see FAQ about pending events.

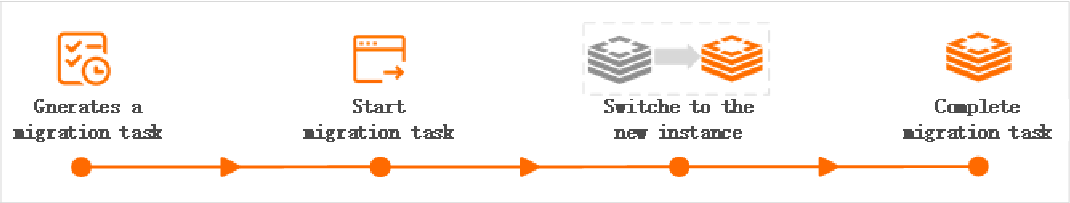
Related API operations

Operation	Description
ModifyActiveOperationTask	Modifies the scheduled switchover time for an O&M task of an ApsaraDB for Redis instance.

4.3. FAQ about pending events

To provide a better experience and improve product performance and stability, Alibaba Cloud performs migration tasks from time to time. The system migrates some of your instances to new server nodes to upgrade software, hardware, and network facilities. This topic provides answers to commonly asked questions about instance migration.

Impact of instance migration events



Time	Description	Impact
Alibaba Cloud generates a migration task	Alibaba Cloud notifies you of the schedule of the migration task by using SMS, voice messages, emails, or internal messages.	No impact.
Start time Note The start time of a task is automatically generated based on the scheduled switchover time.	The system starts to perform operations related to instance migration, such as applying for new instance resources and synchronizing data.	• You can use the database services, but you cannot perform instance-level operations, such as changing configuration and migrating across zones. • You cannot modify the scheduled switchover time when the system is performing the migration task.
Scheduled switchover time	The system switches your workloads to the new instance.	• The instance experiences transient connection errors and stays in the read-only state for 30 seconds or less to wait till all data is synchronized. Upgrade the instance during the off-peak hours and ensure that your application is configured with a reconnection mechanism. Note If the instance is of a cluster or read /write splitting architecture, each shard involved in a switchover experiences transient connection errors and stays in the read-only state for 30 seconds or less. • Data Management Service (DMS) and Data Transmission Service (DTS) are temporarily affected. After the switchover is completed, these services are restored to be normal.
Migration end time	The task related to the instance is removed from the pending events.	No impact. The zone, account, network, and endpoint of the instance are unaffected.

Start time and scheduled switchover time

- Q: How long in advance will be a notification of a migration task sent?

  A: Due to the time gap from when the system sends a notification email to when the actual task is generated, the system sends a notification at least 40 hours in advance. After the scheduled time is reached, the system will switch over to the new instance during the upcoming maintenance window by default. You can adjust the scheduled switchover time on your own. For more information, see Query and manage pending events.

- Q: How can I modify the scheduled switchover time?

  A: You can modify the scheduled switchover time in the ApsaraDB for Redis console or by using the related API operation. For more information, see Query and manage pending events and ModifyActiveOperationTask.

- Q: Why do I fail to modify the scheduled switchover time?

  A: If the event has already been executed, (that is, the migration operation has already been started), you cannot change the scheduled switchover time.

- Q: Can I estimate the end time of a migration task?

  A: Only the start time of the migration is displayed. The end time cannot be estimated because it is affected by multiple factors, such as the network latency, task queue, and data size.

Other FAQ

- Q: Why are instances migrated?

  A: To provide a better experience and improve product performance and stability, Alibaba Cloud will perform migration tasks from time to time. The system migrates some of your instances to new server nodes to upgrade software, hardware, and network facilities.

- Q: Will instances be migrated to other zones?

  A: No. It will only be migrated in the current zone.

- Q: Can I cancel an instance migration task?

  A: You are not allowed to cancel the migration task because the instance migration is a task of high urgency. You can postpone a scheduled switchover time. For example, you can select a time during off-peak hours to perform the switchover task.

- Q: Are data shards or proxy nodes in a cluster or read/write splitting instance switched over in parallel?

  A:

  - Data shards: The system performs the switchover in parallel.
  - Proxy nodes: The system performs the switchover in parallel. A maximum of half of proxy nodes in the ApsaraDB for Redis instance can be concurrently switched over.

- Q: How will the system handle a failed switchover?

  A: The system confirms whether the execution is successful within one minute after a switchover task is completed. If the switchover fails, the system will re-initiate the switchover after a period of time.

- Q: How can I confirm whether an instance migration task is completed?

A: After a task is completed, the task related to the instance is removed from the pending events. You can create an event alert rule. Cloud Monitor monitors tasks and a notification is automatically sent to you when a task is completed. For more information, see Alert settings.

4.4. Subscribe to event notifications

ApsaraDB for Redis is integrated with CloudMonitor. You can configure alert rules for ApsaraDB for Redis in the CloudMonitor console. Alerts are immediately generated when thresholds are exceeded or when events are detected. This allows you to make informed business decisions.

Context

CloudMonitor is a service that monitors Internet applications and Alibaba Cloud resources. For more information, see What is CloudMonitor? You can configure CloudMonitor to notify you of system anomalies. Then, you can automate the anomaly handling process based on alert notifications. CloudMonitor supports the following alert notification methods:

- Send alert notifications by using emails or DingTalk chatbots.
- Push events to Message Service (MNS), Function Compute, Log Service, or the specified callback URL. This allows you to automate the anomaly handling process based on your business requirements.

Step 1: Create an alert rule

> **Note** CloudMonitor sends alerts to alert contacts in alert contact groups. You must first create an alert contact and an alert contact group, and add the alert contact to the alert contact group. For more information, see Create an alert contact or alert contact group.

1.
2.
3.
4. In the left-side navigation pane, click **Alarm Settings**.
5. In the upper-right corner of the page, click **Alert Settings** to go to the CloudMonitor console.
6. In the left-side navigation pane, choose **Event Monitoring > System Events**.
7. Click the **Event Alert** tab.
8. Click **Create Event Alert**. In the panel that appears, configure the parameters.

Parameter	Description
Alert Rule Name	Enter the name of the alert rule. The name can be up to 30 characters in length and can contain letters, digits, and underscores (_).
Event Type	Select **System Event**.
Product Type	Select **ApsaraDB for Redis**. You can also create alert rules for other cloud services. Follow the preceding steps to create alert rules for other cloud services.
Event Type	Keep the default value **All types**.
Event Level	Select the event severity level. You can select **CRITICAL** or **WARN**.

Parameter	Description
Event Name	Select the name of the event. **Note** ◦ The options for this parameter vary based on the value of the Event Level parameter. For more information about the relationship between event types and event severity levels, see System events for ApsaraDB for Redis. For more information about other cloud services, see System events overview. ◦ If you want to test the event notification feature in Step 2, do not select **All Events** for this parameter.
Resource Range	Select **All Resources** or **Application Groups**. If you select **Application Groups**, you must specify the group information. For more information, see Create an application group.
Notification Method	Select the following notification methods based on your business requirements: ◦ **Alert Notification**: This is the default option. You must specify a contact group and a notification method. ◦ **MNS Queue**: pushes the event alert to a specific queue in MNS. For more information, see What is MNS?. ◦ **Function Compute**: pushes the event alert to a specific function in Function Compute. For more information, see Overview. ◦ **URL Callback**: pushes the event alert to a specific callback URL. CloudMonitor delivers event alerts to the specific callback URL by using the POST or GET method. For more information about the method procedure, see Configure callbacks for system event-triggered alerts. ◦ **Log Service**: pushes the event alert to a specific Logstore in Log Service. For more information, see What is Log Service?.

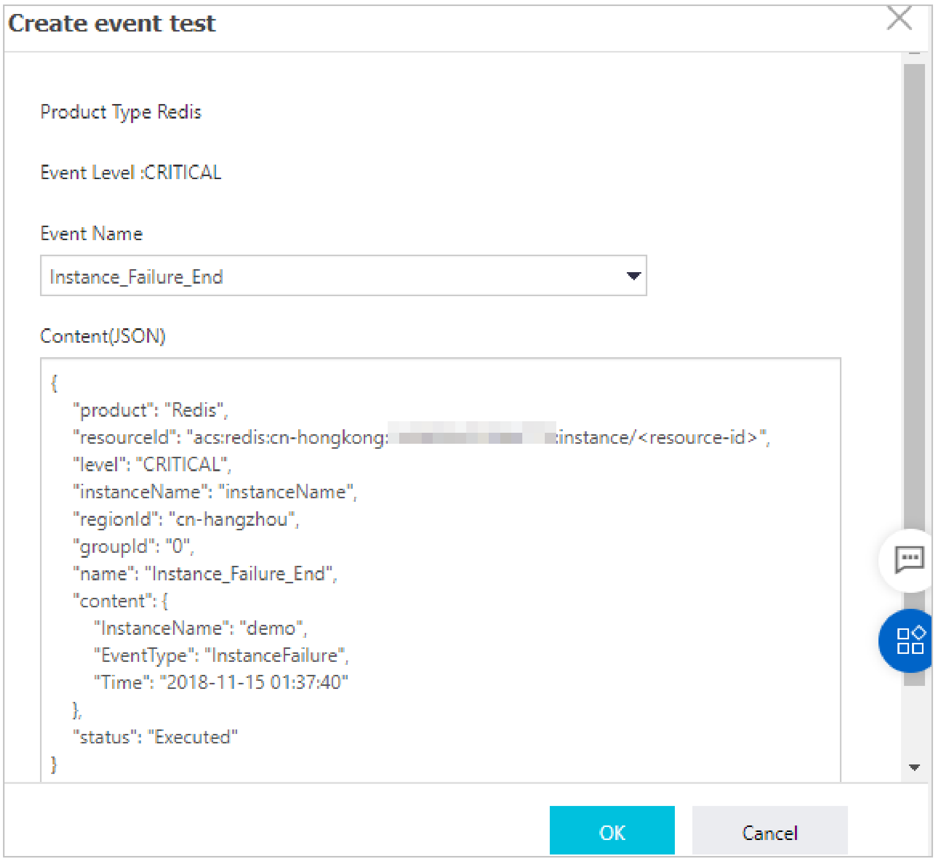
9. Click **OK**.

Step 2: Test the alert rule

After you create a system event-triggered alert rule, you can test the alert rule. You can check whether alert notifications can be received or whether event alerts can be pushed to MNS, Function Compute, Log Service, or the specified callback URL.

1. Log on to the CloudMonitor console.
2. In the left-side navigation pane, choose **Alerts** > **Alert Rules**.
3. Click the **Event Alert** tab.
4. Find the alert rule that you want to test and click **Test** in the **Actions** column.
5. In the **Create event test** panel, select the event that you want to test and modify the content.

   Create an event test



Create event test

Product Type Redis

Event Level :CRITICAL

Event Name

Instance_Failure_End

Content(JSON)

```
{
    "product": "Redis",
    "resourceId": "acs:redis:cn-hongkong:[redacted]:instance/<resource-id>",
    "level": "CRITICAL",
    "instanceName": "instanceName",
    "regionId": "cn-hangzhou",
    "groupId": "0",
    "name": "Instance_Failure_End",
    "content": {
        "InstanceName": "demo",
        "EventType": "InstanceFailure",
        "Time": "2018-11-15 01:37:40"
    },
    "status": "Executed"
}
```

6. Click **OK**.
   CloudMonitor sends an event that contains specific content and an alert is sent by using the specified notification methods. For example, the alert may be sent through a notification and by using MNS.

Related API operations

Operation	Description
PutEventRule	Creates or modifies an event-triggered alert rule.

5.Manage instances

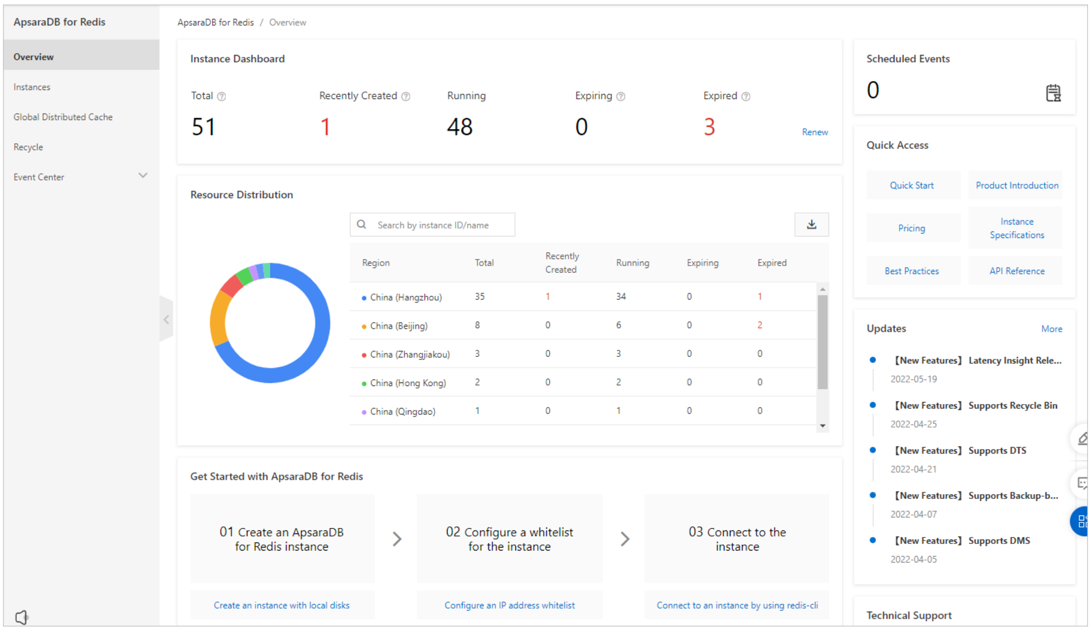
5.1. Overview page of ApsaraDB for Redis

The Overview page in the console displays the instance dashboard, resource distribution, and scheduled events within your account, and provides quick access to common features and updates.

Overview page

You can log on to the ApsaraDB for Redis console and check the overview information on the **Overview** page.

Overview page of ApsaraDB for Redis



ApsaraDB for Redis
Overview
Instances
Global Distributed Cache
Recycle
Event Center
ApsaraDB for Redis / Overview
Instance Dashboard
Renew
Resource Distribution
Search by instance ID/name
Scheduled Events
Quick Access
Quick Start
Product Introduction
Pricing
Instance Specifications
Best Practices
API Reference
Updates
More
Get Started with ApsaraDB for Redis
01 Create an ApsaraDB for Redis instance
02 Configure a whitelist for the instance
03 Connect to the instance
Create an instance with local disks
Configure an IP address whitelist
Connect to an instance by using redis-cli
Technical Support

Total	Recently Created	Running	Expiring	Expired
51	1	48	0	3

Region	Total	Recently Created	Running	Expiring	Expired
China (Hangzhou)	35	1	34	0	1
China (Beijing)	8	0	6	0	2
China (Zhangjiakou)	3	0	3	0	0
China (Hong Kong)	2	0	2	0	0
China (Qingdao)	1	0	1	0	0

Scheduled Events: 0

- 【New Features】 Latency Insight Rele... 2022-05-19
- 【New Features】 Supports Recycle Bin 2022-04-25
- 【New Features】 Supports DTS 2022-04-21
- 【New Features】 Supports Backup-b... 2022-04-07
- 【New Features】 Supports DMS 2022-04-05

- **Instance Dashboard**: displays the total number of instances, , and .

  For instances that are due to expire or have already expired, you can click Renew in the lower-right corner of the Instance Dashboard section to go to the Renewal page. For more information about how to renew an instance, see Renew an instance.

  Recently CreatedExpiringExpired
- **Resource Distribution**: displays instances that belong to a variety of regions.
  - You can click a number in this section to view the instances that correspond to the number. After the instances are displayed, you can click an instance ID to go to the Instance Information page of the instance.
  - You can click the ⤓ icon to export instance details.
  - You can search for an instance by instance ID or instance name in the search box. Fuzzy search is supported.
- **Scheduled Events**: displays the **Pending Events** of all regions. For more information about

scheduled events, see Query and manage pending events.

- **Updates**: displays the latest released features and product updates.
- **Quick Access** and **Get Started with ApsaraDB for Redis**: provide quick access to common features.
- **Technical Support**: allows you to scan the QR code to join a DingTalk group and explore the latest Redis technologies and practices.

> ? **Note** If a Resource Access Management (RAM) user has custom permissions but does not have the permissions to call the DescribeInstancesOverview operation, the number of instances displayed on the Overview page may be inaccurate. For more information about how to grant a RAM user the permissions to call the DescribeInstancesOverview operation, see Modify the document and description of a custom policy.

Related API operations

Operation	Description
DescribeInstancesOverview	Queries the overview information of one or more ApsaraDB for Redis instances.

5.2. Instance states and impacts

This topic describes the different states of an ApsaraDB for Redis instance to help you manage your instances. When you restart or change the configurations of an instance, the state of the instance changes. This may cause issues. For example, a transient connection within seconds occurs on the instance.

Instance states

You can query the state of an instance by using the following methods:

- Use the ApsaraDB for Redis console: You can view the state of instances on the Instances page of the ApsaraDB for Redis console.
- Call an API operation: You can call the DescribeInstances operation to query the state of an instance.

State in the console	State returned by calling the API operation	Operation that leads to the state	Description
Creating	Creating	Step 1: Create an ApsaraDB for Redis instance	After you create an instance, the instance enters the state. The instance cannot provide database services until the instance enters the **Running** state.
Running	Normal	None	The instance can provide database services in this state.

State in the console	State returned by calling the API operation	Operation that leads to the state	Description
Converting	Transforming	Change the billing method to subscription	After you change the billing method of the instance to subscription, you cannot change the billing method back to pay-as-you-go. Before you change the billing method, we recommend that you evaluate your business requirements to make better use of the instance.
Changing Configuration	Changing	Change the configurations of an instance	For more information, see Change process and impacts.
		Adjust the number of shards for an 云盘 ApsaraDB for Redis instance	Changes to the configurations of the instance may cause unstable latency. However, ApsaraDB for Redis supports scaling without leading to transient connections and the read-only state. We recommend that you perform this operation during off-peak hours and make sure that your application can automatically reconnect to the instance.
Restarting	Rebooting	Restart one or more ApsaraDB for Redis instances	A transient connection occurs on the instance. We recommend that you perform this operation during off-peak hours and make sure that your application can automatically reconnect to the instance.

State in the console	State returned by calling the API operation	Operation that leads to the state	Description
Updating	MajorVersion Upgrading	Upgrade the major version	• When you apply for resources, upgrade the replica node, or synchronize data, your ApsaraDB for Redis service remains available. • When you switch your workloads over from the original instance to a new instance or from the master node to the replica node in the original instance, you may experience transient connections that last for a few seconds. The original instance stays in the read-only state for up to 60 seconds until all data is synchronized. We recommend that you upgrade the original instance during off-peak hours and make sure that your application is configured to automatically re-establish a connection. • If the original instance runs Redis 4.0, Bloom filter-related API operations such as **BF.ADD** are no longer supported after you upgrade the major version of the original instance to a version later than Redis 4.0. **Notice** Bloom filter-related API operations on existing instances of ApsaraDB for Redis 4.0 are only for internal use. In addition, new instances that run Redis 4.0 or later no longer support Bloom filter-related API operations. Therefore, if you call Bloom filter-related API operations the new instances, you cannot perform cache analytics and unknown errors may occur. We recommend that you change the original instance into a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair) to support optimized Bloom filters. For more information about performance-enhanced instances, see Performance-enhanced instances.
Upgrading Minor Version	MinorVersion Upgrading	Update the minor version	

State in the console	State returned by calling the API operation	Operation that leads to the state	Description
Configuring Network	NetworkModifying	Change the network type from classic network to VPC	- After you switch the network type of an instance from classic network to VPC, you cannot switch back to classic network. - The instance may experience a transient connection of a few seconds. We recommend that you perform this operation during off-peak hours and make sure that your applications can automatically reconnect to the instance. - When you switch the network type, you can specify whether to retain the classic network endpoint of the instance. If you do not retain the classic network endpoint, the endpoint is released after the network type is switched. Then, clients cannot connect to the instance by using the classic network endpoint. In this case, you must change the database endpoint on your client at the earliest opportunity.
		Change the endpoint or port number of an ApsaraDB for Redis instance	Clients can no longer connect to the instance by using the original endpoints. Update the connection information on the clients at the earliest opportunity.
		Apply for a public endpoint for an ApsaraDB for Redis instance	None
		Release a public endpoint for an ApsaraDB for Redis instance	Clients can no longer connect to the instance by using the original public endpoint. Update the connection information on the clients at the earliest opportunity.
		Enable the direct connection mode	If you enable the direct connection mode, you cannot perform the following operations: - Change the configurations of an instance - Upgrade the major version - Migrate an instance across zones You must release the private endpoint before you can perform the preceding operations.
		Release a private endpoint for an ApsaraDB for Redis instance	Clients can no longer connect to the instance by using the original direct connection endpoint. Update the connection information on the clients at the earliest opportunity.

State in the console	State returned by calling the API operation	Operation that leads to the state	Description
Configuring SSL	SSLModifying	Configure SSL encryption	The instance restarts after you enable SSL encryption or update the certificate validity period. A transient connection occurs on the instance. We recommend that you perform this operation during off-peak hours and make sure that your application can automatically reconnect to the instance.
Migrating to Another Zone	ZoneMigrating	Migrate an instance across zones	• A transient connection occurs on the instance. We recommend that you perform this operation during off-peak hours and make sure that your application can automatically reconnect to the instance. • If you migrate an instance across zones, the virtual IP address (VIP) of the instance such as 172.16.88.60 is changed. However, the endpoint of the instance remains unchanged. We recommend that you connect to the instance by using the endpoint. If you use a VIP to connect to the instance, the connection fails. • If the minor version of the instance is outdated, the system updates it to the latest version to ensure high performance and stability.
Restarting	Rebooting	Restart one or more ApsaraDB for Redis instances	A transient connection occurs on the instance. We recommend that you perform this operation during off-peak hours and make sure that your application can automatically reconnect to the instance.
Flushing Instance	Flushing	Delete data	If you choose to delete all data, take note of the following impacts: • The instance runs the **FLUSHALL** command to delete all data from the instance. The deleted data cannot be recovered. • This operation immediately deletes all data from the instance and adversely affects your online services. Proceed with caution. We recommend that you perform this operation during off-peak hours.

State in the console	State returned by calling the API operation	Operation that leads to the state	Description
Deleting Expired Data	CleaningUpExpiredData		If you choose to delete expired data, take note of the following impacts: • The instance runs the **SCAN** command to delete all the expired data from the instance. The deleted data cannot be recovered. • This operation immediately deletes all the expired data from the instance and adversely affects your online services. Proceed with caution. We recommend that you perform this operation during off-peak hours.
Switching	HASwitching	Manually switch workloads from a master node to a replica node	• The data nodes on which the switchover is performed are disconnected for a few seconds. A switchover has potential data loss risks. For example, the data may become inconsistent between the master and replica nodes due to synchronization latency. To prevent potential data loss risks caused by the switchover and data doublewrite caused by the Domain Name System (DNS) cache, the data nodes become read-only for up to 30 seconds. • After an instance enters the **Switching** state, you cannot manage this instance. For example, you cannot modify the instance configurations or migrate the instance to another zone.
Disabled	Inactive	None	The instance expires and cannot provide database services. If you want to continue to use the instance, you must manually renew the instance. For more information, see Renew an instance.
Released	None	Manually or automatically release an instance that is overdue or expired	The instance is released and cannot provide database services. The instance is not displayed on the Instances page. If you want to continue to use the instance, you can find and restore the instance from the recycle bin. For more information, see Manage instances in the recycle bin.

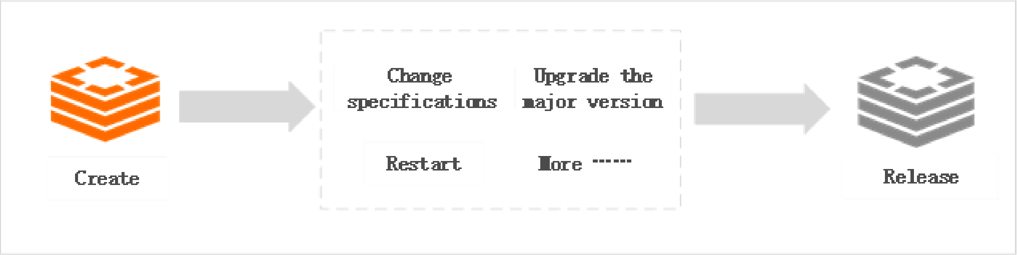
5.3. Lifecycle management

5.3.1. Lifecycle of an ApsaraDB for Redis instance

This topic describes the lifecycle of an ApsaraDB for Redis instance. The lifecycle starts from the time when the instance is created to the time when the instance is released or unsubscribed.

Lifecycle

Lifecycle



Notice If you perform specific operations on an instance, the state of the instance changes. This affects the instance in different ways. For example, the service may be interrupted for a few seconds. For more information, see Instance states and impacts.

Operation	Description
Create an ApsaraDB for Redis instance	ApsaraDB for Redis provides two editions: Community Edition and Enhanced Edition (Tair). ApsaraDB for Redis Enhanced Edition (Tair) has the following types of instance: Performance-enhanced instances, Persistent memory-optimized instances, and Storage-optimized instances. You can choose one based on your business requirements.
Change the configurations of an instance	You can change the specifications, architecture, and type of your instance to meet the performance and compatibility requirements of different scenarios.
Adjust the number of shards for an 云盘 ApsaraDB for Redis instance	A larger number of shards result in higher instance performance. You can adjust the number of shards for an instance based on your business requirements. For more information about Community Edition with cloud disks, see Community Edition with cloud disks.
Restart one or more ApsaraDB for Redis instances	If the number of connections reaches the upper limit or if performance issues occur, you can restart the ApsaraDB for Redis instance to close all connections.
Upgrade the major version	You can upgrade the major version of an ApsaraDB for Redis instance, for example, from Redis 2.8 to Redis 4.0. After the major version of the instance is upgraded, you can use the features of the new version. For more information about new major versions, see the following topics: • New features of ApsaraDB for Redis 7.0 • New features of ApsaraDB for Redis 6.0 • New features of ApsaraDB for Redis 5.0 • Features of ApsaraDB for Redis 4.0
Update the minor version	Alibaba Cloud continuously optimizes the kernel of ApsaraDB for Redis to fix security vulnerabilities and improve service stability. We recommend that you periodically check and update the minor version of your instances at the earliest opportunity.
Release 或退订instances	You can release idle pay-as-you-go instances to save resources.

Operation	Description
Manage instances in the recycle bin	ApsaraDB for Redis provides a recycle bin to store expired, overdue, and released ApsaraDB for Redis instances. You can renew, recreate, or permanently delete instances in the recycle bin.

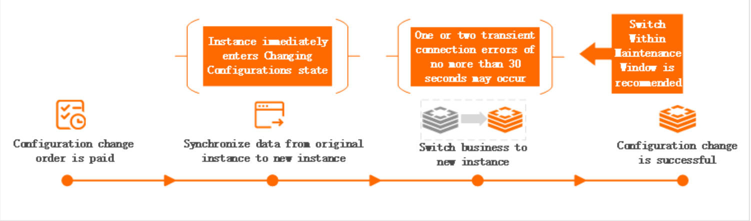
5.3.2. Change the configurations of an instance

ApsaraDB for Redis allows you to change the configurations of instances. You can change the configurations such as the architecture and specifications of an instance to meet different performance and capacity requirements.

Billing

For more information, see Configuration change.

Change process and impacts

Instance	Change process and impact
- Instances that use local disks - Standard instances that use cloud disks	Change process of instances that use local disks  Instance immediately enters Changing Configurations state One or two transient connection errors of no more than 30 seconds may occur Switch Within Maintenance Window is recommended Configuration change order is paid Synchronize data from original instance to new instance Switch business to new instance Configuration change is successful - The cluster architecture imposes requirements on Lua scripts. After you change an instance to the cluster architecture, the Lua scripts may be lost because script content does not meet the requirements. You must back up the Lua scripts in advance. For more information, see Limits on commands supported by cluster instances. - If you change the configurations of an instance, the instance may experience one or two transient connections that last for less than 30 seconds. For Lettuce clients, the instance may be disconnected for 2 to 10 minutes. **Note** We recommend that you set the **Switching Time** parameter to **Switch Within Maintenance Window** when you change the configurations of an instance. In this case, the instance configurations are changed and transient connections occur in the next maintenance window. - To synchronize incremental data from the original instance to the new instance and prevent dual write caused by the Domain Name System (DNS) cache, the instance becomes read-only for less than 1 minute during the configuration change. This ensures data consistency between the new and original instances. - To ensure higher performance and stability, the system updates the minor version of an instance to the latest version during the configuration change.

Instance	Change process and impact
• Cluster instances that use cloud disks	During the configuration change, ApsaraDB for Redis migrates slots to the newly added data shards. This can help achieve imperceptible scaling. During this process, the latency remains under 100 milliseconds.

Note For more information about instances that use local disks and those that use cloud disks, see Comparison between ApsaraDB for Redis instances that use local disks and those that use cloud disks.

Limits

Instance	Limits
Instances that use local disks	• Before you change a non-cluster instance into a cluster instance, you must evaluate how the command limits of cluster instances affect your workloads. For more information, see Limits on commands supported by cluster instances. • If you change the architecture of an instance (such as from standard to cluster), the existing alert settings become invalid and the data flashback feature is disabled. ○ To continue using alerts, you must reconfigure alert settings. For more information, see Alert settings. ○ To continue using data flashback, you must re-enable the feature. For more information, see Use data flashback to restore data by point in time. • If a private endpoint is allocated to an ApsaraDB for Redis cluster instance and you want to change the number of shards for the instance, you can only double the current number of shards. For more information about private endpoints, see Enable the direct connection mode. For example, if the original instance has two shards, you can scale the instance only to four shards. You cannot directly scale the instance to eight shards. If you want to scale the instance from two shards to eight shards, use one of the following methods: ○ Scale the instance from two shards to four shards. Then, scale the instance from four shards to eight shards. ○ Release the private endpoint. Then, directly scale the instance to eight shards. For more information, see Release a private endpoint for an ApsaraDB for Redis instance.
Cluster instances that use cloud disks	• You can change the quantity but not the specifications of data shards. • An instance can contain 2 to 256 shards. You can add or reduce up to 64 shards each time. • The specifications of the shards that you want to add are the same as those of existing shards and cannot be modified. For more information, see Adjust the number of shards for an 云盘ApsaraDB for Redis instance.

Instance	Limits
Child instance in a distributed instance	- You cannot change the architecture of a child instance (such as from cluster to standard). - To change the configurations of a child instance in a distributed instance, you must change the configurations of all the other child instances in the distributed instance in the same way. Otherwise, performance or capacity issues may occur. - You can only double the current number of shards for a child instance. For example, if the original instance has two shards, you can scale the instance only to four shards. You cannot directly scale the instance to eight shards. **Note** If you want to scale the instance from two shards to eight shards, you can scale the instance from two shards to four shards and then scale the instance from four shards to eight shards.

When you downgrade the configurations of an instance, note that 90% of the memory capacity of the new instance is greater than the amount of occupied memory of the original instance. Otherwise, the instance cannot be downgraded. For example, assume that you have a performance-enhanced master-replica instance with 8 GB of memory and that 3 GB of the instance memory is occupied. You can change the instance into a performance-enhanced master-replica instance that has 4 GB of memory.

Manually change the configurations of an instance

1.
2. Perform the corresponding operations described in the following table based on the billing method of your instance.

Billing method	Procedure
Pay-as-you-go	i. In the upper-right corner of the page, click **Change Configurations** . ii. On the **Upgrade/Downgrade** page, make required configuration changes and click **Buy Now**.
Subscription	i. In the upper-right corner of the page, click **Upgrade**. ii. On the **Upgrade/Downgrade** page, make required configuration changes and click **Buy Now**.

Note

- If you want to change a non-cluster instance into a cluster instance, or change a cluster instance into another cluster instance, take note of the following items:
  - You must read and confirm the related limits of cluster instances. For more information about these limits, see Limits on commands supported by cluster instances.
  - If the original instance contains Lua scripts, back up the Lua scripts in advance. Then, select **Force Upgrade** during the configuration change to delete the Lua scripts of the original instance.
- We recommend that you set the **Switching Time** parameter to **Switch Within Maintenance Window** when you change the configurations of an instance. In this case, the instance configurations are changed in the next maintenance window.

3. Pay for the order.

Related API operations

Operation	Description
ModifyInstanceSpec	Changes the configurations of an ApsaraDB for Redis instance.

Related information

-
- Lifecycle of an ApsaraDB for Redis instance

5.3.3. Adjust the number of shards for an ApsaraDB for Redis instance

ApsaraDB for Redis instances with cloud disks are based on the new-generation control architecture of ApsaraDB for Redis. These instances allow you to adjust the number of shards and perform smooth scaling. In smooth scaling, transient connections do not occur and instances do not enter the read-only state. You can use these features to flexibly handle reads and writes on hot data and data skews.

Prerequisites

The instance is a persistent memory-optimized cluster instance. For more information about persistent memory-optimized instances and cluster instances, see Persistent memory-optimized instances and Cluster master-replica instances.

Precautions

- An instance can contain 1 to 32 shards.
- The operation cannot be scheduled to be performed in a maintenance window. After you adjust the number of shards, the instance immediately enters the **Changing Configuration** state and evenly distributes data on the shards.

- The duration of a configuration change is based on multiple factors such as the network conditions, task queue size, and data volume. A configuration change may cause a latency fluctuation. We recommend that you change configurations during off-peak hours. Make sure that your applications can automatically reconnect to instances.

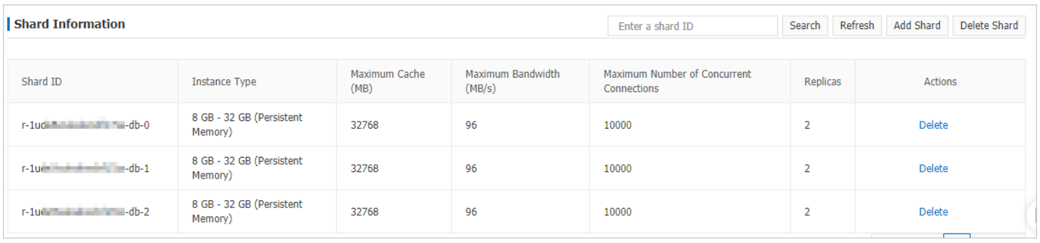
Billing

For more information, see Configuration change.

Procedure

1.
2. In the **Shard Information** section, perform the following operations based on your business requirements.

   Adjust the number of shards

   Shard Information

   Enter a shard ID | Search | Refresh | Add Shard | Delete Shard

Shard ID	Instance Type	Maximum Cache (MB)	Maximum Bandwidth (MB/s)	Maximum Number of Concurrent Connections	Replicas	Actions
r-1ud[redacted]-db-0	8 GB - 32 GB (Persistent Memory)	32768	96	10000	2	Delete
r-1u[redacted]-db-1	8 GB - 32 GB (Persistent Memory)	32768	96	10000	2	Delete
r-1u[redacted]-db-2	8 GB - 32 GB (Persistent Memory)	32768	96	10000	2	Delete

   - Add shards
     a. Click **Add Shard**.
     b. In the panel that appears, specify the number of shards that you want to add to the instance.

     > Note The specifications of the shards to be added are the same as those of existing shards and cannot be modified.

     c. Click **Pay** and complete the payment.
   - Delete shards
     a. Click **Delete Shard**.

     > Note You can also click **Delete** in the **Actions** column corresponding to the shard that you want to delete.

     b. In the dialog box that appears, select the shards that you want to delete from the drop-down list. You can also click **Delete All Shards** and clear the shard that you want to retain to delete all the other shards.
     c. Click **OK**.

Related API operations

Operation	Description
AddShardingNode	Adds one or more shards to an ApsaraDB for Redis cluster instance.

Operation	Description
DeleteShardingNode	Deletes one or more shards from an ApsaraDB for Redis cluster instance.

5.3.4. Restart one or more ApsaraDB for Redis instances

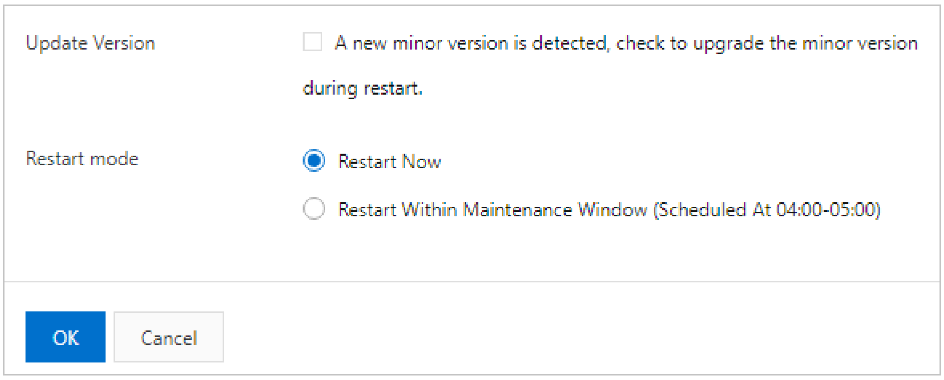
This topic describes how to restart an ApsaraDB for Redis instance to release all connections when the number of connections reaches the upper limit or when performance issues occur.

Impacts

During the restart, an instance may be disconnected within seconds. We recommend that you restart the instance during off-peak hours. Before you restart the instance, make sure that your applications can reconnect to the instance.

Procedure

1.
2. Find the instance that you want to restart and click **restart** in the **Actions** column.
3. In the panel that appears, configure the parameters.



Update Version □ A new minor version is detected, check to upgrade the minor version during restart.

Restart mode ● Restart Now
○ Restart Within Maintenance Window (Scheduled At 04:00-05:00)

Parameter	Description
Update Version	If the system detects a new minor version, this parameter is displayed. If you select the check box, the system upgrades the instance to the latest minor version during the restart. If you do not need to upgrade the minor version, clear the check box.
Restart mode	○ **Restart Now**: immediately restarts the instance. ○ **Restart Within Maintenance Window**: restarts the instance within the specified maintenance window. For more information, see Set a maintenance window.

4. Click **OK**.

5.3.5. Change the billing method to subscription

After you purchase a pay-as-you-go instance, you can change its billing method to subscription.

Prerequisites

- The billing method of the instance is pay-as-you-go and the instance is in the **Running** state.

  > **Note** Before you pay for an order for changing the billing method of a pay-as-you-go instance to subscription, if the state of this instance changes (for example, to Locked), your payment may fail. You can continue to pay for the order only after the state of the instance changes back to Running.

- You have no unpaid orders in your account for the instance for which you want to change the billing method.

Precautions

- Unexpired subscription instances cannot be released.
- An instance starts to be billed on a subscription basis immediately after the billing method of the instance is changed from pay-as-you-go to subscription.
- When you change the billing method of a pay-as-you-go instance to subscription, the system generates an order. The new billing method takes effect only after you pay for this order. If you have unpaid or pending orders,these orders are displayed on the Orders page. In this case, you cannot purchase a new instance or change the billing method of another instance until you pay these orders.

  > **Note**
  >
  > - If you have an unpaid order for changing the billing method of a pay-as-you-go instance to subscription and you have scaled up the instance, the order amount is insufficient for the billing method change due to changed instance configurations. The order cannot be paid. You must cancel this unpaid order and change the billing method of the instance again.
  > - If you want to cancel the order, you can cancel the order on the Orders page in the Billing Management console.

Procedure

1. Log on to the ApsaraDB for Redis console and go to the Instances page. In the top navigation bar, select the region in which the instance is deployed.
2. Find the instance for which you want to change the billing method. In the **Actions** column, choose ⋮ > **Switch to Subscription**.
3. Select a period for **Subscription Duration**.
4. Read and select ApsaraDB for Redis (Subscription) Terms of Service. Click **Buy Now** and pay for the order.

Related API operations

Operation	Description
TransformToPrePaid	Changes the billing method of an ApsaraDB for Redis instance from pay-as-you-go to subscription.

5.3.6. Change the billing method to pay-as-you-go

You can change the billing method of an ApsaraDB for Redis instance from subscription to pay-as-you-go based on your business requirements.

Prerequisites

The billing method of the ApsaraDB for Redis instance is subscription.

Billing

After you change the billing method of an ApsaraDB for Redis instance to pay-as-you-go, a refund is returned by using the method used to pay for the instance. Coupons and vouchers cannot be refunded.

The refund is calculated by using the following formula: Refund = Fee actually paid - Fee for consumed resources.

- The fee actually paid is the money that you paid and does not include the amount covered by coupons or vouchers.
- The fee for consumed resources is calculated based on the following formula: Fee for consumed resources = Daily fee x Consumed subscription duration x Discount for the consumed subscription duration. The daily fee is equal to the order-specific fee divided by 30.

> **Note** The consumed subscription duration is accurate to the day. The part that is less than one day is counted as one day.

Precautions

- When you change the billing method of an ApsaraDB for Redis instance to pay-as-you-go, the workloads on the instance are not interrupted.

> **Notice** Each pay-as-you-go ApsaraDB for Redis instance is billed and paid by hour. To prevent overdue payments that may cause downtime, make sure that your account balance is sufficient.

- The subscription billing method is more cost-effective than the pay-as-you-go billing method, and you are offered higher discounts for longer subscription periods. For long-term use, we recommend that you select the subscription billing method.

Procedure

1.
2. On the Instances page, find the instance for which you want to change the billing method. In the **Actions** column, choose ⋮ > **Switch to Pay-as-you-go Billing**.

3. On the page that appears, read the notes and check the total configuration cost in the lower part of the page.

> **Note** For more information about the refund, see the Billing section of this topic.

4. Read and select ApsaraDB for Redis (Pay-As-You-Go) Terms of Service.
5. Click **Buy Now**.

Related API operations

Operation	Description
TransformInstanceChargeType	Changes the billing method of an ApsaraDB for Redis instance from subscription to pay-as-you-go.

5.3.7. Renew an instance

After an ApsaraDB for Redis subscription instance expires, you must renew the instance within 15 days. We recommend that you manually renew an instance or enable auto-renewal before the instance expires. This avoids service interruptions caused by expired subscriptions.

Prerequisites

The billing method of the instance is subscription.

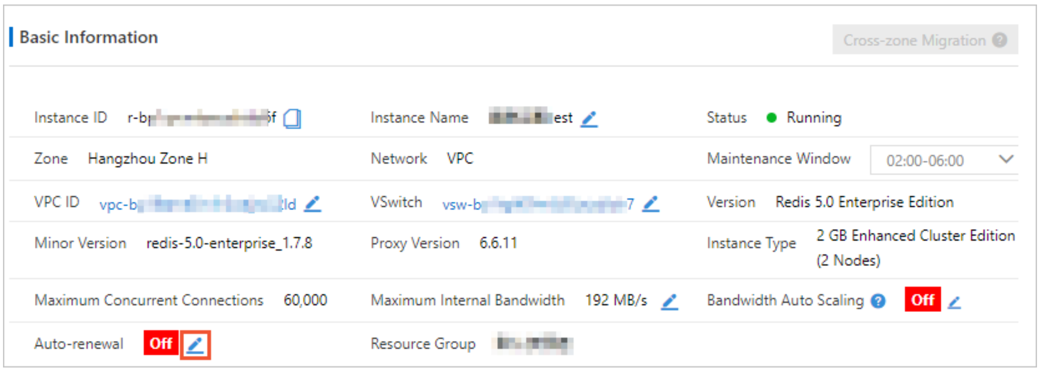
Impacts of expiration

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

Enable auto-renewal for an instance

1.
2. Click the ✎ icon next to **Auto-Renew**.

Enable auto-renewal



Basic Information

Instance ID	r-bp[redacted]f	Instance Name	[redacted]est ✎	Status	● Running
Zone	Hangzhou Zone H	Network	VPC	Maintenance Window	02:00-06:00
VPC ID	vpc-b[redacted]ld ✎	VSwitch	vsw-b[redacted]7 ✎	Version	Redis 5.0 Enterprise Edition
Minor Version	redis-5.0-enterprise_1.7.8	Proxy Version	6.6.11	Instance Type	2 GB Enhanced Cluster Edition (2 Nodes)
Maximum Concurrent Connections	60,000	Maximum Internal Bandwidth	192 MB/s ✎	Bandwidth Auto Scaling	Off ✎
Auto-renewal	Off ✎	Resource Group	[redacted]		

3. In the **Auto-Renew** panel, turn on **Auto-Renew**, read the prompt, and then select a duration.

> Note After you enable auto-renewal, the system automatically renews the instance based on the renewal duration that you specify. For example, if you set the renewal duration to three months, you are charged for a subscription of three months each time the instance is automatically renewed.

4. Click **Auto Renew**.

Manually renew an instance

1.
2. Find the instance that you want to renew and click **Renew** in the **Actions** column.

   > Note To renew multiple instances at a time, you can choose **Expenses > Renewal Management** in the upper part of the page. On the **Renewal** page, select multiple instances and click **Batch Renew**.

3. On the **Renew** page, select a renewal duration.
4. Read and select ApsaraDB for Redis (Subscription) Terms of Service and then click **Pay**.
5. Pay for the order.

Related API operations

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

5.3.8. Upgrade the major version

This topic describes how to upgrade the major version of an ApsaraDB for Redis instance in the ApsaraDB for Redis console to use the features of the new major version. For example, you can upgrade the major version from Redis 2.8 to Redis 4.0.

Workflow

The upgrade workflow varies based on the architecture of the instance. For more information, see the following table.

Architecture	Workflow

Architecture	Workflow
Cluster Read/write splitting	1. Apply for resources that are required to create an instance of a new major version. These resources include proxy node resources. 2. Synchronize full data and incremental data from the original instance to the new instance. 3. Switch your workloads over from the original instance to the new instance. When data is synchronized to completion, the original instance is put into the read-only state and remains in the state for up to 60 seconds until all data is synchronized. After data synchronization is complete, ApsaraDB for Redis disassociates the virtual IP addresses (VIPs) from the original instance and associates the VIPs with the new instance. **Note** If you select **Update During Maintenance**, ApsaraDB for Redis performs the switchover within the specified maintenance window. 4. Check that the upgrade is complete. Then, release the original instance and change the state of the new instance to **Running**.
Standard	1. Upgrade the original replica node. In this step, the system stops the original replica node and creates a replica node of a new major version. 2. Synchronize data from the original master node to the new replica node. 3. Switch your workloads over from the original master node to the new replica node. When data is synchronized to completion, the instance is put into the read-only state and remains in the state for up to 60 seconds until all data is synchronized. After data synchronization is complete, ApsaraDB for Redis disassociates the VIPs from the original master node and associates the VIPs with the new replica node. In addition, ApsaraDB for Redis switches your workloads over from the original master node to the new replica node and promotes the new replica node to the new master node. The original master node is demoted to the new replica node. **Note** If you select **Update During Maintenance**, ApsaraDB for Redis performs the switchover within the specified maintenance window. 4. Upgrade the new replica node (original master node). ApsaraDB for Redis repeats Step 1 and Step 2 to perform the upgrade and synchronize data. 5. Check that the upgrade is complete. If the instance is in the **Running** state, the upgrade is successful.

Impacts

- When you apply for resources, upgrade the replica node, or synchronize data, your ApsaraDB for Redis service remains available.
- When you switch your workloads over from the original instance to a new instance or from the master node to the replica node in the original instance, you may experience transient connections that last for a few seconds. The original instance stays in the read-only state for up to 60 seconds until all data is synchronized. We recommend that you upgrade the original instance during off-peak hours and make sure that your application is configured to automatically re-establish a connection.

- If the original instance runs Redis 4.0, Bloom filter-related API operations such as **BF.ADD** are no longer supported after you upgrade the major version of the original instance to a version later than Redis 4.0.

> **Notice** Bloom filter-related API operations on existing instances of ApsaraDB for Redis 4.0 are only for internal use. In addition, new instances that run Redis 4.0 or later no longer support Bloom filter-related API operations. Therefore, if you call Bloom filter-related API operations the new instances, you cannot perform cache analytics and unknown errors may occur. We recommend that you change the original instance into a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair) to support optimized Bloom filters. For more information about performance-enhanced instances, see Performance-enhanced instances.

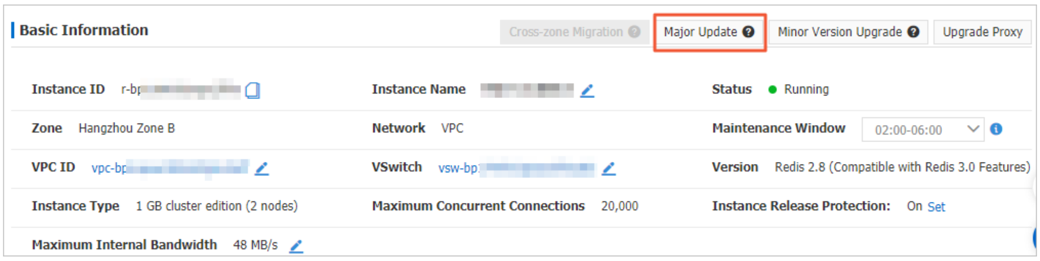
Precautions

- If a private endpoint is allocated to an ApsaraDB for Redis instance or if a Data Transmission Service (DTS) task is created for the instance, a major version upgrade cannot be performed. You can release the private endpoint or close the DTS task before a major version upgrade. For more information about how to release a private endpoint for an instance, see Release a private endpoint for an ApsaraDB for Redis instance.
- Major version upgrades are not needed for ApsaraDB for Redis Enhanced Edition (Tair) instances.

Procedure

1.
2. In the **Basic information** section, click **Major Update**.

   Upgrade the major version



Basic Information

Instance ID	r-bp[redacted]	**Instance Name**	[redacted]	**Status**	● Running
Zone	Hangzhou Zone B	**Network**	VPC	**Maintenance Window**	02:00-06:00
VPC ID	vpc-bp[redacted]	**VSwitch**	vsw-bp[redacted]	**Version**	Redis 2.8 (Compatible with Redis 3.0 Features)
Instance Type	1 GB cluster edition (2 nodes)	**Maximum Concurrent Connections**	20,000	**Instance Release Protection:**	On Set
Maximum Internal Bandwidth	48 MB/s				

> **Note** If the **Major Update** button does not exist, you are using the latest major version.

3. In the panel that appears, specify the new major version and the time when you want to perform the upgrade.

> **Notice** When you switch your workloads over from the original instance to a new instance or from the master node to the replica node in the original instance, you may experience transient connections that last for a few seconds. The original instance stays in the read-only state for up to 60 seconds until all data is synchronized. We recommend that you select **Update During Maintenance**. This way, ApsaraDB for Redis performs the switchover within the specified maintenance window to minimize impacts on your workloads. For more information about how to modify the maintenance window, see Set a maintenance window.

4. Click **OK**.

Related API operations

Operation	Description
ModifyInstanceMajorVersion	Upgrades the major version of an ApsaraDB for Redis instance.

FAQ

- Q: Why does the state of an instance change to Upgrading Major Version after I select **Update During Maintenance** to upgrade the major version?

  A: The state of an instance changes to Upgrading Major Version because the system is preparing for the upgrade. During the preparation, the instance service remains available. When the system prepares for upgrades, such as applying for resources and synchronizing data, instances or master and replica nodes are not switched over and services provided are not affected.

  > ? **Note** The instance experiences transient connections for a few seconds and then remains in the read-only state for up to 60 seconds only during instance switchovers or master/replica switchovers.

- Q: Why does a major version upgrade of an instance fail?

  A: If your instance is of a phrased-out instance type, its major version cannot be upgraded.In this case, you must change the instance type. You can select a valid instance type for the new instance that has the same specifications as the original instance. Then, you can upgrade the major version of the instance. For more information, see .

Related information

- New features of ApsaraDB for Redis 5.0
- Features of ApsaraDB for Redis 4.0

5.3.9. Update the minor version

Alibaba Cloud continuously optimizes the kernel of ApsaraDB for Redis to provide more features or fix known issues and enhance service stability. You can update the kernel version (minor version) of an ApsaraDB for Redis instance with a few clicks in the ApsaraDB for Redis console.

Workflow

Object	Architecture	Workflow

Object	Architecture	Workflow
Instance minor version	Cluster Read/write splitting	1. Apply for resources that are required to create an instance of a new major version. These resources include proxy node resources. 2. Synchronize full data and incremental data from the original instance to the new instance. 3. Switch your workloads over from the original instance to the new instance. When data is synchronized to completion, the original instance is put into the read-only state and remains in the state for up to 60 seconds until all data is synchronized. After data synchronization is complete, ApsaraDB for Redis disassociates the virtual IP addresses (VIPs) from the original instance and associates the VIPs with the new instance. **Note** If you select **Update During Maintenance**, ApsaraDB for Redis performs the switchover within the specified maintenance window. 4. Check that the upgrade is complete. Then, release the original instance and change the state of the new instance to **Running**.
	Standard	1. Upgrade the original replica node. In this step, the system stops the original replica node and creates a replica node of a new major version. 2. Synchronize data from the original master node to the new replica node. 3. Switch your workloads over from the original master node to the new replica node. When data is synchronized to completion, the instance is put into the read-only state and remains in the state for up to 60 seconds until all data is synchronized. After data synchronization is complete, ApsaraDB for Redis disassociates the VIPs from the original master node and associates the VIPs with the new replica node. In addition, ApsaraDB for Redis switches your workloads over from the original master node to the new replica node and promotes the new replica node to the new master node. The original master node is demoted to the new replica node. **Note** If you select **Update During Maintenance**, ApsaraDB for Redis performs the switchover within the specified maintenance window. 4. Upgrade the new replica node (original master node). ApsaraDB for Redis repeats Step 1 and Step 2 to perform the upgrade and synchronize data. 5. Check that the upgrade is complete. If the instance is in the **Running** state, the upgrade is successful.

Object	Architecture	Workflow
Proxy minor version	Cluster Read/write splitting	Proxy nodes support hot updates. A proxy node of a new version can restore a connection based on the client connection information of the proxy node of the earlier version. This ensures uninterrupted connections. However, a millisecond-level latency jitter may occur during the update.

Impacts

Object	Impact
Instance minor version	• When you apply for resources, upgrade the replica node, or synchronize data, your ApsaraDB for Redis service remains available. • When you switch your workloads over from the original instance to a new instance or from the master node to the replica node in the original instance, you may experience transient connections that last for a few seconds. The original instance stays in the read-only state for up to 60 seconds until all data is synchronized. We recommend that you upgrade the original instance during off-peak hours and make sure that your application is configured to automatically re-establish a connection. • If the original instance runs Redis 4.0, Bloom filter-related API operations such as **BF.ADD** are no longer supported after you upgrade the major version of the original instance to a version later than Redis 4.0. **Notice** Bloom filter-related API operations on existing instances of ApsaraDB for Redis 4.0 are only for internal use. In addition, new instances that run Redis 4.0 or later no longer support Bloom filter-related API operations. Therefore, if you call Bloom filter-related API operations the new instances, you cannot perform cache analytics and unknown errors may occur. We recommend that you change the original instance into a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair) to support optimized Bloom filters. For more information about performance-enhanced instances, see Performance-enhanced instances.
Proxy minor version	• Proxy nodes support hot upgrades. Proxy nodes of the new version can restore a connection based on the client connection information of proxy nodes of the earlier version. This ensures that upgrades do not interrupt services. However, a millisecond-level latency jitter may occur during the upgrades. • The hot upgrades are valid only for normal connections. The execution of the block, transaction, Pub, and Sub commands is interrupted during hot upgrades. Make sure that these commands support the reconnection mechanism. • If a Redis client uses a private endpoint to connect to the ApsaraDB for Redis instance, no commands are affected by a proxy upgrade. **Note** For more information about proxy endpoints and private endpoints, see Proxy mode and Direct connection mode.

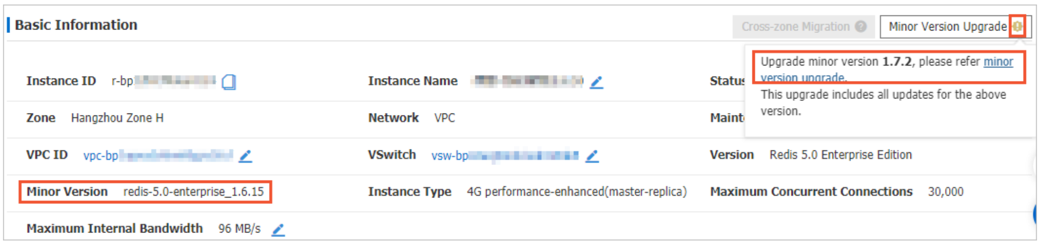
Precautions

Grayscale releases of later instance minor versions may be implemented only in specific regions. The system checks the minor version of your instance. If your instance is of the latest minor version, the **Minor Version Upgrade** or **Upgrade Proxy** button in the ApsaraDB for Redis console cannot be found or clicked.

Procedure

1.
2. In the **Basic information** section, move the pointer over the icon on the right of **Minor Version Upgrade** or **Upgrade Proxy** to view the minor version of the current instance, the minor version to which you can update, and the release notes of minor versions.

   View release notes of minor versions



Basic Information

Upgrade minor version **1.7.2**, please refer minor version upgrade.
This upgrade includes all updates for the above version.

Instance ID	r-bp[redacted]	**Instance Name**	[redacted]	**Status**	
Zone	Hangzhou Zone H	**Network**	VPC	**Maint**	
VPC ID	vpc-bp[redacted]	**VSwitch**	vsw-bp[redacted]	**Version**	Redis 5.0 Enterprise Edition
Minor Version	redis-5.0-enterprise_1.6.15	**Instance Type**	4G performance-enhanced(master-replica)	**Maximum Concurrent Connections**	30,000
Maximum Internal Bandwidth	96 MB/s				

   The icon changes colors based on the minor version update level. The update level is displayed in green, yellow, or red to represent the regular, recommended, or critical update.

Update level	Color	Description
LOW	Green	Regular update. This level includes routine feature updates, such as adding a feature.
MEDIUM	Yellow	Recommended update. This level includes optimization of features and modules. LOW-level updates are also included in MEDIUM-level updates.
HIGH	Red	Critical update. This level includes major updates that ensure stability or security, such as fixing a vulnerability or defect. LOW-level and MEDIUM-level updates are also included in HIGH-level updates.

> **Note** For the complete release notes of minor versions, see ApsaraDB for Redis Enhanced Edition (Tair), ApsaraDB for Redis Community Edition, and ApsaraDB for Redis proxy nodes.

3. After you view the release notes of minor versions, click **Minor Version Upgrade** or **Upgrade Proxy**.
4. In the panel that appears, select the time when you want to perform the update.

> ? **Note** The instance experiences transient connections for a few seconds and stays in the read-only state for up to 60 seconds during instance switchovers or master/replica switchovers. We recommend that you click **Update During Maintenance** to configure switchovers to be performed during the maintenance window of the instance. This way, the impacts of switchovers are minimized for the instance. For more information about how to modify the maintenance window, see Set a maintenance window.

5. Click **OK**.

FAQ

Q: Why does an instance change to the Upgrading Minor Version state after I select **Update During Maintenance** to update the minor version?

A: When the system prepares for updates, such as applying for resources and synchronizing data, instances or master and replica nodes are not switched over. This way, instances that provide services are not affected.

> ? **Note** The instance experiences transient connections for a few seconds and then remains in the read-only state for up to 60 seconds only during instance switchovers or master/replica switchovers.

Related API operations

Operation	Description
DescribeEngineVersion	Queries the major version and minor version of an ApsaraDB for Redis instance and the release notes of minor versions.
ModifyInstanceMinorVersion	Updates the minor version of an ApsaraDB for Redis instance.

Related information

- Upgrade proxy nodes

5.3.10. Release instances

You can release idle pay-as-you-go instances to save resources.

Prerequisites

The billing method of the instances that you want to release is pay-as-you-go.

Precautions

- Only instances in the Running state can be released.
- After an instance is released, it cannot be restored. Proceed with caution. We recommend that you create a backup for the instance and download the backup before you release the instance. For more information, see Backup and restoration solutions.
- Child instances that belong to one or more distributed instances must be released on the **Global**

Distributed Cache tab of the Global Distributed Cache page in the ApsaraDB for Redis console. For more information, see Release a distributed instance.

- ApsaraDB for Redis moves a released instance to the recycle bin if the instance meets specific requirements. For more information, see Manage instances in the recycle bin.

Procedure

1.
2. Find the instance that you want to release and choose ⋮ > **Release** in the **Actions** column.
3. In the Release Instance panel, click **OK**.

Related API operations

Operation	Description
DeleteInstance	Releases a pay-as-you-go ApsaraDB for Redis instance. **Note** You cannot release a subscription ApsaraDB for Redis instance by calling an API operation.

5.3.11. Manage instances in the recycle bin

ApsaraDB for Redis provides a recycle bin to store ApsaraDB for Redis instances that have expired, have overdue payments, or are released. You can unlock, recreate, or permanently delete instances in the recycle bin.

Precautions

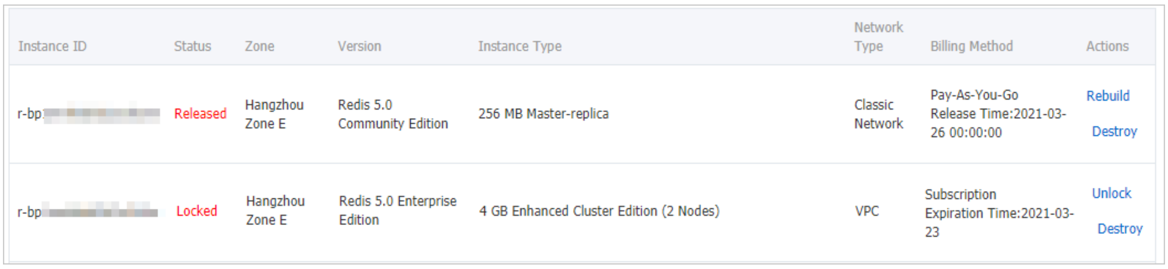
- When an ApsaraDB for Redis instance expires or has overdue payments, the instance is handled based on the amount of time elapsed since the expiration date and the billing method of the instance. For more information, see Expiration, overdue payments, and renewal.

  Warning When the grace period of the instance expires, the system permanently deletes the instance.

- Pay-as-you-go instances that are manually released are retained in the recycle bin for 38 days and then are permanently deleted.

Procedure

1. Log on to the ApsaraDB for Redis console. In the left-side navigation pane, click Recycle. In the upper-left part of the page, select the region where the instance that you want to manage resides.
2. On the **Recycle** page, find the instance that you want to manage and click a button in the **Actions** column based on your business requirements to perform an operation.



Instance ID	Status	Zone	Version	Instance Type	Network Type	Billing Method	Actions
r-bp[redacted]	Released	Hangzhou Zone E	Redis 5.0 Community Edition	256 MB Master-replica	Classic Network	Pay-As-You-Go Release Time:2021-03-26 00:00:00	Rebuild Destroy
r-bp[redacted]	Locked	Hangzhou Zone E	Redis 5.0 Enterprise Edition	4 GB Enhanced Cluster Edition (2 Nodes)	VPC	Subscription Expiration Time:2021-03-23	Unlock Destroy

Operation	Description
Unlock	Renews the instance. After the instance is renewed, it enters the **Running** state and continues to provide services.
Rebuild	Restores all data and specific configurations of the instance to a new instance. The following configurations can be restored: ○ IP address whitelists. **Note** The security group settings in the whitelists of the instance cannot be restored. You must reconfigure the whitelists for security groups. For more information, see Add security groups. ○ The password of the account. On the buy page, set **Set Password** to **Later**. ○ The name of the instance. On the buy page, leave the **Instance Name** field empty. ○ The port number. ○ Parameter settings. ○ Backup settings, such as the backup cycle and backup time.
Destroy	Deletes the instance and its data backup permanently. **Warning** After you delete the instance, data in the instance is permanently deleted and can no longer be restored. Proceed with caution.

5.4. Manage bandwidth

5.4.1. Adjust the bandwidth of an ApsaraDB for Redis instance

This topic describes how to adjust the bandwidth of an ApsaraDB for Redis instance. If the bandwidth of an instance is insufficient to handle unexpected or scheduled traffic spikes during activities such as flash sales, you can increase the bandwidth for the instance so that you can focus on business improvements.

Prerequisites

The instance is an instance of the ApsaraDB for Redis Community Edition or a performance-enhanced or hybrid-storage instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances and hybrid-storage instances, see Performance-enhanced instances and Hybrid-storage instances (phased out).

Scenarios

The bandwidth of an instance varies based on the instance type. If the traffic of an instance reaches the allocated bandwidth, network congestion may occur and instance performance may deteriorate. In the following scenarios, you can adjust the bandwidth of an instance to prevent network congestion and performance deterioration. A bandwidth adjustment helps you increase bandwidth at a lower cost than an instance specification change and can prevent transient connections.

Note We recommend that you use the bandwidth auto scaling feature. When the bandwidth usage of an instance reaches a specified threshold, the bandwidth is automatically increased or decreased. This feature reduces costs and facilitates O&M. For more information, see 开启带宽弹性伸缩.

Scenario	Description
Handle traffic spikes	You can adjust the bandwidth of the instance to handle traffic spikes during promotional events such as a flash sale. After these events end, you can decrease the bandwidth of the instance to reduce costs.
Mitigate impacts on your business	If a large number of read and write operations are performed on large keys within a short period of time, you must temporarily increase the bandwidth of the instance to mitigate impacts on your business and to reserve time to process these operations.
Deal with skewed requests at low costs	If the instance is a cluster or read/write splitting instance, some data shards or read replicas of the instance are frequently accessed. As a result, the bandwidth usages of these data shards or read replicas frequently reach the allocated bandwidths. However, the bandwidth usages of other data shards or read replicas are low. In this case, you can increase the bandwidths only of specific data shards or read replicas instead of increasing the bandwidth or changing the specifications for the instance. For more information about cluster and read/write splitting instances, see Cluster master-replica instances and Read/write splitting instances.
Increase the bandwidth of a standard instance	If the instance is a standard instance of the highest specifications and the bandwidth of the instance does not meet your business requirements, you can increase the bandwidth of the instance without the need to upgrade the instance to a cluster instance. This allows you to focus on business improvements. For more information about standard instances, see Standard master-replica instances.

Limits

- The bandwidth of an instance can be increased to up to three times the bandwidth that is provided for the instance type. If you require higher bandwidth for an instance, you can change the specifications of the instance or upgrade the instance from a standard instance to a cluster instance. Then, you can increase the bandwidth of the instance. If the increased bandwidth is still insufficient, submit a ticket.

Note For more information about the bandwidths that are supported by different instance types, see Overview.

- If you perform the following operations on an instance, the extra bandwidth that you purchased becomes invalid and you receive a refund. You must re-adjust the instance bandwidth based on your business requirements. By default, the adjusted bandwidth inherits the expiration time of the previously purchased extra bandwidth.

Operation	Exception
Upgrade the major version	None
Change the configurations of an instance	If the instance is a standard instance, the bandwidth settings remain valid after you change the specifications of the instance.
Migrate an instance across zones	If the instance is a standard instance, the bandwidth settings remain valid.

Billing

You are charged per day based on the amount and usage duration of the extra bandwidth that you purchase. The fees vary based on the region that you select. For more information, see Billable items and prices.

Note You are not charged for the default bandwidth that is provided for the instance type. You are charged only for the extra bandwidth that you purchase.

Procedure

1.

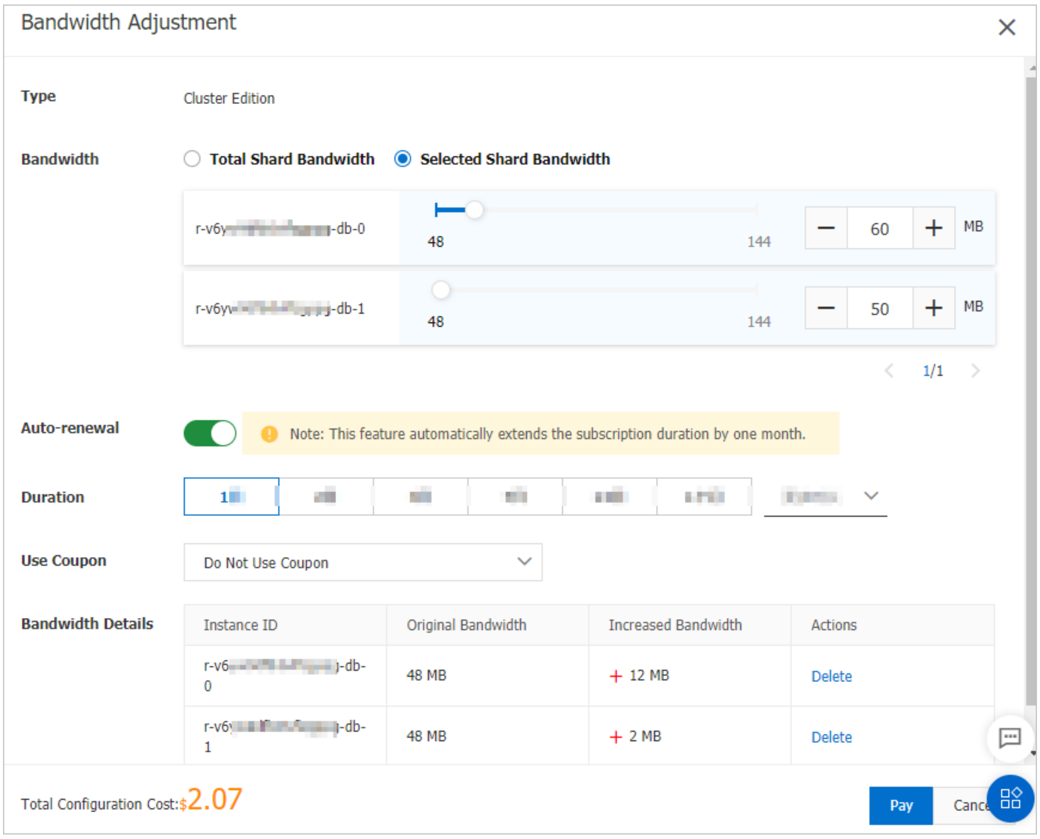
2. In the **Basic Information** section, click the ✎ icon next to **Maximum Internal Bandwidth**.

   Note

   - In the Basic Information section, you can view the maximum internal bandwidth of the instance. If the instance is a cluster instance or a read/write splitting instance, the bandwidth is the sum of the bandwidths that are allocated to all data nodes within the instance.
   - During off-peak hours, you can reduce the bandwidth of the instance to save resources and reduce costs. For more information about how to view the traffic usage of an instance, see View monitoring data.

3. In the panel that appears, specify the amount and subscription duration of the extra bandwidth that you want to purchase.

   Adjust the bandwidth of an instance



Bandwidth Adjustment

Type: Cluster Edition

Bandwidth: ○ Total Shard Bandwidth ● Selected Shard Bandwidth

r-v6y[redacted]-db-0	48 – 144	− 60 +	MB
r-v6yv[redacted]-db-1	48 – 144	− 50 +	MB

1/1

Auto-renewal: Note: This feature automatically extends the subscription duration by one month.

Duration: 1[redacted] | [redacted] | [redacted] | [redacted] | [redacted] | [redacted] | [redacted]

Use Coupon: Do Not Use Coupon

Bandwidth Details:

Instance ID	Original Bandwidth	Increased Bandwidth	Actions
r-v6[redacted]-db-0	48 MB	+ 12 MB	Delete
r-v6[redacted]-db-1	48 MB	+ 2 MB	Delete

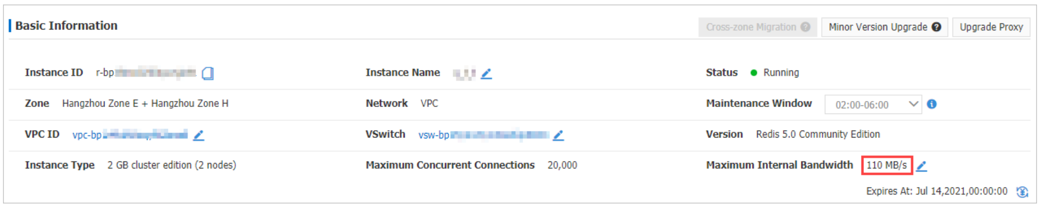
Total Configuration Cost:$2.07

Parameter	Description

Parameter	Description
Bandwidth	Select the maximum extra bandwidth that you want to purchase. The bandwidth of an instance can be increased to up to three times the bandwidth that is provided for the instance type. For more information about the bandwidths that are provided for various instance types, see Overview. The maximum extra bandwidth that you can select varies based on the instance type. ○ If the instance is a standard instance, adjust the bandwidth of the instance. ○ If the instance is a cluster instance or a read/write splitting instance, select one of the following options: ■ **Total Shard Bandwidth**: allows you to adjust the bandwidths of all data shards or data nodes within the instance at the same time. For example, if you want to handle traffic spikes during activities such as flash sales, you can select this option to increase bandwidths, instead of changing the specifications of the instance. ■ **Selected Shard Bandwidth**: allows you to adjust the bandwidths of one or more specific data shards or data nodes. For example, if you want to process skewed requests, you can select this option to adjust the bandwidths of the selected data nodes based on how requests are skewed. Note You can use the diagnostic report feature to analyze whether an instance receives skewed requests. For more information, see Create a diagnostic report.
Auto-Renew	Specify whether to enable auto-renewal. If auto-renewal is enabled, the renewal cycle is fixed to one month.
Duration	Select a subscription period. The maximum subscription period spans five years. Note If the previously purchased extra bandwidth has not expired, the subscription period of the previously purchased extra bandwidth is inherited and this option is not displayed.

4. Click **Pay** and complete the payment.
After you complete the payment, wait for about 1 minute. Then, the maximum internal bandwidth of the instance is updated to the sum of the existing bandwidth and the purchased extra bandwidth, and the expiration time of the extra bandwidth is displayed.

Updated bandwidth after a bandwidth adjustment



Basic Information

Instance ID r-bp[redacted]

Instance Name [redacted]

Status ● Running

Zone Hangzhou Zone E + Hangzhou Zone H

Network VPC

Maintenance Window 02:00-06:00

VPC ID vpc-bp[redacted]

VSwitch vsw-bp[redacted]

Version Redis 5.0 Community Edition

Instance Type 2 GB cluster edition (2 nodes)

Maximum Concurrent Connections 20,000

Maximum Internal Bandwidth 110 MB/s

Expires At: Jul 14,2021,00:00:00

> **Note** You can click the icon next to the expiration time of the extra bandwidth to increase the subscription period.

FAQ

- Q: Can I adjust the bandwidth of a specific data shard or data node within a master-replica cluster instance or a read/write splitting instance?

  A: Yes, you can adjust the bandwidth of a specific data shard or data node within a master-replica cluster instance or a read/write splitting instance based on your business requirements. This way, you can handle skewed requests with more flexibility.

- Q: Does a bandwidth adjustment cause transient connections?

  A: No, a bandwidth adjustment immediately takes effect and does not cause transient connections.

- Q: If I purchase a specific amount of extra bandwidth at 15:00 on March 1, 2021 and select a one-day subscription period, when does the extra bandwidth that I purchase expire?

  A: The extra bandwidth that you purchase expires at 00:00 on March 3, 2021. You are not charged for the usage of the extra bandwidth on March 1, 2021.

- Q: How do I unsubscribe from the extra bandwidth that I purchase for my instance?

  A: Move the bandwidth adjustment slider to the leftmost position to decrease the bandwidth of your instance to the value provided for the instance type. Then, check the refund to your Alibaba Cloud account and complete the payment.

  > **Note** You can go to the Orders page to view the progress of the order.

- Q: My subscription instance has expired and I do not plan to renew the instance. If the extra bandwidth that I purchased does not expire until next month, can I receive a refund?

  A: Yes, after your instance is locked, you will receive a refund based on the remaining subscription period of the extra bandwidth that you purchased. If the extra bandwidth that you purchased expires before the instance, Alibaba Cloud sends you a notification. This helps prevent the impacts of bandwidth changes on your business.

- Q: How do I check whether auto-renewal is enabled for the extra bandwidth that I purchased for my instance?

  A: Go to the Renewal page. Then, enter the ID of your instance and the *-bw* suffix in the **Instance ID** field. Example: *r-bp1zxszhcgatnx****-bw*.

- Q: If I change the specifications of my instance after I purchase a specific amount of extra bandwidth, am I still charged for the extra bandwidth?

  A: If you change the specifications of your instance after you purchase a specific amount of extra bandwidth, the extra bandwidth remains valid and you are charged for the extra bandwidth. The bandwidth of your instance is the sum of the bandwidth that is provided for the new instance type and the extra bandwidth that you purchase. These rules apply only to standard instances. For other instance architectures, the extra bandwidth that you purchase becomes invalid and you receive a refund after you change the specifications of your instance. In this case, you must adjust the bandwidth of the instance again if needs arise.

> ? **Note** For more information about the bandwidths that are supported by different instance types, see Overview.

Related API operations

Operation	Description
EnableAdditionalBandwidth	Adjusts the bandwidth of an ApsaraDB for Redis instance.

5.5. Network connection management

5.5.1. Change the VPC or vSwitch of an ApsaraDB for Redis instance

You can change the virtual private cloud (VPC) or vSwitch of an ApsaraDB for Redis instance. For example, you can change the VPC of an ApsaraDB for Redis instance to the VPC to which an Elastic Compute Service (ECS) instance belongs. This way, the ApsaraDB for Redis instance can communicate with the ECS instance over the internal network.

Prerequisites

- The instance is deployed in a VPC.

> ? **Note** You can view the network type of an instance on the **Instance Information** page. If the network type is classic network, you can change the network type from classic network to VPC. For more information, see Change the network type from classic network to VPC.

- The instance does not have a private endpoint. If the direct connection mode is enabled for the instance, you can release the private endpoint, change the VPC of instance, and then re-enable the direct connection mode. For more information about private endpoints and the direct connection mode, see Enable the direct connection mode.
- The password-free access feature is disabled for the instance. No Data Transmission Service (DTS) data migration tasks or synchronization tasks that involve the instance are running. Otherwise, an error is reported. For more information about password-free access and DTS, see Enable password-free access and What is DTS?

Scenarios

Operation	Scenario
Change the VPC of an ApsaraDB for Redis instance	Clients are unable to communicate with an ApsaraDB for Redis instance because the clients and the ApsaraDB for Redis instance belong to different VPCs. For example, the ECS instance on which your workloads are running is deployed in VPC A and the ApsaraDB for Redis instance is deployed in VPC B. To connect the ApsaraDB for Redis instance to the ECS instance, you can change the VPC of the ApsaraDB for Redis instance to VPC A.

Operation	Scenario
Change the vSwitch of an ApsaraDB for Redis instance	To centrally manage cloud resources and IP address whitelists, you can group the cloud resources based on workloads and then allocate IP addresses. For example, you can connect cloud resources that are related to database services, such as ECS instances and ApsaraDB for Redis instances, to the same vSwitch. Then, these cloud resources are assigned IP addresses that belong to the same CIDR block.

Impacts

- If you change the VPC or vSwitch of an instance, the instance experiences transient connections for 30 seconds. Make sure that you change the VPC or vSwitch during off-peak hours and your applications can automatically reconnect to the instance.
- If you change the VPC or vSwitch of an instance, the virtual IP address (VIP) of the instance is changed. If your applications are connected to the VIP of the instance, the connections are closed after the VIP is changed.

> **Note** After you change the VPC or vSwitch of an ApsaraDB for an instance, the endpoint of the instance, such as `r-hp3bpn39cs1vu****.redis.hangzhou.rds.aliyuncs.com`, remains unchanged. We recommend that you connect your applications to instances by using endpoints.

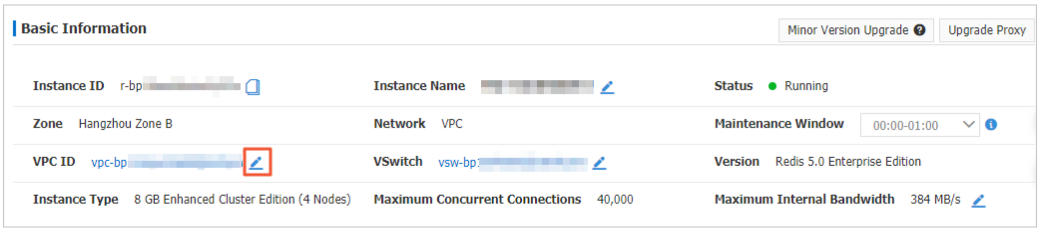
- A VIP change interrupts Data Management (DMS) services for a short period of time. For more information about DMS, see Overview. After the VIP is changed, the connections are resumed.
- After you change the VPC or vSwitch of an instance, clear the cache on clients. Otherwise, clients may be unable to write data to the instance and can only read data from the instance.

Procedure

1.

2. In the **Basic Information** section, click the ✎ icon next to the VPC ID.

Change the VPC of an instance



Basic Information Minor Version Upgrade Upgrade Proxy

Instance ID	r-bp[redacted]	Instance Name	[redacted]	Status	● Running
Zone	Hangzhou Zone B	Network	VPC	Maintenance Window	00:00-01:00
VPC ID	vpc-bp[redacted]	VSwitch	vsw-bp[redacted]	Version	Redis 5.0 Enterprise Edition
Instance Type	8 GB Enhanced Cluster Edition (4 Nodes)	Maximum Concurrent Connections	40,000	Maximum Internal Bandwidth	384 MB/s

> **Note** If you want to change only the vSwitch, you can click the ✎ icon next to the vSwitch ID.

3. In the panel that appears, select the VPC and the vSwitch that you want to use.

> **Note** If no VPC or vSwitch is available in the drop-down list, create a VPC and a vSwitch. The vSwitch and the instance must belong to the same zone. For more information, see Create and manage a VPC and Work with vSwitches.

4. Click **OK**.

> **Warning** If you change the VPC or vSwitch of an instance, the instance experiences transient connections for 30 seconds. Make sure that you change the VPC or vSwitch during off-peak hours and your applications can automatically reconnect to the instance.

5. In the message that appears, read the content and click **OK**.

Related API operations

Operation	Description
SwitchNetwork	Changes the VPC or vSwitch of an ApsaraDB for Redis instance. If the instance is deployed in the classic network, the network type of the instance changes from classic network to VPC.

5.5.2. Change the network type from classic network to VPC

You can switch the network type of an ApsaraDB for Redis instance from classic network to virtual private cloud (VPC). Then, clients in the same VPC can communicate with the ApsaraDB for Redis instance at higher security levels and lower network latency.

Prerequisites

The instance runs in the classic network.

> **Note** You can view the network type in the **Basic information** section of the instance.

Impacts

- After you switch the network type of an instance from classic network to VPC, you cannot switch back to classic network.
- The instance may experience a transient connection of a few seconds. We recommend that you perform this operation during off-peak hours and make sure that your applications can automatically reconnect to the instance.
- When you switch the network type, you can specify whether to retain the classic network endpoint of the instance. If you do not retain the classic network endpoint, the endpoint is released after the network type is switched. Then, clients cannot connect to the instance by using the classic network endpoint. In this case, you must change the database endpoint on your client at the earliest opportunity.

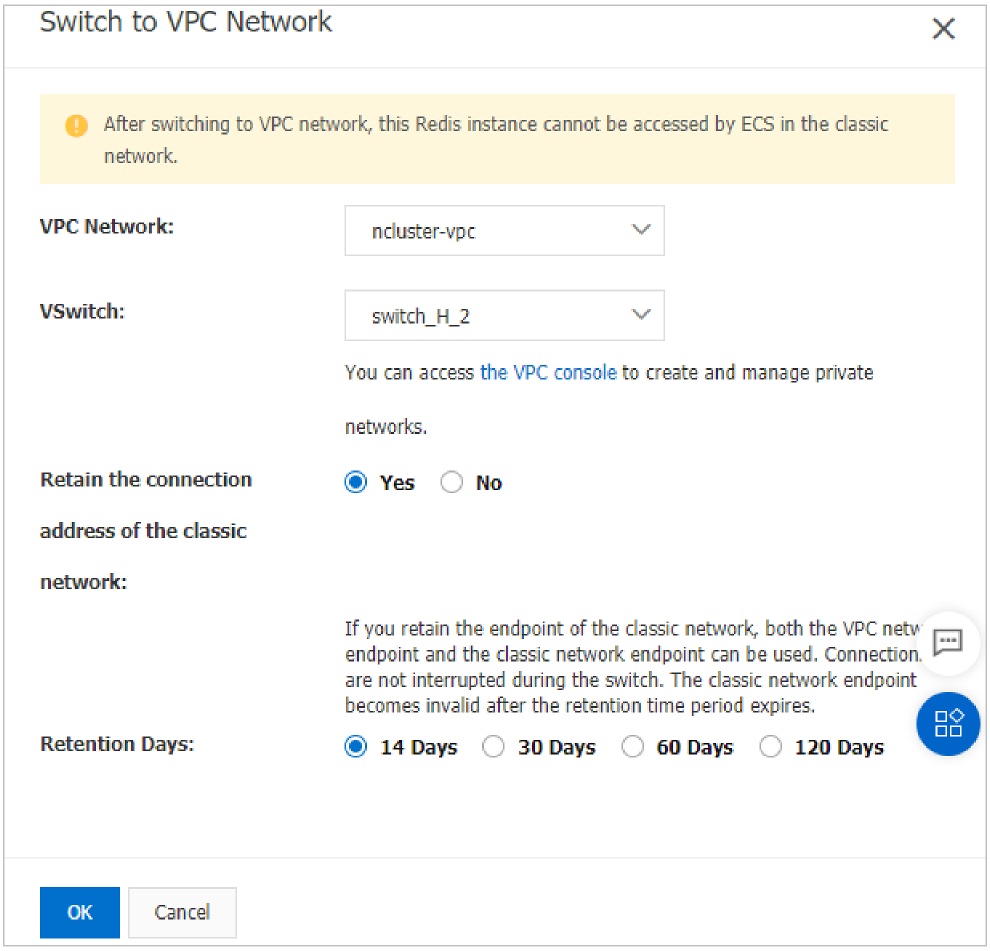
Network types of ApsaraDB for Redis instances

Network type	Description
VPC (recommended)	A VPC is a private network dedicated to your Alibaba Cloud account. VPCs are logically isolated from each other at Layer 2 to provide higher security and performance. If a client is deployed on an Elastic Compute Service (ECS) instance, you can connect the client to an ApsaraDB for Redis instance over a VPC for higher security and lower network latency. For more information about ECS instances, see What is ECS?
Classic network	Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by using security groups or whitelists.

> **Note** If a client is deployed in a VPC and an ApsaraDB for Redis instance is deployed in the classic network, the client cannot connect to the instance. To connect the client to the instance, switch the network of the instance to the client VPC.

Procedure

1.
2. On the right side of the **Connection Information** section, click **Switch to VPC**.
3. In the panel that appears on the right side of the page, set the parameters.



Switch to VPC Network

After switching to VPC network, this Redis instance cannot be accessed by ECS in the classic network.

VPC Network: ncluster-vpc

VSwitch: switch_H_2

You can access the VPC console to create and manage private networks.

Retain the connection address of the classic network: ● Yes ○ No

If you retain the endpoint of the classic network, both the VPC netw endpoint and the classic network endpoint can be used. Connection are not interrupted during the switch. The classic network endpoint becomes invalid after the retention time period expires.

Retention Days: ● 14 Days ○ 30 Days ○ 60 Days ○ 120 Days

Parameter	Description
VPC	Select the VPC and vSwitch that you want to use. **Note** If no VPC or vSwitch is available in the drop-down list, create a VPC and a vSwitch. The vSwitch and the instance must belong to the same zone. For more information, see Create and manage a VPC and Work with vSwitches.
VSwitch	
Retain the connection address of the classic network	Specify whether to retain the classic network endpoint of the ApsaraDB for Redis instance: ○ **Yes**: allows the instance to use both the classic network and VPC endpoints within a specified period of time. Clients can connect to the instance by using both endpoints. However, you must change the database endpoint on your client to the VPC endpoint before the classic network endpoint becomes invalid. ○ **No**: releases the classic network endpoint. Clients cannot connect to the instance by using the classic network endpoint.
Retention Days	Specify the retention period of the classic network endpoint. Unit: day. You can also modify the retention period after you switch the network type. For more information, see Modify the expiration date of a classic network endpoint.

4. Click **OK**.

Related API operations

Operation	Description
SwitchNetwork	Switches the network type of an ApsaraDB for Redis instance from classic network to VPC.

Related information

- Troubleshooting for connection issues in ApsaraDB for Redis
- Change the VPC or vSwitch of an ApsaraDB for Redis instance

5.5.3. Change the endpoint or port number of an ApsaraDB for Redis instance

ApsaraDB for Redis allows you to change the endpoints or port numbers of ApsaraDB for Redis instances. You can change the port number of an instance to improve security. You can also change the endpoint of a new instance to the endpoint of the original instance without the need to reconfigure your application.

Prerequisites

The instance that you want to manage is in the **Running** state.

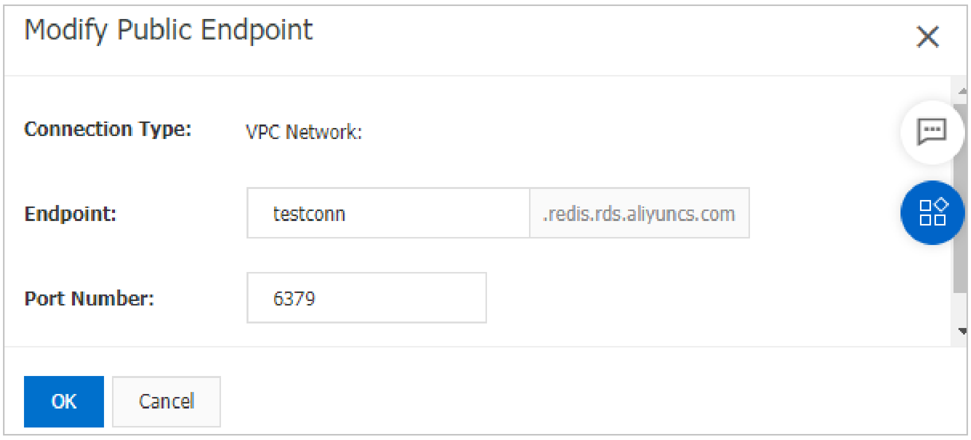
Impacts

After you change the endpoint or port number of an instance, you must update the connection information on your client. This way, the client can connect to the instance by using the new endpoint or port number.

Procedure

1.
2. In the **Connection Information** section, find the connection type that you want to manage and click **Modify Public Endpoint** in the **Actions** column.
3. In the panel that appears, enter a new endpoint and port number.

   Specify a new endpoint and port number



Modify Public Endpoint

Connection Type: VPC Network:

Endpoint: testconn .redis.rds.aliyuncs.com

Port Number: 6379

Parameter	Description
Endpoint	○ You can modify only the prefix of the endpoint. By default, the prefix is the instance ID. ○ The prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
Port	When you modify the endpoint, you can also modify the port number. Valid values for the port number: 1024 to 65535.

4. Click **OK**.

Related API operations

Operation	Description
ModifyDBInstanceConnectionString	Changes the endpoint and port number of an ApsaraDB for Redis instance.

5.5.4. Apply for a public endpoint for an ApsaraDB for Redis instance

By default, ApsaraDB for Redis provides internal endpoints. To access an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint for the instance first.

Precautions

- Public endpoints are not supported by cloud disk-based cluster instances for which the direct connection mode is enabled.
- For security concerns, if password-free access over a virtual private cloud (VPC) is enabled for an instance and you use a public endpoint to connect to the instance, you still need to enter a password.

> **Note** If a public endpoint cannot be allocated to an instance, you can update the instance to the latest minor version. For more information, see Update the minor version.

Network types of endpoints

Network type	Description
VPC	• A VPC is a private network dedicated to you on Alibaba Cloud. VPCs are logically isolated from each other to provide higher security and performance. For more information about VPCs, see What is a VPC? • By default, an ApsaraDB for Redis instance provides a VPC endpoint. You can connect to an ApsaraDB for Redis instance over a VPC to achieve higher security and performance.
Classic network	Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by using security groups or whitelists. If your cloud services are deployed in the classic network, we recommend that you change the network type to VPC. For more information, see Change the network type from classic network to VPC.

Network type	Description
Public	Security risks exist when you connect to an ApsaraDB for Redis instance over the Internet. For this reason, ApsaraDB for Redis does not provide public endpoints by default. To connect to an ApsaraDB for Redis instance over the Internet, you can apply for a public endpoint for the instance in the following scenarios: ● The device on which the client is installed, such as an Elastic Compute Service (ECS) instance, and the ApsaraDB for Redis instance are not deployed in the same VPC. For more information about ECS instances, see What is ECS? ● The device on which the client is installed and the ApsaraDB for Redis instance are not deployed in the same region. ● The client is installed on a device outside of Alibaba Cloud, such as an on-premises device. **Note** ● You are not charged for applying for public endpoints. You are also not charged for the traffic that is generated after you use public endpoints to connect to your instances. ● If you use public endpoints, data security is compromised. Proceed with caution. ● To accelerate and secure data transmission, we recommend that you migrate your applications to an ECS instance that is deployed in the same region and has the same network type as the ApsaraDB for Redis instance. This allows you to connect to the ApsaraDB for Redis instance by using an internal endpoint.

Procedure

1.
2. In the **Connection Information** section, click **Apply for Endpoint** to the right of **Public Access**.
3. (Optional) In the panel that appears, enter an endpoint and a port number.

Parameter	Description
Endpoint	○ You can modify only the prefix of the endpoint. By default, the prefix is the instance ID. ○ The prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
Port	When you modify the endpoint, you can also modify the port number. Valid values for the port number: 1024 to 65535.

4. Click **OK**.
   After the application is submitted, the public endpoint is displayed in the **Connection Information** section.

Related API operations

Operation	Description
AllocateInstancePublicConnection	Applies for a public endpoint for an ApsaraDB for Redis instance.

What's next

Use a public endpoint to connect to an ApsaraDB for Redis instance

5.5.5. Release a public endpoint for an ApsaraDB for Redis instance

You can release the public endpoints that are no longer required for ApsaraDB for Redis instances.

Prerequisites

A public endpoint is allocated to an instance. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance.

Precautions

- After the public endpoint of an instance is released, you cannot use the public endpoint to connect to the instance.
- After the public endpoint is released, we recommend that you delete the corresponding public IP address from the whitelist to ensure data security. For more information, see Step 2: Configure whitelists.

Procedure

1.
2. In the **Connection Information** section, click **Release Endpoint** next to **Public Access**.
3. In the panel that appears, click **OK**.

Related API operations

Operation	Description
ReleaseInstancePublicConnection	Releases a public endpoint for an ApsaraDB for Redis instance.

5.5.6. Enable the direct connection mode

By default, ApsaraDB for Redis local disk-based cluster instances provide proxy connection. You can also enable the direct connection mode for a local disk-based cluster instance to obtain a private endpoint. Then, clients can use the endpoint to bypass proxy nodes and connect to the instance in the same manner as they connect to a native Redis cluster.

Prerequisites

- The instance is a local disk-based cluster instance.

> **Note** The direct connection mode is not supported by cloud disk-based cluster instances that have the proxy mode enabled.

- The instance is deployed in a virtual private cloud (VPC). If the instance is deployed in the classic network, you can change the network type to VPC. For more information, see Change the network type from classic network to VPC.
- SSL encryption is disabled for the instance. For more information, see Configure SSL encryption.
- The vSwitch to which the instance is connected has sufficient IP addresses that can be allocated. For more information, see Obtain the number of available IP addresses in the vSwitch to which an ApsaraDB for Redis instance is connected.

  > **Note** For example, an ApsaraDB for Redis instance contains eight shards and you want to apply for a private endpoint for the instance. An IP address is allocated to the master node of each shard in the instance, and an additional IP address is allocated to the private endpoint. Therefore, the vSwitch must provide at least nine IP addresses. Otherwise, you cannot enable the direct connection mode for the instance.

Comparison of connection modes

- Proxy mode: By default, a cluster instance that has the proxy mode enabled provides a proxy endpoint. You can use the endpoint to connect to the cluster instance in the same manner as you connect to a standard instance. For more information, see Features of proxy nodes.
- Direct connection mode: An instance for which the direct connection mode is enabled provides a private endpoint. Clients can use the endpoint to bypass proxy nodes and access backend data nodes in the instance in the same manner as they access a native Redis cluster.

Precautions

- Connectivity performance degrades because the proxy nodes are bypassed. For cluster instances of the ApsaraDB for Redis Community Edition, the maximum number of connections for a single shard is 10,000 and the number of new connections per second is 2,000. For cluster instances of the ApsaraDB for Redis Enhanced Edition (Tair), the maximum number of connections for a single shard is 30,000 and the number of new connections per second is 2,000. For more information, see Overview.
- Data skew may take place for ApsaraDB for Redis cluster instances. Data skew occurs when one data shard in an instance receives a large number of requests while other data shards in the instance remain idle. During data skew, the maximum number of connections to a data shard may be reached and new connections to the shard may be rejected. In this case, the overall performance of the instance is affected.

  > **Note** In most cases, data skew is caused by hotkeys or large keys. For more information about how to troubleshoot hotkeys and large keys, see Use the real-time key statistics feature and Offline key analysis.

- If you cannot perform the following operations, you must release the private endpoint first. For more information, see Release a private endpoint for an ApsaraDB for Redis instance.
  - Change the configurations of an instance
  - Upgrade the major version
  - Migrate an instance across zones

Procedure

1.
2. In the **Connection Type** section, click **Apply for Endpoint** to the right of **Private Endpoint**.
3. (Optional) In the panel that appears, enter an endpoint and a port number.

Parameter	Description
Endpoint	○ You can modify only the prefix of the endpoint. By default, the prefix is the instance ID. ○ The prefix must be 8 to 64 characters in length and can contain lowercase letters and digits. It must start with a lowercase letter.
Port	When you modify the endpoint, you can also modify the port number. Valid values for the port number: 1024 to 65535.

4. Click **OK**.

FAQ

- Q: My instance meets the requirements of the prerequisites. What do I do if I am unable to find Apply for Endpoint to the right of Private Endpoint?

  A: You can update your instance to the latest minor version. For more information, see Update the minor version.

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

  A: No, the service is not interrupted when you enable the direct connection mode.

- Q: Can I enable both the direct connection mode and the proxy mode for an ApsaraDB for Redis instance at the same time?

  A: Yes, you can enable the two modes for an instance at the same time. If your instance is a local disk-based cluster instance, you can enable the two modes for the instance at the same time. If your instance is a cloud disk-based cluster instance, you can enable only a single mode for the instance.

References

- Use a private endpoint to connect to an ApsaraDB for Redis instance
- Release a public endpoint for an ApsaraDB for Redis instance
- AllocateDirectConnection
- ReleaseDirectConnection

5.5.7. Release a private endpoint for an ApsaraDB for Redis instance

If you no longer need 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, you can release the private endpoint to disable the direct connection mode before you change configurations or upgrade a major version.

Prerequisites

- The instance has been allocated a private endpoint. For more information, see Enable the direct connection mode.
- The private endpoint configured in the application is changed to another available endpoint, such as the internal endpoint in proxy mode. For more information, see View endpoints.

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

Procedure

1.
2. In the **Connection Information** section, click **Release Endpoint** next to **Direct Connection**.
3. In the panel that appears, click **OK**.

5.5.8. Obtain the number of available IP addresses in the vSwitch to which an ApsaraDB for Redis instance is connected

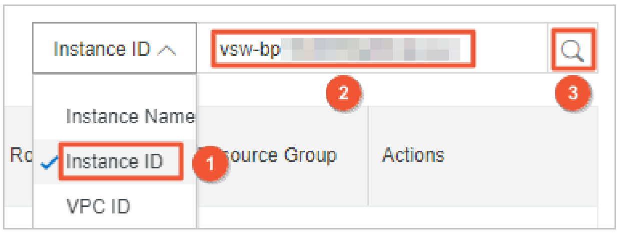
When you apply for a private endpoint, make sure that the vSwitch to which the instance is connected has sufficient IP addresses that can be allocated. This topic describes how to obtain the number of these IP addresses in the vSwitch.

Context

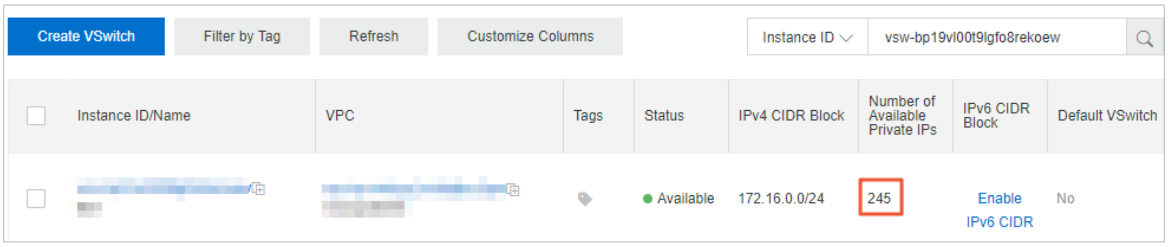
When you apply for a private endpoint, an IP address is allocated to the master node of each shard and another IP address is allocated to the private endpoint. If the vSwitch cannot provide sufficient IP addresses, the private endpoint cannot be enabled. For more information, see Enable the direct connection mode.

Procedure

1.
2. In the **Basic Information** section, copy the vSwitch ID and click the virtual private cloud (VPC) ID. You are directed to the VPC console.
3. On the details page of the VPC, click the **Resources** tab and then click the number corresponding to the vSwitch field.
4. In the upper-right corner of the page, select **Instance ID**, paste the vSwitch ID that you previously obtained, and then search for the vSwitch.



5. After you find the vSwitch, you can view the number of available IP addresses that can be allocated by the vSwitch.



□	Instance ID/Name	VPC	Tags	Status	IPv4 CIDR Block	Number of Available Private IPs	IPv6 CIDR Block	Default VSwitch
□	[redacted]	[redacted]		● Available	172.16.0.0/24	245	Enable IPv6 CIDR	No

> Note If the number of IP addresses that can be allocated does not meet your requirements, submit a ticket for technical support.

5.5.9. Modify the expiration date of a classic network endpoint

After you retain a classic network endpoint, you can extend its retention period by changing its expiration date in the ApsaraDB for Redis console.

Prerequisites

A classic network endpoint is retained after an ApsaraDB for Redis instance switches the network type from classic network to virtual private cloud (VPC). For more information, see Change the network type from classic network to VPC.

Precautions

During the period in which your instance can be connected over the classic network or a VPC, you can specify an expiration date for the classic network endpoint based on your business requirements. The new expiration date immediately takes effect. For example, if the classic network endpoint is due to expire on August 18, 2017 and you modify the expiration date to 14 days later on August 15, 2017, the classic network endpoint is released on August 29, 2017.

> Note You can modify the expiration date multiple times.

Procedure

1.

2. On the right side of the **Connection Information** section, click **Change Expiration Date** next to **Retained Connection Address of the Classic Network**.

3. In the panel that appears, set a new expiration date and click **OK**.

Related API operations

Operation	Description
ModifyInstanceNetExpireTime	Modifies the expiration date of a classic network endpoint.

5.5.10. Release a classic network endpoint

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

Prerequisites

A classic network endpoint is retained after an ApsaraDB for Redis instance switches the network type from classic network to virtual private cloud (VPC). For more information, see Change the network type from classic network to VPC.

Procedure

1.
2. In the left-side navigation pane, click **Connection**.
3. Click **Change Expiration Date** next to **Classic Network**.
4. In the dialog box that appears, select **Release Now**.

   > **Warning** A classic network endpoint immediately becomes unavailable after it is released. Before you release a classic network endpoint, change the classic network endpoint configured on your application to a VPC endpoint to ensure service availability.

5. Click **OK**.

5.6. System Parameters

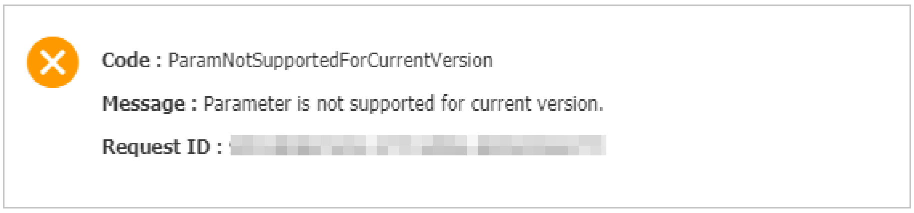
5.6.1. Modify parameters of an instance

ApsaraDB for Redis allows you to modify specific instance parameters. The instance parameters that can be modified vary based on engine versions and architectures. This topic describes how to modify instance parameters.

Precautions

Some parameters are not supported in earlier instance minor versions. If your instance is in an earlier minor version, the following error message may appear when you set the parameters. You must update your instance to the latest minor version. For more information, see Update the minor version.

Outdated minor version alert



Code : ParamNotSupportedForCurrentVersion

Message : Parameter is not supported for current version.

Request ID : [redacted]

Procedure

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the System Parameters page, find the parameter that you want to modify and click **Modify** in the **Actions** column.
4. In the dialog box that appears, modify the parameter value and click OK. For more information about the parameters and their valid values, see Supported parameters.

   > **Warning** After you submit a modification to a specific instance parameter, the instance is restarted. The instance may experience transient connections for a few seconds during the restart. Modify instance parameters with caution. Before you modify an instance parameter, check the **Restart and Take Effect** column of the parameter to make sure that the instance does not need a restart for the modification to take effect.

5. Click **OK**.

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

Related information

-
- Common parameter adjustment cases

5.6.2. Supported parameters

ApsaraDB for Redis allows you to modify specific instance parameters. This topic describes the parameters that are supported in different engine versions and architectures.

Precautions

- Specific parameters are not supported in earlier minor versions. If your instance is of an earlier minor version, an error may occur when you attempt to configure these parameters. For more information about how to update the minor version, see Update the minor version.
- After you submit the modifications for specific parameters, your instance is automatically restarted. The instance experiences transient connections that last for a few seconds during the restart. On the

page for modifying parameters in the ApsaraDB for Redis console, the **Restart and Take Effect** column indicates whether the instance must be restarted for the modification to take effect. For more information, see Modify parameters of an instance.

Supported parameters and descriptions

The following items describe the comments that are used in the tables of this topic:

- The ✔☺ symbol indicates that the major version or instance architecture supports the parameter.
- The ▯ symbol indicates that the major version or instance architecture does not support the parameter.

> **Note**
> - To ensure the stability of ApsaraDB for Redis instances, only specific parameters can be set. The parameters that are not described in this topic cannot be set.
> - For more information about instance architectures, see Standard master-replica instances, Cluster master-replica instances, and Read/write splitting instances.

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
#no_loose_audit-read-enabled	Specifies whether to enable read request audit. After you enable this feature, you can view information about read requests in the audit logs.Enable the new audit log feature Default value: no. Valid values: - *yes*: enables read request audit. - *no*: disables read request audit. **Note** ApsaraDB for Redis instances that use cloud disks do not support this parameter.	▯	▯Standard ✔☺Cluster ✔☺Read/write splitting	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
#no_loose_check-whitelist-always	Specifies whether to check that a client IP address is included in a whitelist of an ApsaraDB for Redis instance if password-free access over a virtual private cloud (VPC) is enabled. If you set this parameter to yes, the whitelist still takes effect for password-free access over a VPC. Default value: no. Valid values: • *yes*: checks whether a client IP address is included in the whitelist. • *no*: does not check whether a client IP address is included in the whitelist. **Note** ApsaraDB for Redis instances that use cloud disks do not support this parameter.	▯	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
#no_loose_disabled-commands	Specifies to disable the commands that may have high risks or high time complexity based on your business requirements, such as **FLUSHALL**, **FLUSHDB**, **KEYS**, **HGETALL**, **EVAL**, **EVALSHA**, and **SCRIPT**. Note - Specify commands in lowercase letters and separate the commands with commas (,). - Even if you disable the **FLUSHALL** command, the **Clear Data** feature in the ApsaraDB for Redis console can still be used.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺
#no_loose_sentinel-enabled	Specifies whether to enable the Sentinel-compatible mode. Default value: no. Valid values: - *yes*: enables the Sentinel-compatible mode. - *no*: disables the Sentinel-compatible mode.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯
#no_loose_ssl-enabled	Specifies whether to enable SSL encryption. Default value: no. Valid values: - *yes*: enables SSL encryption. - *no*: disables SSL encryption. Note ApsaraDB for Redis instances that use cloud disks do not support this parameter.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺
			Standard ▯	Standard ▯	Standard ▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
#no_loose_statistics-cmds		▯	Cluster ✔☺ Read/write splitting ✔☺	Cluster ✔☺ Read/write splitting ✔☺	Cluster ✔☺ Read/write splitting ✔☺
#no_loose_statistics-ip-enable	This parameter is related to the observability of ApsaraDB for Redis. After you specify this parameter for an instance, you must also enable the audit log feature for the instance. Audit logs must be collected at an interval of 5 seconds. For more information about the observability of ApsaraDB for Redis and audit logs, see Observability of ApsaraDB for Redis and Enable the new audit log feature. - #no_loose_statistics-cmds: specifies the commands whose statistics you want to collect. The source IP addresses from which the commands are issued and the frequencies at which the commands are run are collected. This parameter is empty by default, which indicates that no statistics are collected. Separate the commands with commas (,). - #no_loose_statistics-ip-enable: specifies whether to enable collection of statistics about IP addresses or whether to record IP addresses of established connections. Default value: no. Valid values: *yes* and *no*. - #no_loose_statistics-keys: specifies the keys whose statistics you want to collect. The source IP addresses from which and the frequencies at which these keys are queried or updated are collected. This parameter is empty by default, which indicates that no statistics are collected. Separate the keys with commas (,).	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis □ 6.0	Redis 5.0	Redis 4.0	Redis 2.8
#no_loose_statistics-keys	Note - To prevent performance degradation, we recommend that you do not specify an excessive number of commands for the #no_loose_statistics-cmds parameter and an excessive number of keys for the #no_loose_statistics-keys parameter. In addition, make sure that these parameters are enabled only when you want to troubleshoot issues or perform O&M tasks. - You can download audit logs in the Log Service console and use keywords described in the following section to find the logs that you want to view. For more information about how to download audit logs, see Enable the new audit log feature. ◦ A type value of 7: indicates the queries per second (QPS) of IP addresses. ◦ A type value of 8: indicates the connection statistics of IP addresses. ◦ A type value of 9: indicates the statistics of keys. ◦ A type value of 10: indicates the statistics of commands.		Standard □ Cluster ✔ Read/write splitting ✔	Standard □ Cluster ✔ Read/write splitting ✔	Standard □ Cluster ✔ Read/write splitting ✔

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
#no_loose_tls-min-version	Specifies the earliest Transport Layer Security (TLS) version supported by the instance. Default value: TLSv1. Valid values: - *TLSv1* - *Tlsv1.1* - *TLSv1.2*	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺
cluster_compat_enable	Specifies whether to enable support for the syntax of native Redis clusters. Default value: 1. Valid values: - *0*: disables support for the syntax of native Redis clusters. - *1*: enables support for the syntax of native Redis clusters.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺
max_session_processing	Specifies the maximum number of pending requests that are allowed per connection. If a proxy node forwards requests from a client to a data node but does not receive a response from the data node, these requests are pending. This parameter is used to limit the number of pending requests that are caused by capability differences between the frontend and backend of proxy nodes. This prevents an increase in memory usage.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺
ptod_enabled	Specifies whether to pass through client IP addresses to data nodes by using proxy nodes. Default value: 0. Valid values: - *0*: does not pass through client IP addresses to data nodes. - *1*: passes through client IP addresses to data nodes.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
query_cache_enabled	Specifies whether to enable the proxy query cache feature. For more information, see Use proxy query cache to address issues caused by hotkeys. Default value: 0. Valid values: - *0*: disables the proxy query cache feature. - *1*: enables the proxy query cache feature. Notice - Only performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair) support this parameter. You must update these instances and proxy nodes in these instances to the latest minor version before you can use this parameter. For more information, see Performance-enhanced instances. - The key-value pair information of the hotkeys that is cached on proxy nodes is not updated within the validity period. Therefore, make sure that your business supports eventual consistency	▯	Standard ▯ Cluster ✔☺ Read/write splitting ▯	▯	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
query_cache_expire	Specifies the validity period of cached data. Unit: milliseconds. Default value: *1000*. Valid values: *100* to *60000*. - If the cached data is modified within the validity period, the modified data is not synchronized to the cache. In this case, dirty data is returned for identical read requests until the cache expires. - You must evaluate the value of this parameter based on your business requirements and tolerance for dirty data. If the value is less than required, the cache hit rate is reduced. If the value is greater than required, the client reads dirty data for an extended period of time. ⓘ **Note** Only performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair) support this parameter. You must update these instances and proxy nodes in these instances to the latest minor version before you can use this parameter. For more information, see Performance-enhanced instances.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ▯	▯	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
query_cache_mode	Specifies the proxy query cache mode. Default value: 0. Valid values: - *0*: caches data of only the hotkeys pushed by data shards. - *1*: caches data of all keys. The cached keys are evicted based on the Least Recently Used (LRU) algorithm. Notice - Only performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair) support this parameter. You must update these instances and proxy nodes in these instances to the latest minor version before you can use this parameter. For more information, see Performance-enhanced instances. - The maximum cache capacity of each proxy node is 100 MB per thread. Therefore, if this parameter is set to *1*, the proxy nodes evict keys based on the LRU algorithm. This may reduce the cache hit rate and degrade the overall performance.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ▯	▯	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
readonly_lua_route_ronode_enable	Specifies whether to enable Lua scripting on read replicas. Default value: 0. Valid values: - *0*: disables Lua scripting. Lua scripts are processed by master nodes instead of read replicas. - *1*: enables Lua scripting. Lua scripts that include only read requests are forwarded to read replicas.	▯	Standard ▯ Cluster ▯☺ Read/write splitting ✔☺	Standard ▯ Cluster ▯☺ Read/write splitting ✔☺	▯
read_request_only_ronode_whenrwsplit_enable	Specifies whether to enable unidirectional forwarding for requests from accounts that have read-only permissions. Default value: 0. Valid values: - *0*: disables unidirectional forwarding. Requests from accounts that have read-only permissions are forwarded based on weights to all nodes including master nodes. - *1*: enables unidirectional forwarding. Requests from accounts that have read-only permissions are forwarded only to read replicas.	▯	Standard ▯ Cluster ▯☺ Read/write splitting ✔☺	Standard ▯ Cluster ▯☺ Read/write splitting ✔☺	Standard ▯ Cluster ▯☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
rt_threshold_ms	Specifies the threshold of slow logs for a proxy node. Unit: milliseconds. If the proxy node processes a request for a period of time longer than the specified threshold, the request is recorded in a slow log. **Note** This period of time starts when the proxy node sends a request to a data node and ends when the proxy node receives the response.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺
script_check_enable	Specifies whether to check that the keys used in Lua scripts are mapped to the same slot. Default value: 1. 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.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
transfer_subscribe_to_psubscrible_enable	Specifies whether to enable the feature of converting **SUBSCRIBE** to **PSUBSCRIBE**. Default value: 0. Valid values: • *0*: disables this feature. Proxy nodes do not convert SUBSCRIBE to PSUBSCRIBE. • *1*: enables this feature. Proxy nodes convert **SUBSCRIBE** to **PSUBSCRIBE**. ? **Note** If you use Pub/Sub commands in Lua scripts and the channel to which you have subscribed cannot receive messages, you can enable this feature to fix this issue.	▯	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	Standard ▯ Cluster ✔☺ Read/write splitting ✔☺	▯
appendonly	Specifies whether to enable append-only file (AOF) persistence for master nodes. Default: yes. Valid values: • *yes*: enables AOF persistence. • *no*: disables AOF persistence.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
client-output-buffer-limit pubsub	Specifies the output buffer limits of publisher and subscriber clients. The clients are disconnected when the specified limits are reached. Specify a value for this parameter in the following format: `<hard limit> <soft limit> <soft seconds>`. Default value: *33554432 8388608 60*. - `<hard limit>`: the hard limit. A client is disconnected if the output buffer of the client is larger than or equal to the hard limit value. The hard limit value is measured in bytes. - `<soft limit>`: the soft limit. `<soft seconds>`: the maximum number of seconds that the soft limit is reached or exceeded. A client is disconnected if its output buffer remains larger than or equal to the soft limit value for a period of time that is longer than or equal to the soft seconds value. The soft limit value is measured in bytes. The soft seconds value is measured in seconds.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺
dynamic-hz	Specifies whether to enable a dynamic hz value. Default value: yes. Valid values: - *yes*: enables a dynamic hz value. - *no*: disables a dynamic hz value.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ▯ Read/write splitting ▯	▯	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
hash-max-ziplist-entries	Specifies the maximum size of the keys and values of key-value pairs stored in a hash. Ziplist encoding is used only if both of the following conditions are met: • The keys and values of key-value pairs stored in the hash are all smaller than the value of the hash-max-ziplist-value parameter. The keys and values are measured in bytes. • The number of key-value pairs stored in the hash is smaller than the value of the hash-max-ziplist-entries parameter.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺
hash-max-ziplist-value	Specifies the maximum size of the keys and values of key-value pairs stored in a hash. Ziplist encoding is used only if both of the following conditions are met: • The keys and values of key-value pairs stored in the hash are all smaller than the value of the hash-max-ziplist-value parameter. The keys and values are measured in bytes. • The number of key-value pairs stored in the hash is smaller than the value of the hash-max-ziplist-entries parameter.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
hz	Specifies how frequently tasks are performed in the background. For example, you can specify how frequently expired keys are evicted. Valid values: 1 to 500. The default value is 10, which specifies that each task is performed 10 times per second. **Note** A greater value results in higher CPU consumption but allows the system to delete expired keys and close timeout connections more frequently. We recommend that you specify a value smaller than or equal to 100.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺
lazyfree-lazy-eviction	Specifies whether to enable the eviction feature based on the lazyfree mechanism. Default value: no. Valid values: • *yes*: enables the eviction feature. • *no*: disables the eviction feature.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯
lazyfree-lazy-expire	Specifies whether to delete expired keys based on the lazy free mechanism. Default value: yes. Valid values: • *yes*: deletes expired keys. • *no*: does not delete expired keys.	Standard ✔☺ Cluster ▯	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
lazyfree-lazy-server-del	Specifies whether to asynchronously delete data based on the lazy free mechanism for an implicit **DEL** operation. Default value: yes. Valid values: • *yes*: asynchronously deletes data. • *no*: does not asynchronously delete data.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯
lazyfree-lazy-user-del	Specifies whether to asynchronously delete data based on the lazyfree mechanism when a user runs the **DEL** command. Default value: yes. Valid values: • *yes*: asynchronously deletes data. • *no*: does not asynchronously delete data.	Standard ✔☺ Cluster ✔☺	▯	▯	▯
list-compress-depth	Specifies the number of nodes that are not compressed at both ends of a list. Default value: 0. Valid values: 0 to 65535. • *0*: does not compress list nodes. • A value in the range of *1* to *65535*: does not compress the specified number of nodes at both ends of a list but compresses in-between nodes.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯
list-max-ziplist-entries	Specifies the maximum size of the elements stored in a list. Ziplist encoding is used only if both of the following conditions are met: • The elements stored in the list are all smaller than the value of the list-max-ziplist-value parameter. The elements are measured in bytes. • The number of elements stored in the list is smaller than the value of the list-max-ziplist-entries parameter.	▯	▯	▯	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
list-max-ziplist-value	Specifies the maximum size of the elements stored in a list. Ziplist encoding is used only if both of the following conditions are met: • The elements stored in the list are all smaller than the value of the list-max-ziplist-value parameter. The elements are measured in bytes. • The number of elements stored in the list is smaller than the value of the list-max-ziplist-entries parameter.	▯	▯	▯	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
list-max-ziplist-size	• Specifies the maximum size of each ziplist in a quicklist. A positive number indicates 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 value indicates the maximum number of bytes in each ziplist of a quicklist. Default value: -2. Valid values: ◦ *-5*: indicates that each ziplist of a quicklist cannot exceed 64 KB. ◦ *-4*: indicates that each ziplist of a quicklist cannot exceed 32 KB. ◦ *-3*: indicates that each ziplist of a quicklist cannot exceed 16 KB. ◦ *-2*: indicates that each ziplist of a quicklist cannot exceed 8 KB. ◦ *-1*: indicates that each ziplist of a quicklist cannot exceed 4 KB.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
maxmemory-policy	Specifies the policy used to evict keys if memory is exhausted. LRU indicates least recently used. LFU indicates least frequently used. LRU, LFU, and time-to-live (TTL) policies are implemented by using approximation and randomized algorithms. Valid values: • *volatile-lru*: evicts the LRU keys among keys with a TTL. • *allkeys-lru*: evicts the LRU keys among all keys. • *volatile-lfu*: evicts LFU keys among keys with a TTL. • *allkeys-lfu*: evicts LFU keys among all keys. • *volatile-random*: randomly evicts keys among keys with a TTL. • *allkeys-random*: randomly evicts keys among all keys. • *volatile-ttl*: evicts the key that has the shortest TTL among keys with a TTL. • *noeviction*: does not evict keys, but returns error messages for write operations.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
notify-keyspace-events	Specifies the event types of which the server can notify clients. The value of this parameter is a combination of the following characters: • *K*: keyspace events. All events are published with a `__keyspace@<db>__` prefix. • *E*: keyevent events. All events are published with a `__keyevent@<db>__` prefix. • *g*: generic events that are not related to specific commands, such as **DEL**, **EXPIRE**, and **RENAME**. • *$*: events of string commands. • *l*: events of list commands. • *s*: events of set commands. • *h*: events of hash commands. • *z*: events of sorted set commands. • *x*: expiration events. An expiration event is triggered when an expired key is deleted. • *e*: eviction events. An eviction event is triggered when a key is evicted based on maxmemory policies. • *A*: the alias for the g$lshzxe parameter. ? **Note** The specified value must include at least *K* or *E*. Otherwise, no events are triggered.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
set-max-intset-entries	Specifies the maximum number of data entries for which a set supports intset encoding. A set uses intset encoding when the following conditions are met: - All data entries in the set are strings. - The set contains only radix-10 integers in the range of 64-bit signed integers.	Standard ✔ Cluster ✔	Standard ✔ Cluster ✔ Read/write splitting ✔	Standard ✔ Cluster ✔ Read/write splitting ✔	Standard ✔ Cluster ✔ Read/write splitting ✔
slowlog-log-slower-than	Specifies the threshold to log slow queries. When an operation is executed for a period of time that exceeds the specified threshold, the operation is logged. Unit: microseconds. Valid values: *10000* to *10000000*. Default value: *20000*.	Standard ✔ Cluster ✔	Standard ✔ Cluster ✔ Read/write splitting ✔	Standard ✔ Cluster ✔ Read/write splitting ✔	Standard ✔ Cluster ✔ Read/write splitting ✔
slowlog-max-len	Specifies the maximum number of entries that can be stored in the slow log. Valid values: *100* to *10000*. Default value: *1024*.	Standard ✔ Cluster ✔	Standard ✔ Cluster ✔ Read/write splitting ✔	Standard ✔ Cluster ✔ Read/write splitting ✔	Standard ✔ Cluster ✔ Read/write splitting ✔
stream-node-max-bytes	Specifies the maximum amount of memory that each macro node in a stream can consume. Valid values: *0* to *999999999999999*. Unit: bytes. ? **Note** A value of *0* indicates no limit.	Standard ✔ Cluster ✔	Standard ✔ Cluster ▯ Read/write splitting ▯	▯	▯
stream-node-max-entries	Specifies the maximum number of entries stored on each macro node in a stream. Valid values: *0* to *999999999999999*. ? **Note** A value of *0* indicates no limit.	Standard ✔ Cluster ✔	Standard ✔ Cluster ▯ Read/write splitting ▯	▯	▯

Parameter	Description	Major version and instance architecture			
		Redis 6.0	Redis 5.0	Redis 4.0	Redis 2.8
timeout	Specifies a timeout period. The system closes a connection to a client if the connection remains idle for the specified period of time. Valid values: *0* to *100000*. Unit: seconds. **Note** A value of *0* indicates that no timeout periods are specified.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	▯
zset-max-ziplist-entries	Specify the maximum number of key-value pairs stored in a sorted set and the maximum size of the keys and values of key-value pairs stored in the sorted set. Ziplist encoding is used only if the following conditions are met: • The keys and values of key-value pairs stored in the sorted set are all smaller than the value of the zset-max-ziplist-value parameter. The keys and values are measured in bytes. • The number of key-value pairs stored in the sorted set is smaller than the value of the zset-max-ziplist-entries parameter.	Standard ✔☺ Cluster ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺	Standard ✔☺ Cluster ✔☺ Read/write splitting ✔☺
zset-max-ziplist-value					

5.6.3. Common Parameter Adjustment Cases

5.6.3.1. Disable AOF persistence

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

Overview

ApsaraDB for Redis provides two data persistence options: AOF persistence and Redis Database (RDB) persistence. AOF persistence logs every write operation received by the server, such as **SET** . When you restart an ApsaraDB for Redis instance, the service reruns the commands in the AOF files to restore data. If AOF files are larger than required, open source Redis runs an AOF rewrite task to recreate AOF files and minimize the size of AOF files.

You can specify the AOF_FSYNC_EVERYSEC policy to enable AOF persistence in ApsaraDB for Redis. After you specify this policy, the system records all write commands in an AOF file every second and saves the AOF file to disks. The policy has a negligible impact on the performance and can minimize data loss caused by accidental operations. ApsaraDB for Redis allows you to archive incremental backups based on AOF files and ensures service performance when the system runs the AOF rewrite task.

AOF persistence may affect write performance. If an ApsaraDB for Redis instance is used in a cache-only scenario, you can perform the operations described 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 appendonly parameter to no, the following impacts exist:
    - The system disables AOF persistence without the need to restart the instance.
    - After AOF persistence is disabled, AOF files cannot be used to restore data.
    - The existing AOF logs remain unaffected.
    - For a standard instance, AOF persistence is disabled for the master node and the replica node is not affected.
    - For a cluster instance, AOF persistence is disabled for the master nodes of all data shards and the replica nodes are not affected.
    - For a read/write splitting instance, AOF persistence is disabled for the master node and all read replicas and the replica node is not affected.
- If you change the appendonly value to yes, the system enables AOF persistence without the need to restart the instance.

Disable AOF persistence in the ApsaraDB for Redis console

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the System Parameters page, click **Modify** in the **Actions** column corresponding to the appendonly parameter.
4. In the dialog box that appears, perform the following steps:

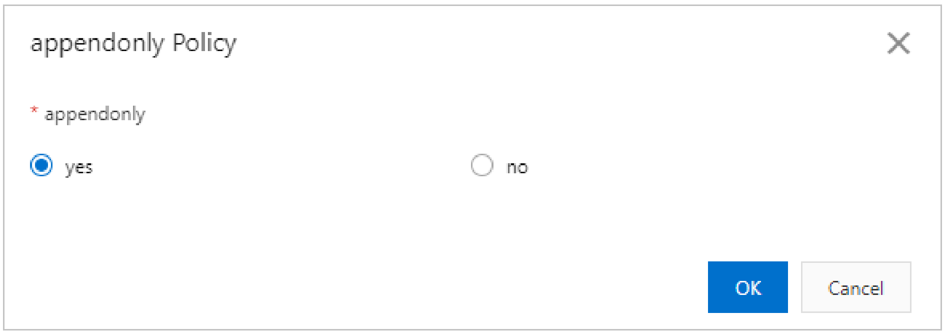
    > **Warning** After you disable AOF persistence, you can no longer use AOF files to restore data. You can only use RDB files to restore data. Proceed with caution. For more information about RDB files, see Backup and restoration solutions.

    i. Set the appendonly parameter.

    Valid values:

    - yes: enables AOF persistence.
    - no: disables AOF persistence.

ii. Click **OK**.



appendonly Policy

* appendonly

● yes ○ no

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.2. 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 an excessive amount of 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 ApsaraDB for Redis processes the commands from clients, ApsaraDB for Redis temporarily stores command output data in output buffers and then sends the data to the clients. If you do not limit the data size 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 failure. 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.

You can set the client-output-buffer-limit pubsub parameter to a proper value to prevent the output buffers of Pub/Sub clients from consuming an excessive amount of memory.

Options

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

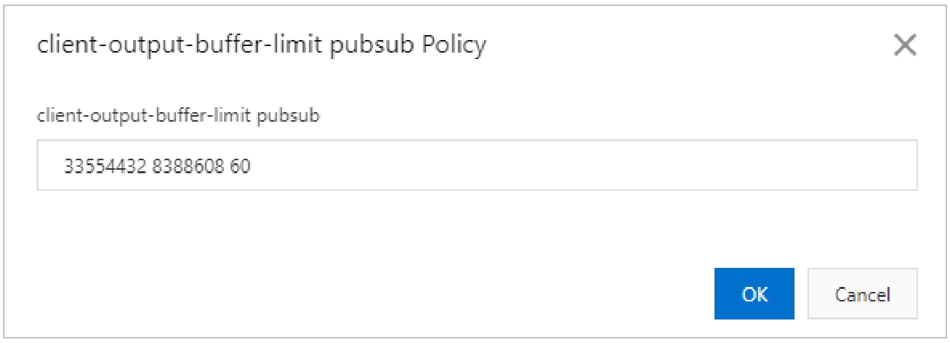
- `hard limit` specifies a fixed limit. Unit: bytes. If the output buffer of a Pub/Sub client reaches or exceeds the `hard limit` value, the client is immediately disconnected.
- `soft limit` specifies a limit that depends on the time. Unit: bytes. `soft seconds` specifies the

amount of time to continuously trigger a soft limit. Unit: seconds. If the output buffer of a Pub/Sub client reaches or exceeds the `soft limit` value for a period of time in seconds that is specified by the `soft seconds` parameter, the client is disconnected.

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

Specify the parameter in the ApsaraDB for Redis console

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the page that appears, find the client-output-buffer-limit pubsub parameter and click **Modify** in the **Actions** column.
4. In the dialog box that appears, perform the following steps:
   i. Specify the client-output-buffer-limit pubsub parameter based on the description in the Options section of this topic.
   ii. Click **OK**.



client-output-buffer-limit pubsub Policy

client-output-buffer-limit pubsub

33554432 8388608 60

> **Note** After you complete the settings, you can view these settings in the ApsaraDB for Redis console. You can also use redis-cli to connect to the ApsaraDB for Redis instance and run the CONFIG GET **client-output-buffer-limit** command to view the settings. For more information about redis-cli, see Use redis-cli to connect to an ApsaraDB for Redis instance.

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.3. Change the frequency of background tasks

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

Relationship between background tasks and the hz parameter

To periodically 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. The following section lists some background tasks:

- Calculate the least recently used (LRU) information and delete expired keys.
- Close client connections that have timed out.
- Manage hash tables.
- Perform Redis Database (RDB) or Append Only File (AOF) persistence operations.
- Update statistics.

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

Scenarios

ApsaraDB for Redis runs background tasks to delete expired keys. The following section describes this process:

1. ApsaraDB for Redis randomly selects 20 keys whose time to live (TTL) is specified and checks whether these keys are expired.
2. ApsaraDB for Redis identifies expired keys and deletes them.
3. If more than 25% of the selected keys are expired, ApsaraDB for Redis runs the background task again.

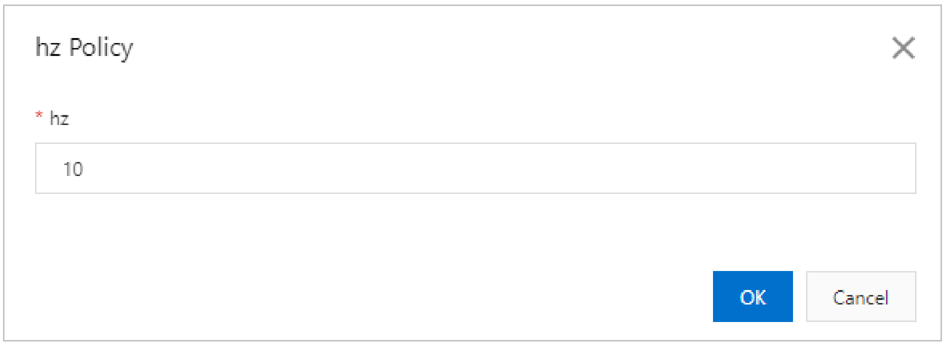
If a large number of expired keys or a sharp increase in expired keys exists and ApsaraDB for Redis does not frequently delete expired keys, the remaining expired keys occupy a large amount of memory and may affect the performance of ApsaraDB for Redis. To resolve this issue, you can increase the value of the hz parameter to allow background tasks to be executed more frequently.

Valid values and suggested settings for the hz parameter

Valid values of the hz parameter are 1 to 500. If you increase the value of the hz parameter, background tasks are executed more frequently but the CPU utilization of ApsaraDB for Redis also increases. You can use the default value 10 in most cases. If you want to run specific 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 CPU utilization.

Modify parameters in the ApsaraDB for Redis console

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the page that appears, find the hz parameter and click **Modify** in the **Action** column.
4. In the dialog box that appears, perform the following steps:
    i. Change the value of the hz parameter based on your business requirements.
    ii. Click **OK**.



Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.4. Enable dynamic frequency control for background tasks

The dynamic-hz parameter is a new parameter added to 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 or later.
- The ApsaraDB for Redis instance uses the standard master-replica architecture.

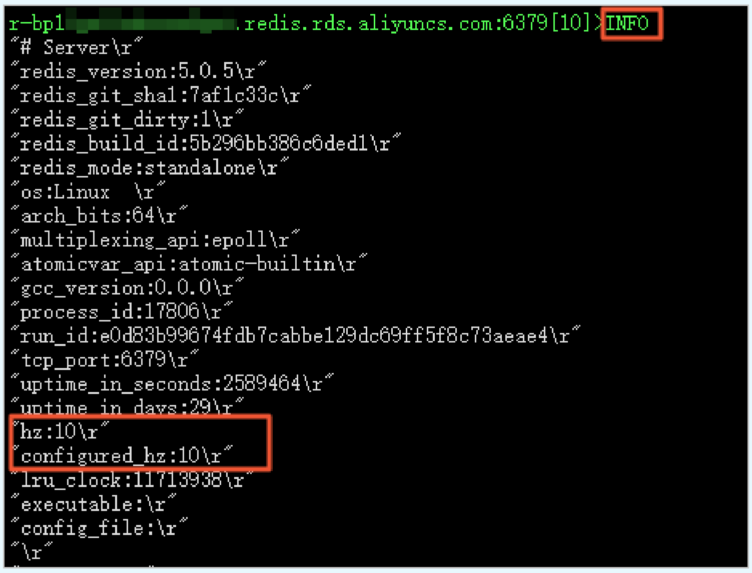
Relationship between the hz and dynamic-hz parameters

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 at which background tasks are performed 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 lower than required, resources cannot be timely recycled when a large number of client connections have timed out or when a large number of expired keys exist. This may lead to poor performance or even crashes of ApsaraDB for Redis.
- If the frequency is higher than required, background tasks consume a large amount of CPU resources. The performance of ApsaraDB for Redis may also deteriorate.

To balance CPU utilization and task efficiency, 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.

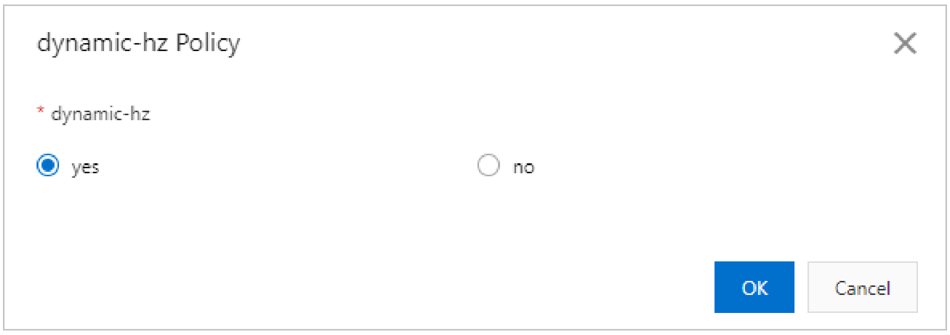


```
r-bp1[redacted].redis.rds.aliyuncs.com:6379[10]>INFO
"# Server\r"
"redis_version:5.0.5\r"
"redis_git_sha1:7af1c33c\r"
"redis_git_dirty:1\r"
"redis_build_id:5b296bb386c6ded1\r"
"redis_mode:standalone\r"
"os:Linux  \r"
"arch_bits:64\r"
"multiplexing_api:epoll\r"
"atomicvar_api:atomic-builtin\r"
"gcc_version:0.0.0\r"
"process_id:17806\r"
"run_id:e0d83b99674fdb7cabbe129dc69ff5f8c73aeae4\r"
"tcp_port:6379\r"
"uptime_in_seconds:2589464\r"
"uptime_in_days:29\r"
"hz:10\r"
"configured_hz:10\r"
"lru_clock:11713938\r"
"executable:\r"
"config_file:\r"
"\r"
```

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. The default value is `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 changes the hz value based on the number of client connections. The hz value increases with the number of client connections. Accordingly, background tasks are performed more frequently.

Modify parameters in the ApsaraDB for Redis console

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the page that appears, find the dynamic-hz parameter and click **Modify** in the Actions column.
4. In the dialog box that appears, perform the following steps:
   i. Specify the dynamic-hz parameter.
   ii. Click **OK**.



dynamic-hz Policy

* dynamic-hz

● yes ○ no

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.5. Customize the size of macro nodes in streams

You can modify the stream-node-max-bytes parameter to specify the maximum amount of 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 or later.
- The ApsaraDB for Redis instance uses the standard master-replica architecture.

Relationship between Redis streams and macro nodes

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

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

- stream-node-max-entries: The default value is 100, which indicates that each macro node can store up to 100 stream entries. Valid values: 0 to 999,999,999,999,999. A value of 0 indicates that no limit exists. 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 indicates that each macro node can consume up to 4,096 bytes of memory. Valid values: 0 to 999,999,999,999,999. A value of 0 indicates that no limit exists.

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 permanently store messages, you can use the *MAXLEN* parameter to specify the maximum number of messages that can be stored 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 of value5001 to field1 of cap
pedstream and set the maximum number of messages to 5,000.
```

When the number of messages in a stream reaches the upper limit, the earliest message is deleted each time a new message is added. This way, the maximum length of the stream remains unchanged regardless of how many messages are added to the stream. In addition, the memory consumed by deleted messages is released.

> **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 immediately released. ApsaraDB for Redis deletes a macro node and releases the consumed memory only when all messages on the macro node are marked as deleted.

If you set the maximum queue length to an exact value such as 5,000 messages, performance is significantly degraded. To optimize memory usage, a Redis stream stores data on delta-compressed macro nodes that are connected in 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 frequently added and deleted. The performance of ApsaraDB for Redis degrades if it frequently deletes messages. Therefore, we recommend that you add a tilde (~) in the XADD command to specify an approximate maximum length. Example:

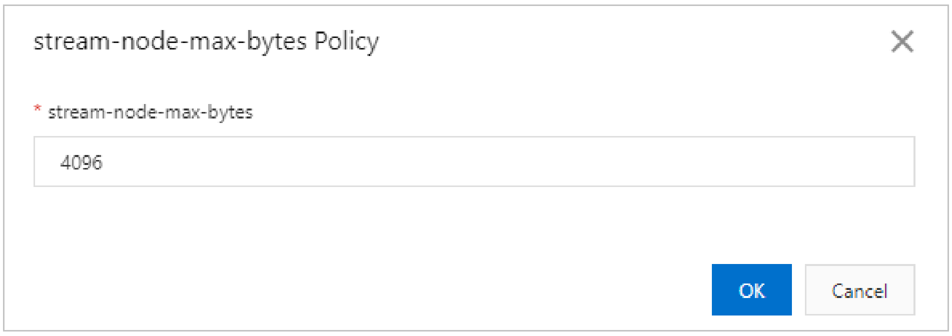
```
XADD cappedstream MAXLEN ~ 5000 * field value1 //Add a value of value5001 to field1 of capp
edstream and set the maximum number of messages to approximately 5,000.
```

This way, 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 on 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.

Procedure

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the page that appears, find the stream-node-max-bytes parameter and click **Modify** in the **Actions** column.
4. In the dialog box that appears, perform the following steps:
    i. Change the value of the stream-node-max-bytes parameter based on your business requirements.
    ii. Click **OK**.



stream-node-max-bytes Policy

* stream-node-max-bytes

4096

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.6. 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 architecture is used, and the engine version of the instance is Redis 4.0 or later.

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

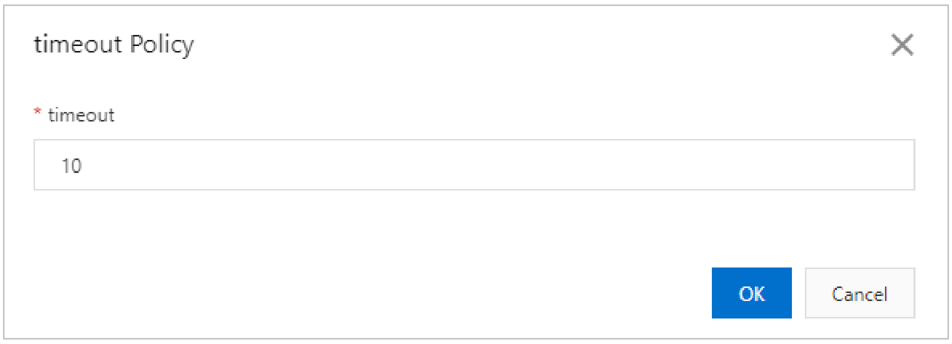
Manage client connections in ApsaraDB for Redis

In common scenarios, you can use clients to manage connections. For example, you can use clients to 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 the 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 filled 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 of this parameter is 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 value. For example, if the timeout parameter is set to 10 seconds, a client connection may remain idle for more than 10 seconds and be closed when many client connections have been established on the server. To reduce the latency, you can specify a larger value for the hz parameter to increase the frequency of the background task that closes idle connections.

Specify a timeout period

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the System Parameters page, find the timeout parameter and click **Modify** in the Actions column.
4. In the dialog box that appears, change the value of the timeout parameter.



timeout Policy

* timeout

10

5. Click **OK**.

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.7. Enable the Sentinel-compatible mode

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

Prerequisites

-
-

Overview of Redis Sentinel

Redis Sentinel provides open source Redis with features such as master and replica monitoring, fault alerting, and automatic failover. Redis Sentinel is suitable for many business scenarios that use self-managed Redis databases and require high reliability. To facilitate the migration of these self-managed Redis databases to the cloud, Alibaba Cloud provides the Sentinel-compatible mode.

Note ApsaraDB for Redis uses the HA component developed by Alibaba Cloud without the need to use Redis Sentinel. For more information about the HA component, see Feature.

After you enable the Sentinel-compatible mode, you can run the Sentinel commands described in the following table.

Command	Description
SENTINEL sentinels	Queries Sentinel instances of a specified master and the status of these Sentinel instances. The following syntax is used: `SENTINEL sentinels <Name of a master>`
SENTINEL get-master-addr-by-name	Queries the IP address and port number of a specified master. The following syntax is used: `SENTINEL get-master-addr-by-name <Name of a master>`

Procedure

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the System Parameters page, find the #no_loose_sentinel-enabled parameter and click **Modify** in the **Actions** column.

   > **Note** If a Redis 4.0 instance does not support this parameter, update the minor version of the instance. For more information about, see Update the minor version.

4. In the dialog box that appears, select **yes** and click **OK**.

   For more information about the parameters, see Modify the parameters of an ApsaraDB for Redis instance.

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

Related information

- Use the Sentinel-compatible mode to connect to an ApsaraDB for Redis instance

5.6.3.8. Disable high-risk commands

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

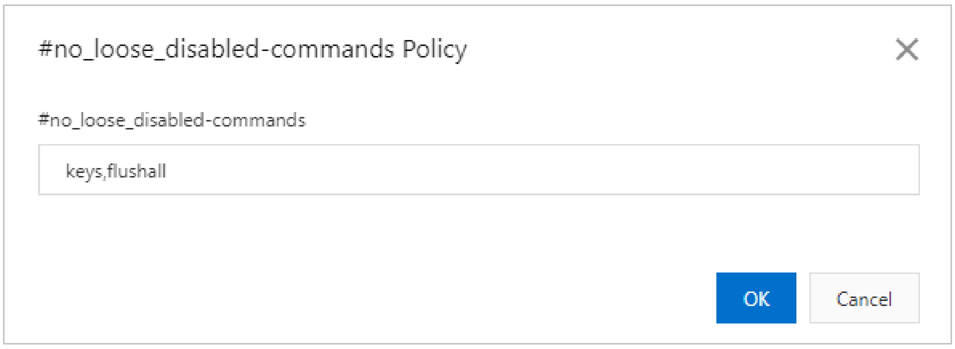
Background information

In some scenarios, unlimited use of commands may cause some issues. Some Redis commands can delete a large volume of data or all data from a database, such as **flushall** and **flushdb**. Improper uses of some commands such as **keys** and **hgetall** may cause blocking in the single-threading Redis model and degrade service performance.

To ensure stable and efficient management, you can disable specific commands to minimize risks for your workloads.

Procedure

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the System Parameters page, find the #no_loose_disabled-commands parameter and click **Modify** in the **Actions** column.
4. In the dialog box that appears, specify the commands that you want to disable.



#no_loose_disabled-commands Policy

#no_loose_disabled-commands

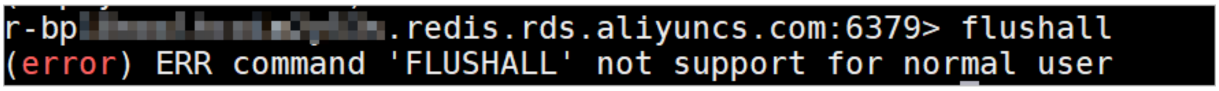
keys,flushall

> **Note** The commands that you specify can contain only lowercase letters. Separate multiple commands with commas (,). Example: *keys,flushall*.

5. Click **OK**.

Results

If you use redis-cli to connect to an instance and run the disabled **FLUSHALL** command, ApsaraDB for Redis returns the following error: `(error) ERR command 'FLUSHALL' not support for normal user`.



```
r-bp[redacted].redis.rds.aliyuncs.com:6379> flushall
(error) ERR command 'FLUSHALL' not support for normal user
```

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.6.3.9. Use a whitelist in password-free access mode

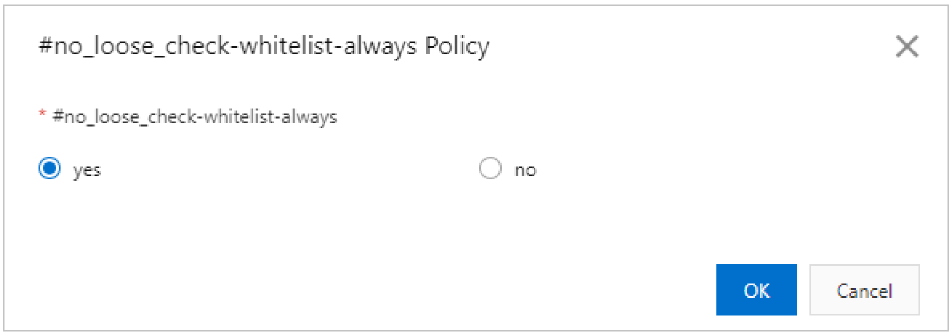
This topic describes how to modify the value of the #no_loose_check-whitelist-always parameter to use a whitelist even when the password-free access feature is enabled.

Background information

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

Procedure

1.
2. In the left-side navigation pane, click **System Parameters**.
3. On the System Parameters page, find the #no_loose_check-whitelist-always parameter and click **Modify** in the **Actions** column.
4. In the dialog box that appears, select **yes** and click **OK**.



#no_loose_check-whitelist-always Policy

* #no_loose_check-whitelist-always

● yes ○ no

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

Related API operations

Operation	Description
DescribeParameters	Queries the configuration and operational parameters of an ApsaraDB for Redis instance.
ModifyInstanceConfig	Modifies parameters of an ApsaraDB for Redis instance.

5.7. High availability

5.7.1. Causes and impacts of master-replica switchovers

ApsaraDB for Redis can monitor the health states of nodes. If a master node in an instance fails, ApsaraDB for Redis automatically triggers a master-replica switchover. For example, the roles of master and replica nodes are switched over to ensure the high availability (HA) of the instance. ApsaraDB for Redis allows a master-replica switchover to be manually triggered. This feature can be applied to disaster recovery drills and access to nearby nodes that are deployed in different zones.

Causes

- Manual switchover

  A master-replica switchover is manually performed by you or an authorized Alibaba Cloud technical expert. For more information, see Manually switch workloads from a master node to a replica node.

- Risk mitigation

  Alibaba Cloud detects vulnerabilities in an ApsaraDB for Redis instance. These vulnerabilities may cause the ApsaraDB for Redis instance to run not as expected. In this case, ApsaraDB for Redis fixes the vulnerabilities and performs a master-replica switchover during the specified maintenance window. High-risk vulnerability fixes are automatically performed at the earliest opportunities and master-replica switchovers are triggered.

  You can find the events that were triggered under the preceding conditions in history events. For more information, see Query history events. You can also manage pending events of master-replica switchovers. For more information, see Query and manage pending events.

- Instance failure

  Alibaba Cloud detects failures in an ApsaraDB for Redis instance. These failures cause the ApsaraDB for Redis instance to run not as expected. In this case, ApsaraDB for Redis performs a master-replica switchover to switch your workloads over to replica nodes. This minimizes the impacts of the failures.

  You are notified of such events with internal messages in the following format:

  [Alibaba Cloud] Dear ******: Your ApsaraDB for Redis instance r-bp1zxszhcgatnx**** (name: ****) has an error. A switchover is triggered to ensure that your instance runs as expected. We recommend that you check whether your application is still connected to your instance and configure your application to automatically reconnect to the instance.

Impacts

Cause	Impact	Related suggestion
Manual switchover	- The data nodes on which the switchover is performed are disconnected for a few seconds. A switchover has potential data loss risks. For example, the data may become inconsistent between the master and replica nodes due to synchronization latency. To prevent potential data loss risks caused by the switchover and data doublewrite caused by the Domain Name System (DNS) cache, the data nodes become read-only for up to 30 seconds. - After an instance enters the **Switching** state, you cannot manage this instance. For example, you cannot modify the instance configurations or migrate the instance to another zone.	Make sure that your applications are configured to automatically reconnect to the instance or handle exceptions. Otherwise, one of the following error messages may be returned during a switchover: `READONLY You can't write against a read only instance` and
Risk mitigation		

Cause	Impact	Related suggestion
Instance failure	- The data nodes on which the switchover is performed are disconnected for a few seconds. - After an instance enters the **Switching** state, you cannot manage this instance. For example, you cannot modify the instance configurations or migrate the instance to another zone.	only instance and closed. You can't write or read against a disable instance .

> Note After the master-replica switchover is complete, the state of the instance becomes **Running**.

FAQ

- Q: What is the principle behind the master-replica switchover triggered by an instance failure?

  A: The HA system relies on its detection mechanism to detect failures. The following table describes the HA mechanism.

Event	Description
Health check	The HA system checks whether master and replica nodes are healthy.
Master node failure	i. When a master node is determined to be unavailable, a replica node acts as the master node. At the same time, the virtual IP address (VIP) of the master node is switched to the replica node. ii. Another replica node is created to ensure data synchronization.
Replica node failure	When a replica node is determined to be unavailable, another replica node is created to ensure data synchronization and maintain the data persistence of the master-replica architecture.

  > Note Some data that was recently written to a master node may be lost because the synchronization between the master and replica nodes is asynchronously implemented.

- Q: Does a master-replica switchover affect the use of read replicas in read/write splitting instances? For more information about read/write splitting instances, see Read/write splitting instances.

  A: A master-replica switchover does not affect the use of read replicas in read/write splitting instances.

- Q: Does a master-replica switchover triggered for a specific data shard in an instance affect the instance as a whole if the instance is a cluster master-replica instance? For more information about cluster master-replica instances, see Cluster master-replica instances.

  A: The instance as a whole is not affected. Only the data shard is affected. For more information, see Impacts.

5.7.2. Manually switch workloads from a master node to a replica node

This topic describes how to switch workloads from a master node to a replica node. In addition to automatic switchovers, ApsaraDB for Redis allows you to manually switch workloads from a master node to a replica node. Manual switchovers can help you perform disaster recovery drills and verify the error processing capabilities of clients. If your ApsaraDB for Redis instance is deployed in multiple zones, you can also perform a manual switchover to allow applications to connect to the nearest node.

Prerequisites

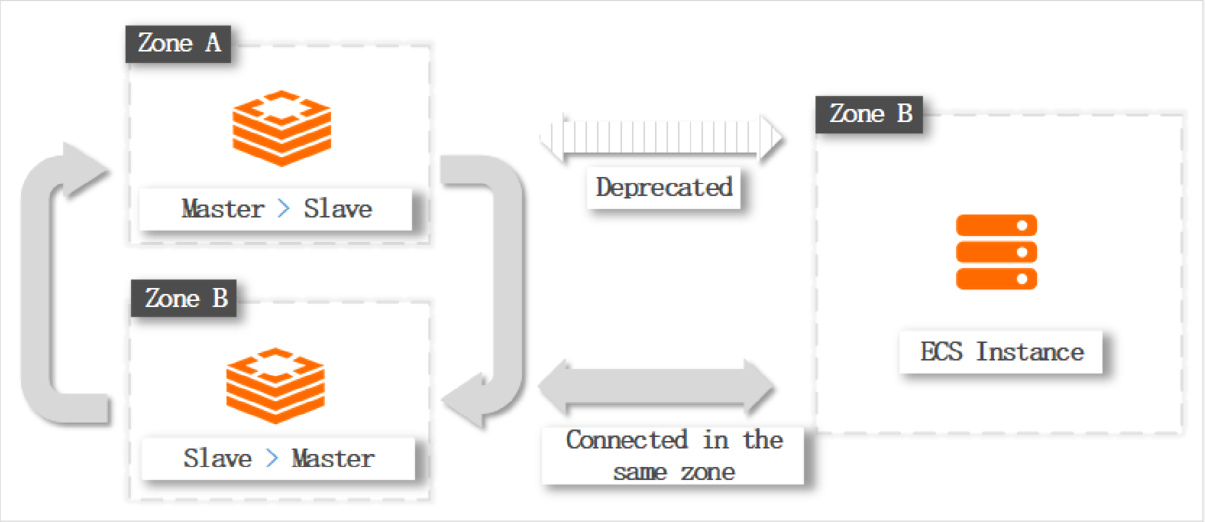
The instance is an ApsaraDB for Redis Community Edition instance or a performance-enhanced or hybrid-storage instance of ApsaraDB for Redis Enhanced Edition (Tair). For more information, see Performance-enhanced instances and Hybrid-storage instances (phased out).

Impacts

- The data nodes on which the switchover is performed are disconnected for a few seconds. A switchover has potential data loss risks. For example, the data may become inconsistent between the master and replica nodes due to synchronization latency. To prevent potential data loss risks caused by the switchover and data doublewrite caused by the Domain Name System (DNS) cache, the data nodes become read-only for up to 30 seconds.
- After an instance enters the **Switching** state, you cannot manage this instance. For example, you cannot modify the instance configurations or migrate the instance to another zone.

Scenarios

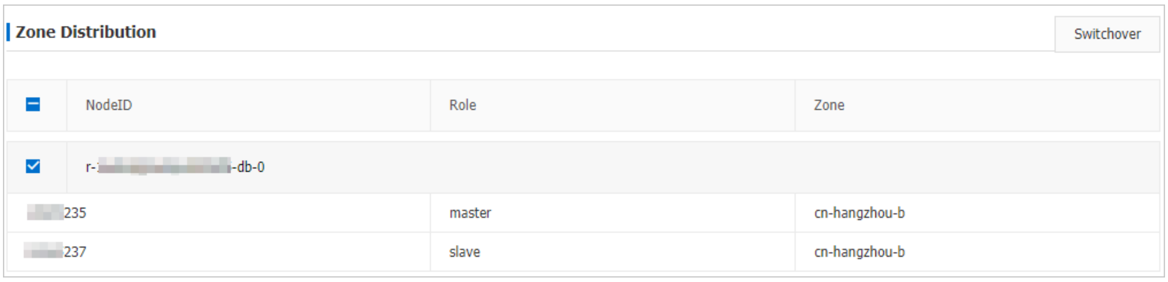
In the following example, the Elastic Compute Service (ECS) instance on which your applications are deployed resides in Zone B, and the master node of the ApsaraDB for Redis instance resides in Zone A. The connection between the ECS instance and the master node of the ApsaraDB for Redis instance spans different zones. This increases network latency. This also affects the performance of the ApsaraDB for Redis instance and your business.



To optimize the deployment architecture of cloud resources, you can switch your workloads from the master node to the replica node to minimize network latency. After the manual switchover, the original replica node is promoted to become the new master node. The manual switchover does not cause changes to the zones and IDs of the master node and replica node.

Procedure

1.
2. In the left-side navigation pane, click **Service Availability**.
3. In the **Zone Distribution** section, select the data shard for which you want to perform a switchover, and click **Switchover**.



Zone Distribution

Switchover

☐	NodeID	Role	Zone
☑	r-[redacted]-db-0		
	[redacted]235	master	cn-hangzhou-b
	[redacted]237	slave	cn-hangzhou-b

> **Note** If the ApsaraDB for Redis instance is a cluster instance, you can view the information about the zones to which the master node and replica node of each data shard belong. For more information, see Master-replica cluster instances.

4. In the panel that appears, select the time when you want to perform the switchover.
   - **Immediate**: performs the switchover immediately.
   - **Maintenance Window**: performs the switchover within the specified maintenance window. For more information about how to view and modify the maintenance window of an ApsaraDB for Redis instance, see Set a maintenance window.
5. Click **OK**.

Related API operations

Operation	Description
SwitchInstanceHA	Manually switches workloads from a master node to a replica node. This operation can be used for disaster recovery drills. This operation can also be used for nearby connections between your applications and an ApsaraDB for Redis instance if the instance is deployed in multiple zones.

References

ApsaraDB for Redis can monitor the health status of nodes. If a master node in an instance fails, ApsaraDB for Redis automatically triggers a master-replica switchover. For example, the roles of master and replica nodes are switched over to ensure the high availability (HA) of the instance. For more information, see Causes and impacts of master-replica switchovers.

5.7.3. Restart or rebuild a proxy node

ApsaraDB for Redis allows you to restart or rebuild a proxy node. This way, you can perform real-time disaster recovery drills. You can also perform O&M tasks when the service is unavailable or experiences a high latency.

Prerequisites

The ApsaraDB for Redis instance to which the proxy node belong must meet the following requirements:

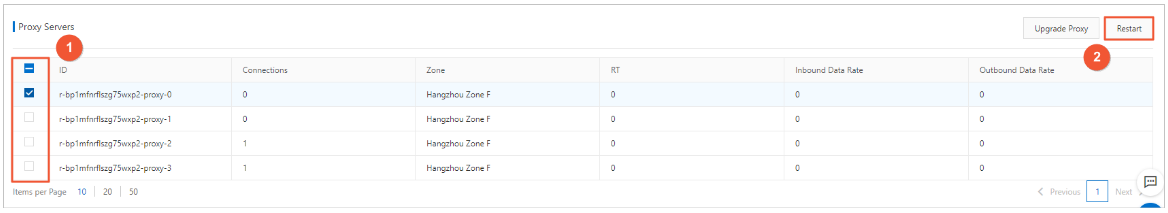
- A cluster instance or a read/write splitting instance is used. For more information, see Cluster master-replica instances or Read/write splitting instances.
- The instance is an ApsaraDB for Redis Community Edition instance or a performance-enhanced or hybrid-storage instance of ApsaraDB for Redis Enhanced Edition (Tair). For more information, see Performance-enhanced instances and Hybrid-storage instances (phased out).

Impacts

Proxy restart and rebuild may result in transient connections, which affect existing connections. Make sure that your applications can automatically reconnect to the proxy node. We recommend that you restart or rebuild a proxy node during off-peak hours.

Procedure

1.
2. In the left-side navigation pane, click **Service Availability**.
3. In the **Proxy Servers** section, select the proxy node that you want to manage and click **Restart**.



Proxy Servers

☐	ID	Connections	Zone	RT	Inbound Data Rate	Outbound Data Rate
☑	r-bp1mfnrflszg75wxp2-proxy-0	0	Hangzhou Zone F	0	0	0
☐	r-bp1mfnrflszg75wxp2-proxy-1	0	Hangzhou Zone F	0	0	0
☐	r-bp1mfnrflszg75wxp2-proxy-2	1	Hangzhou Zone F	0	0	0
☐	r-bp1mfnrflszg75wxp2-proxy-3	1	Hangzhou Zone F	0	0	0

4. In the panel that appears, select Restart Mode.
   - **In-place Restart**: restarts the proxy node.
   - **Rebuild Proxy**: rebuilds the proxy node. If an issue cannot be fixed after the restart, select this mode.
5. Click **OK**.

5.7.4. Upgrade proxy nodes

ApsaraDB for Redis supports manual hot upgrades of proxy nodes. After you upgrade proxy nodes, you can use the latest features. For example, you can view the audit logs of proxy nodes. This topic describes how to upgrade proxy nodes.

Prerequisites

The ApsaraDB for Redis instance to which the proxy nodes belong must meet the following requirements:

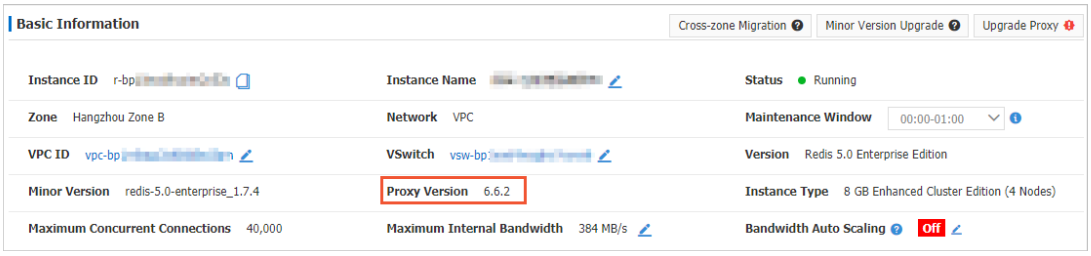
- A cluster instance or a read/write splitting instance is used. For more information, see Cluster master-

- A cluster instance or a read/write splitting instance is used. For more information, see Cluster master-replica instances or Read/write splitting instances.
- The instance is an ApsaraDB for Redis Community Edition instance or a performance-enhanced or hybrid-storage instance of ApsaraDB for Redis Enhanced Edition (Tair). For more information, see Performance-enhanced instances and Hybrid-storage instances (phased out).

View the minor version of proxy nodes

You can view the minor version of proxy nodes in the ApsaraDB for Redis console. For more information, see Update the minor version.

View the minor version of proxy nodes



Basic Information

Instance ID	r-bp[redacted]	Instance Name	[redacted]	Status	● Running
Zone	Hangzhou Zone B	Network	VPC	Maintenance Window	00:00-01:00
VPC ID	vpc-bp[redacted]	VSwitch	vsw-bp[redacted]	Version	Redis 5.0 Enterprise Edition
Minor Version	redis-5.0-enterprise_1.7.4	Proxy Version	6.6.2	Instance Type	8 GB Enhanced Cluster Edition (4 Nodes)
Maximum Concurrent Connections	40,000	Maximum Internal Bandwidth	384 MB/s	Bandwidth Auto Scaling	Off

Note For more information about the detailed release notes of minor versions, see ApsaraDB for Redis proxy nodes.

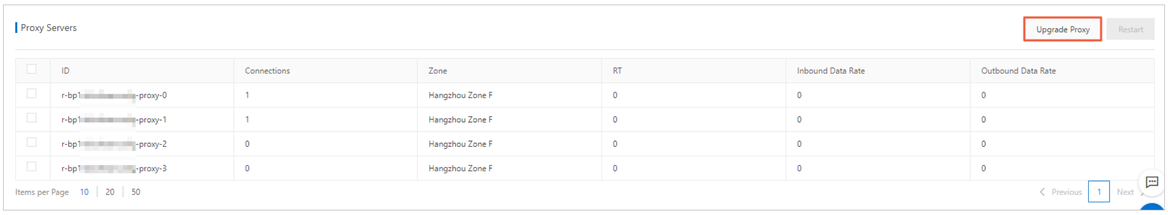
Impacts

- Proxy nodes support hot upgrades. Proxy nodes of the new version can restore a connection based on the client connection information of proxy nodes of the earlier version. This ensures that upgrades do not interrupt services. However, a millisecond-level latency jitter may occur during the upgrades.
- The hot upgrades are valid only for normal connections. The execution of the block, transaction, Pub, and Sub commands is interrupted during hot upgrades. Make sure that these commands support the reconnection mechanism.
- If a Redis client uses a private endpoint to connect to the ApsaraDB for Redis instance, no commands are affected by a proxy upgrade.

Note For more information about proxy endpoints and private endpoints, see Proxy mode and Direct connection mode.

Procedure

1.
2. In the left-side navigation pane, click **Service Availability**.
3. In the **Proxy Servers** section, click **Upgrade Proxy**.



Proxy Servers

□	ID	Connections	Zone	RT	Inbound Data Rate	Outbound Data Rate
□	r-bp1[redacted]-proxy-0	1	Hangzhou Zone F	0	0	0
□	r-bp1[redacted]-proxy-1	1	Hangzhou Zone F	0	0	0
□	r-bp1[redacted]-proxy-2	0	Hangzhou Zone F	0	0	0
□	r-bp1[redacted]-proxy-3	0	Hangzhou Zone F	0	0	0

> **Note** All proxy nodes are upgraded at the same time. You cannot separately upgrade specified proxy nodes.

4. In the panel that appears, select the time when you want the upgrade to take effect.
   - **Immediate**: immediately upgrades proxy nodes.
   - **Maintenance Window**: upgrades proxy nodes within the specified maintenance window. For more information about how to modify the maintenance window of an instance, see Set a maintenance window.
5. Click **OK**.

5.8. Tag management

5.8.1. Create tags

If you have a large number of instances, you can create and add multiple tags to classify and filter instances by tag.

Precautions

- A tag consists of a key-value pair. Each key must be unique for an Alibaba Cloud account in a region. This constraint does not apply to key values.

  > **Note** A key can have zero to multiple values.

- You can edit tags for a maximum of 50 instances at a time.
- You can add up to 20 tags to each instance.
- You can add or remove up to 20 tags at a time.

Procedure

1.
2. Perform one of the following steps to create a tag for one or more instances:
   - Create tags for a single instance

     Find the instance for which you want to add tags. In the **Tag** column corresponding to the instance, click **Edit Tags**.
   - Create tags for multiple instances

     Select the instances and click **Edit Tags** below the instance list.
3. In the dialog box that appears, click **Create**.

   > **Note** If you have created tags, click **Available Tags** to add the tags to one or more instances.

4. Set the key and value of a tag and then click **Confirm**.
5. Repeat Step 4 and Step 5 to create all the tags. Then click **OK** in the lower-right corner of the dialog box.

> Note After you create tags for an instance, you can add the tags to other instances.

Related API operations

Operation	Description
TagResources	Adds tags to one or more ApsaraDB for Redis instances.

What's next

- Filter ApsaraDB for Redis instances by tag
- View the tags that are added to an ApsaraDB for Redis instance

5.8.2. Filter ApsaraDB for Redis instances by tag

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

Procedure

1.
2. On the **Instances** page, select the tag key and tag value, and click **Search**.

> Note
>
> - After you create a tag or update an existing tag, you must refresh the page to view the updated the tag list.
> - To clear a filter condition, click the ⊗ icon next to the selected tag.

Related API operations

Operation	Description
ListTagResources	Queries the ApsaraDB for Redis instances to which specified tags are added or the tags added to specified ApsaraDB for Redis instances.

5.8.3. View the tags that are added to an ApsaraDB for Redis instance

You can view the tags that are added to an ApsaraDB for Redis instance on the Instances page.

Procedure

1.
2. On the **Instances** page, find the instance and view its tags in the **Tag** column of the instance.

Related API operations

Operation	Description
ListTagResources	Queries the ApsaraDB for Redis instances to which specified tags are added or the tags added to specified ApsaraDB for Redis instances.

5.8.4. Remove or delete tags

When a tag is no longer needed by an ApsaraDB for Redis instance, you can remove the tag from the instance. If the tag is not added to an instance, it is deleted.

Precautions

- You can remove up to 20 tags at a time.
- If you remove a tag from all instances, the tag is automatically deleted.
- If you remove a tag from an instance, the normal operation of the instance is not affect. After all tags of an instance are removed, the instance cannot be filtered by tag.

Procedure

1.
2. In the **Tag** column corresponding to the instance for which you want to remove tag, choose ... > **Edit** .
3. In the dialog box that appears, click the ⊗ icon next to the tag that you want to remove.

   > **Note** To delete a tag, remove the tag from all instances.

4. Click **OK**.

Related API operations

Operation	Description
UntagResources	Removes tags from an ApsaraDB for Redis instance.

5.9. Set a maintenance window

You can maintain an ApsaraDB for Redis instance during off-peak hours by modifying the default maintenance window.

Context

- Alibaba Cloud maintains ApsaraDB for Redis instances and servers to ensure that they run stably. Maintenance for instances and servers occurs at irregular intervals. To ensure that the maintenance is successful, instances enter the **Maintaining Instance** state before the maintenance window runs on the day of maintenance. When an instance is in this state, you can still access data in the database.

You are temporarily not allowed to make instance changes in the ApsaraDB for Redis console. For example, you cannot change the configurations of the instance. However, you can perform queries. For example, you can query performance monitoring data.

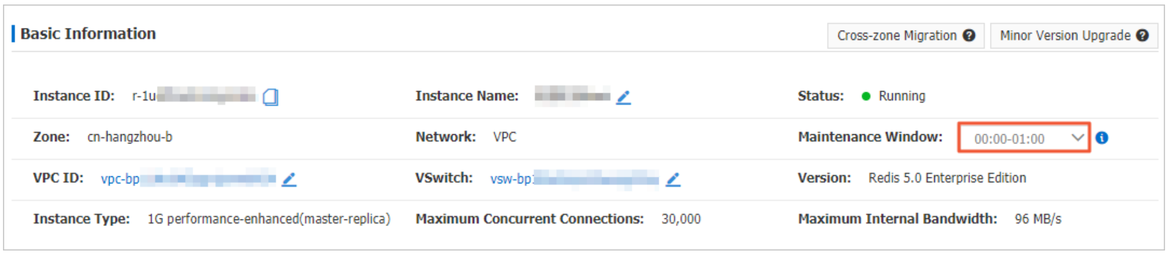
> **Note** Before the maintenance window runs, ApsaraDB for Redis sends SMS messages and emails to the contacts that are associated with your Alibaba Cloud account.

- Instances may be temporarily disconnected when you make changes to the instances. For example, an instance is disconnected when you change the configurations of the instance. To minimize the impacts on your business, we recommend that you perform these operations during the maintenance window.

> **Notice** After the maintenance window runs, the instance is temporarily disconnected. We recommend that you set the maintenance window to a time period that is during off-peak hours.

Procedure

1.
2. In the **Basic Information** section, select a time range from the **Maintenance Window** drop-down list.



Basic Information

Instance ID: r-1u[redacted]	**Instance Name:** [redacted]	**Status:** ● Running
Zone: cn-hangzhou-b	**Network:** VPC	**Maintenance Window:** 00:00-01:00
VPC ID: vpc-bp[redacted]	**VSwitch:** vsw-bp[redacted]	**Version:** Redis 5.0 Enterprise Edition
Instance Type: 1G performance-enhanced(master-replica)	**Maximum Concurrent Connections:** 30,000	**Maximum Internal Bandwidth:** 96 MB/s

3. Select a proper time range.

Related API operations

Operation	Description
ModifyInstanceMaintainTime	Modifies the maintenance window of an ApsaraDB for Redis instance. Alibaba Cloud maintains ApsaraDB for Redis instances during the specified maintenance window.

5.10. Migrate an instance across zones

If the current zone in which your instance is deployed has insufficient resources for a specification upgrade or you want to improve the disaster recovery capability, you can migrate the instance to another zone.

Prerequisites

The ApsaraDB for Redis instance has a classic network endpoint or a virtual private cloud (VPC) endpoint, and does not have the following endpoints:

Notice If the instance has the endpoints that are listed in the following table, release the endpoints before you migrate the instance. Otherwise, the **Cross-zone Migration** button is dimmed and the operation cannot be performed.

Endpoint	Method
Classic network endpoint that is retained when the network type is changed to VPC	Release a classic network endpoint
Public endpoint	Release a public endpoint for an ApsaraDB for Redis instance
Private endpoint	Release a private endpoint for an ApsaraDB for Redis instance

Precautions

- When you migrate an instance across zones, the instance may experience transient connections for a few seconds. Make sure that your application is configured with a reconnection mechanism.
- The time that is required for the migration varies based on factors such as the network conditions, task queue status, and data volume. We recommend that you migrate the instance during off-peak hours.
- If you migrate an instance across zones, the virtual IP address (VIP) of the instance such as 172.16.88.60 is changed. However, the endpoint of the instance remains unchanged. We recommend that you connect to the instance by using the endpoint. If you use a VIP to connect to the instance, the connection fails.
- If an instance is deployed in a VPC, you cannot change the VPC when you migrate the instance across zones.
- To ensure better performance and stability, if the minor version of an instance is outdated, the system updates the minor version of the instance to the latest version during the migration.

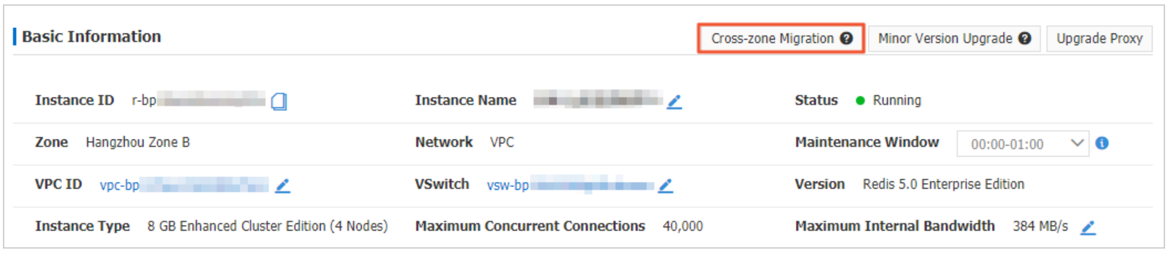
Supported migration types and scenarios

Type	Scenario
Migrate an instance from one zone to another zone	The ApsaraDB for Redis instance is migrated to the zone where an Elastic Compute Service (ECS) instance is deployed. Then, the ECS instance can connect to the ApsaraDB for Redis instance over the internal network with lower network latency.
Migrate an instance from multiple zones to multiple different zones	
Migrate an instance from one zone to multiple zones	You want to implement disaster recovery across data centers for the ApsaraDB for Redis instance. An ApsaraDB for Redis instance that is deployed in a single zone can tolerate server- and rack-level faults. An ApsaraDB for Redis instance that is deployed in multiple zones can tolerate data center-level faults.

Type	Scenario
Migrate an instance from multiple zones to one zone	You want to migrate the ApsaraDB for Redis instance based on your business requirements.

Procedure

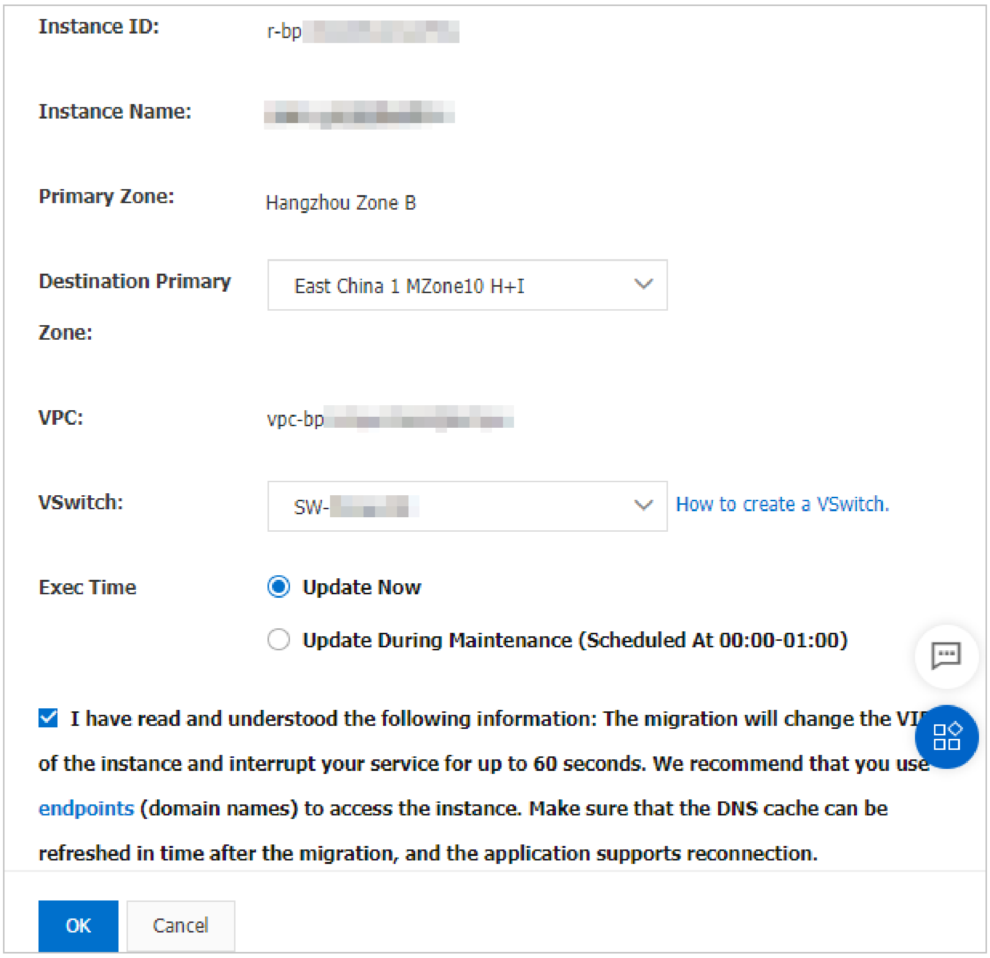
1.
2. In the **Basic Information** section, click **Cross-zone Migration**.



Basic Information

Instance ID	r-bp	Instance Name		Status	Running
Zone	Hangzhou Zone B	Network	VPC	Maintenance Window	00:00-01:00
VPC ID	vpc-bp	VSwitch	vsw-bp	Version	Redis 5.0 Enterprise Edition
Instance Type	8 GB Enhanced Cluster Edition (4 Nodes)	Maximum Concurrent Connections	40,000	Maximum Internal Bandwidth	384 MB/s

3. In the panel that appears, set the following parameters.



Instance ID:	r-bp
Instance Name:	
Primary Zone:	Hangzhou Zone B
Destination Primary Zone:	East China 1 MZone10 H+I
VPC:	vpc-bp
VSwitch:	sw-
Exec Time	● Update Now ○ Update During Maintenance (Scheduled At 00:00-01:00)

☑ I have read and understood the following information: The migration will change the VIP of the instance and interrupt your service for up to 60 seconds. We recommend that you use endpoints (domain names) to access the instance. Make sure that the DNS cache can be refreshed in time after the migration, and the application supports reconnection.

Parameter	Description
Destination Primary Zone	Select the destination zone.

Parameter	Description
VSwitch	Select the destination vSwitch. If no vSwitch exists in the destination zone, you must create one. For more information, see Work with vSwitches. **Note** This parameter appears only when the instance runs in a VPC.
Exec Time	○ **Update Now**: After you click **OK**, the migration immediately starts. When the instance status changes to **Running**, the instance is migrated. ○ **Update During Maintenance**: After you click **OK**, the system immediately performs tasks to prepare for the migration and changes the instance to the **Migrating to Another Zone** state. The instance will not be migrated to another zone until the specified maintenance window starts. For more information about how to modify the maintenance window, see Set a maintenance window.

4. Select the check box for the dialog box and click **OK**.

Related API operations

Operation	Description
MigrateToOtherZone	Migrates an ApsaraDB for Redis instance across zones in the same region.

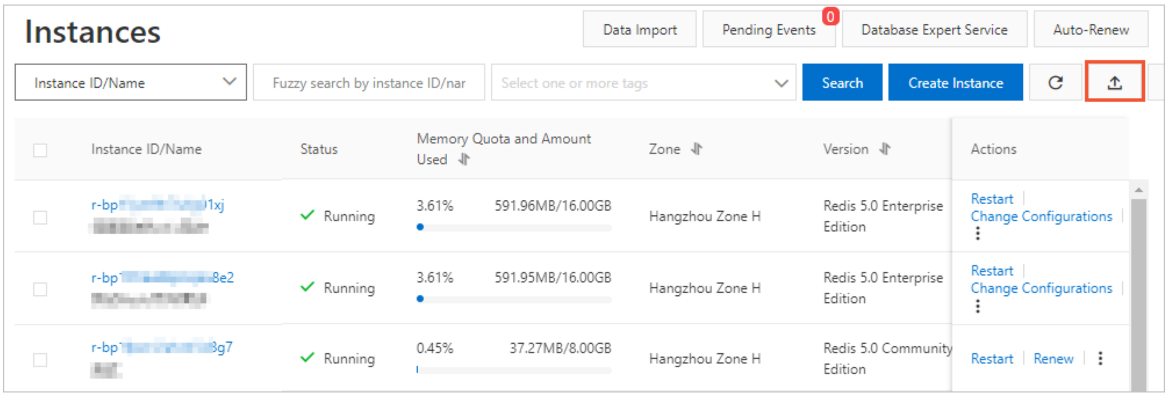
5.11. Export the instance list

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

Procedure

1.

2. On the **Instances** page, click the ⭱ icon.



□	Instance ID/Name	Status	Memory Quota and Amount Used		Zone	Version	Actions
□	r-bp[redacted]1xj [redacted]	✓ Running	3.61%	591.96MB/16.00GB	Hangzhou Zone H	Redis 5.0 Enterprise Edition	Restart | Change Configurations | ⋮
□	r-bp[redacted]8e2 [redacted]	✓ Running	3.61%	591.95MB/16.00GB	Hangzhou Zone H	Redis 5.0 Enterprise Edition	Restart | Change Configurations | ⋮
□	r-bp[redacted]3g7 [redacted]	✓ Running	0.45%	37.27MB/8.00GB	Hangzhou Zone H	Redis 5.0 Community Edition	Restart | Renew | ⋮

3. In the dialog box that appears, select required parameters.
4. Click **OK**.
   The instance information is exported to a CSV file that is automatically downloaded. You can use a text editor to view this file.

6.Security management

6.1. Create and manage database accounts

ApsaraDB for Redis allows you to create multiple database accounts for an instance. You can grant these accounts different permissions, such as the read-only, read/write, and replication permissions. This helps you flexibly manage instances and minimize user errors for data security.

Prerequisites

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

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

Limits

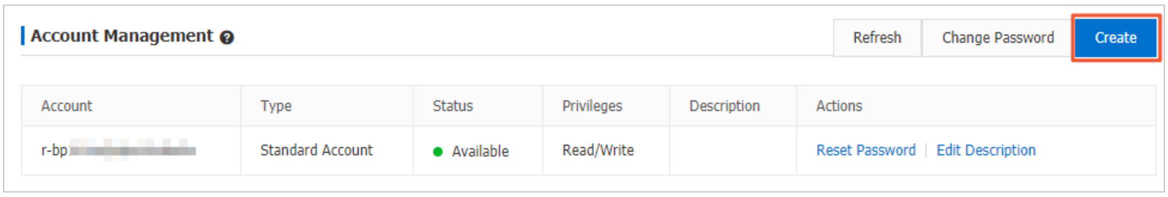
You can create a maximum of 20 accounts for an ApsaraDB for Redis instance.

Procedure

1.
2. In the left-side navigation pane, click **Account Management**.

   > **Note** If your instance runs Redis 4.0 or later and does not support the **Account Management** feature, you can update the minor version of the instance. For more information, see Update the minor version.

3. Click **Create** in the upper-right corner of the page.



Account	Type	Status	Privileges	Description	Actions
r-bp[redacted]	Standard Account	● Available	Read/Write		Reset Password | Edit Description

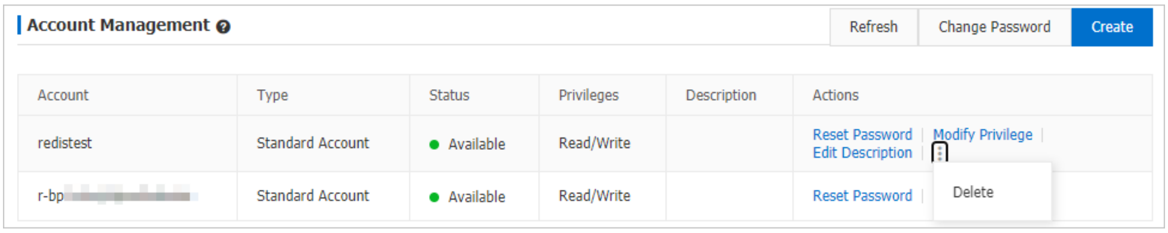
4. In the panel that appears, configure the parameters that are described in the following table.

Parameter	Description
Account	The account name. It must meet the following requirements: ○ The name can contain lowercase letters, digits, and hyphens (-) and must start with a lowercase letter. ○ The name can be up to 35 characters in length. ○ The name cannot be one of the reserved words in the Reserved words for account names section.

Parameter	Description
Privilege	The permissions that are granted to the account. Valid values: ◦ **Read-only**: The account has only permissions to read data and is not allowed to modify data. ◦ **Read/Write**: The account has permissions to read and write data. ◦ **Copy**: The account has 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** permissions.
Password Settings	The password of the account. It 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 must be 8 to 32 characters in length.
Confirm Password	Enter the password again.
Description (optional)	The description of the account. It must meet the following requirements: ◦ The description must start with a letter and cannot start with *http://* or *https://*. ◦ The description can contain letters, digits, underscores (_), and hyphens (-). ◦ The description must be 2 to 256 characters in length.

5. Click **OK**.
   The new account is in the **Creating** state. After about 1 minute, the state of the account changes to **Available**.

6. (Optional) Perform the following operations based on your business requirements to manage the account:



Account	Type	Status	Privileges	Description	Actions
redistest	Standard Account	● Available	Read/Write		Reset Password | Modify Privilege | Edit Description | ⋮
r-bp[redacted]	Standard Account	● Available	Read/Write		Reset Password |

- Reset the password

  Find the account and click **Reset Password** in the **Actions** column. In the panel that appears, reset the password and click **OK**.

- Modify permissions

  Find the account and click **Modify Privilege** in the **Actions** column. In the panel that appears, select the required permissions and click **OK**.

- Modify the description

  Find the account and click **Edit Description** in the **Actions** column. In the panel that appears, modify the description and click **OK**.

- Delete an account

  Find the account and choose ⋮ > **Delete** in the **Actions** column. In the panel that appears, click **OK**.

FAQ

Q: Why does an account already exist after an instance is created?

A: A default account whose name is the instance ID is automatically created after an instance is created to ensure data security. The password of this account has been specified when you create the instance. The password can be reset if you forget it.

Reserved words for account names

When you create an account, the account name cannot be one of the following reserved words. The reserved words are separated by 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 API operations

Operation	Description
CreateAccount	Creates an account that has the specified permissions for an ApsaraDB for Redis instance.
GrantAccountPrivilege	Modifies the permissions of an account for an ApsaraDB for Redis instance.
ModifyAccountDescription	Modifies the description of an account for an ApsaraDB for Redis instance.
ModifyAccountPassword	Changes the password of a specified account for an ApsaraDB for Redis instance.
DeleteAccount	Deletes a specified account for an ApsaraDB for Redis instance.

Related information

- Use redis-cli to connect to an ApsaraDB for Redis instance
- CreateAccount
- DescribeAccounts
- ModifyAccountDescription
- Reset a password
- GrantAccountPrivilege
- DeleteAccount

6.2. Change or reset the password

If you forget your password, want to change your password, or have not set a password for an ApsaraDB for Redis instance, you can set a new password for the instance.

Precautions

If your instance runs Redis 4.0 or later and does not support the **Account Management** feature, you must update the minor version of the instance. For more information, see Update the minor version.

Procedure

1.
2. In the left-side navigation pane, click **Account Management**.
3. In the upper-right corner, click **Change Password**.

   > **Note** If you forget the current password, you can also find the account and click **Reset Password** in the **Actions** column to reset your password.

4. In the dialog box that appears, select the account for which you want to change the password and enter the current password and a new password.

> Note
>
> - The password must be 8 to 32 characters in length.
> - The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include `! @ # $ % ^ & * ( ) _ + - =`

5. Click **OK**.

Related API operations

Operation	Description
ResetAccountPassword	Resets the password of an account for an ApsaraDB for Redis instance.

What's next

After you reset the password, you must replace the password in the Redis client with the new password.

6.3. Configure whitelists

By default, ApsaraDB for Redis instances block access from all IP addresses to ensure the security and stability of databases. Before you use an ApsaraDB for Redis instance, you must add IP addresses or CIDR blocks that are used to access the ApsaraDB for Redis instance to a whitelist of the instance. Whitelists can be used to improve the access security of ApsaraDB for Redis instances. We recommend that you maintain whitelists on a regular basis.

Prerequisites

The ApsaraDB for Redis instance is updated to the latest minor version. For more information about how to update the minor version, see Update the minor version.

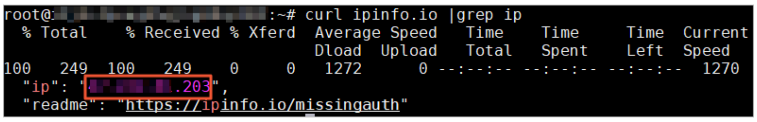
> Note If the **Minor Version Update** button on the **Instance Information** page is dimmed or if a message indicating that the current version is the latest version appears after you click this button, your instance is of the latest minor version.

Preparations

Before you configure a whitelist for an ApsaraDB for Redis instance, you must obtain the IP addresses of clients based on the client installation locations.

Client installation location	Network type	How to obtain the IP address of a client

Client installation location	Network type	How to obtain the IP address of a client
ECS instance (recommended)	VPC	How do I query the IP addresses of ECS instances? Note - Make sure that the ECS instance and the ApsaraDB for Redis instance are deployed in the same VPC. The basic information sections of the instances must display the same VPC ID. If the instances are deployed in different VPCs, you can change the VPC to which the ECS instance belongs. For more information, see Change the VPC of an ECS instance. - The network types of the ECS instance and the ApsaraDB for Redis instance may be different. For example, the ECS instance belongs to the classic network and the ApsaraDB for Redis instance belongs to a VPC. For more information about how to connect to an ApsaraDB for Redis instance from an ECS instance when the instances are deployed in different types of networks, see Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks.

On-premises device or third-party cloud	Internet	Select one of the following methods based on the operating system of the on-premises device:
		- Linux operating system: Run the **curl ipinfo.io grep ip** command on the on-premises device to obtain the public IP address. The following figure shows the sample result.
		
		`root@[redacted]:~# curl ipinfo.io |grep ip`
		`  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current`
		`                                 Dload  Upload   Total   Spent    Left  Speed`
		`100   249  100   249    0     0   1272      0 --:--:-- --:--:-- --:--:--  1270`
		`  "ip": "[redacted].203",`
		`  "readme": "https://ipinfo.io/missingauth"`
		- Windows operating system: Visit ipinfo to obtain the public IP address of the on-premises device.

Methods of configuring a whitelist

Method	Description
Method 1: Manually add a whitelist	Manually add the IP address of a client to a whitelist of the ApsaraDB for Redis instance to allow the client to access the instance.

Method	Description
Method 2: Add ECS security groups as whitelists	A security group is a virtual firewall that is used to control the inbound and outbound traffic of ECS instances in the security group. For more information, see Overview. To authorize multiple ECS instances to access an ApsaraDB for Redis instance, you can associate the ApsaraDB for Redis instance with the security group of these ECS instances. This method is more convenient than manually adding the IP addresses of these ECS instances to an instance whitelist. **Note** The engine version of the ApsaraDB for Redis instance must be Redis 4.0 or later. For more information about how to upgrade the engine version, see Upgrade the major version.

> **Note** You can set IP address whitelists and specify ECS security groups as whitelists for an ApsaraDB for Redis instance. Both IP addresses in the IP address whitelists and ECS instances in the security groups are allowed to access the instance.

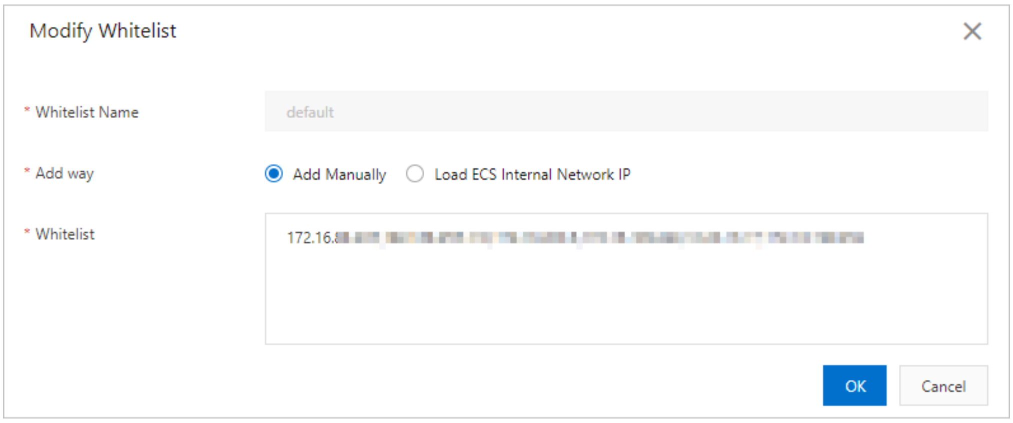
Method 1: Manually add a whitelist

1.
2. In the left-side navigation pane, click **Whitelist Settings**.
3. Find the **default** whitelist and click **Modify**.

   > **Note** You can also click **Add Whitelist** to create a whitelist. The name of a whitelist must be 2 to 32 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.

4. In the dialog box that appears, perform one of the following operations:
   - Manually add IP addresses or CIDR blocks to the whitelist.

     Manually modify the whitelist



Modify Whitelist

* Whitelist Name: default

* Add way: ● Add Manually ○ Load ECS Internal Network IP

* Whitelist: 172.16.[redacted]

Note

- Separate multiple IP addresses with commas (,). A maximum of 1,000 unique IP addresses can be added. You can enter specific IP addresses and CIDR blocks described in the following section:
  - Specific IP addresses. Example: 10.23.12.24.
  - CIDR blocks. Example: 10.23.12.0/24. /24 indicates the length of the IP address prefix. An IP address prefix can be 1 to 32 bits in length. 10.23.12.0/24 indicates an IP address range from 10.23.12.0 to 10.23.12.255. For more information about CIDR blocks, see FAQ about CIDR blocks.
- If you enter CIDR blocks that have a prefix length of 0 such as 0.0.0.0/0 and 127.0.0.1/0, all IP addresses are allowed to access the instance. This poses a high security risk. Proceed with caution.

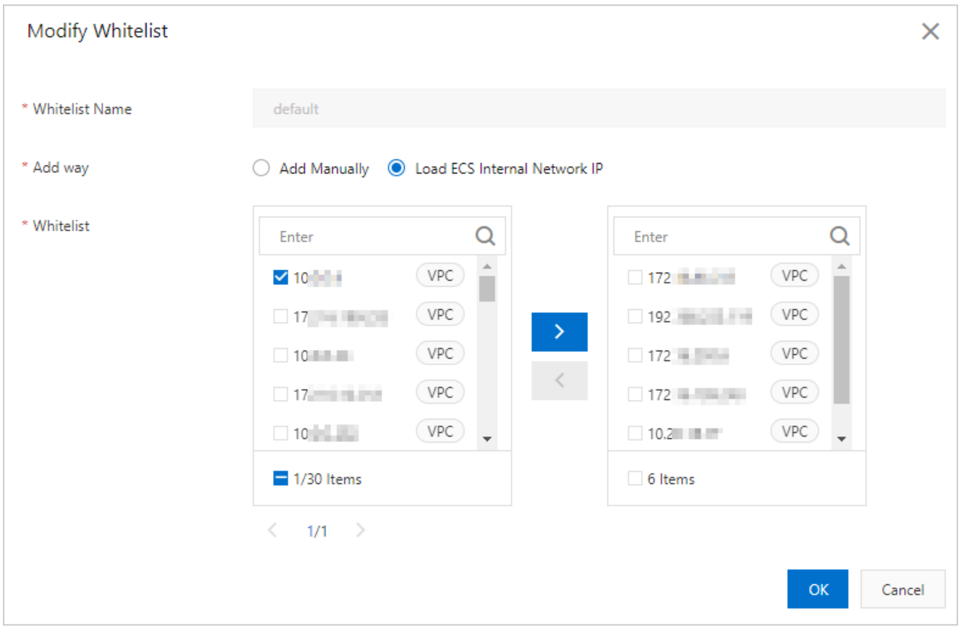
- Add private IP addresses of ECS instances to the whitelist.

  a. Click **Load ECS Internal Network IP**.

  The private IP addresses of ECS instances that are deployed in the same region as the ApsaraDB for Redis instance are displayed.

  b. Select IP addresses based on your business requirements.

  Select private IP addresses of ECS instances



Modify Whitelist ×

* Whitelist Name: default

* Add way: ○ Add Manually ● Load ECS Internal Network IP

* Whitelist

Enter

☑ 10[redacted] VPC

□ 17[redacted] VPC

□ 10[redacted] VPC

□ 17[redacted] VPC

□ 10[redacted] VPC

1/30 Items

1/1

Enter

□ 172 [redacted] VPC

□ 192 [redacted] VPC

□ 172 [redacted] VPC

□ 172 [redacted] VPC

□ 10.2[redacted] VPC

□ 6 Items

Note To find the ECS instance that is assigned a specific IP address, you can move the pointer over the IP address. Then, the system displays the ID and name of the ECS instance to which the IP address is assigned.

5. Click **OK**.
6. (Optional) To delete all IP addresses of a whitelist, click **Delete** in the Actions column corresponding to the whitelist.

Default whitelists generated by the system cannot be deleted, such as **default** and **hdm_security_ips**.

Method 2: Add ECS security groups as whitelists

You can add ECS security groups as whitelists for the ApsaraDB for Redis instance. Then, the ECS instances in the security groups can access the ApsaraDB for Redis instance over an internal network or the Internet. The ApsaraDB for Redis instance must have a public endpoint if you want to access the instance over the Internet. For more information, see Use a public endpoint to connect to an ApsaraDB for Redis instance.

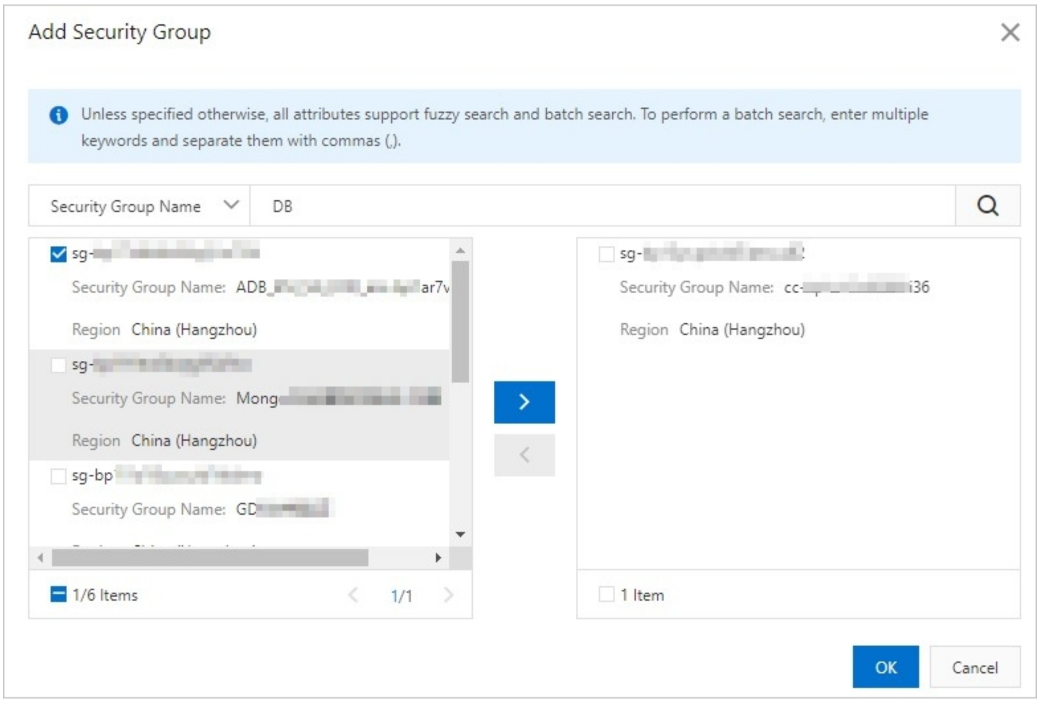
Note

- Before you add a security group as a whitelist, make sure that the network types of the ApsaraDB for Redis instance and the ECS instances in the security group are the same. If the network types of the ApsaraDB for Redis instance and ECS instances are VPC, make sure that they are deployed in the same VPC.
- You cannot add ECS security groups as whitelists for ApsaraDB for Redis instances deployed in the following regions: China (Heyuan), China (Guangzhou), China (Nanjing), and China (Ulanqab).

1.
2. In the left-side navigation pane, click **Whitelist Settings**.
3. Click **Security Groups**.
4. On the **Security Groups** tab, click **Add Security Group**.
5. In the dialog box that appears, select the security groups that you want to add as whitelists.

   You can use a **security group name** or **security group ID** to perform fuzzy search.

Add security groups



Add Security Group

Unless specified otherwise, all attributes support fuzzy search and batch search. To perform a batch search, enter multiple keywords and separate them with commas (,).

Security Group Name: DB

☑ sg-[redacted]
Security Group Name: ADB_[redacted]ar7v
Region China (Hangzhou)

□ sg-[redacted]
Security Group Name: Mong[redacted]
Region China (Hangzhou)

□ sg-bp[redacted]
Security Group Name: GD[redacted]

1/6 Items 1/1

□ sg-[redacted]
Security Group Name: cc-[redacted]i36
Region China (Hangzhou)

□ 1 Item

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

6. Click **OK**.
7. (Optional) To remove all security groups, click **Delete**.

References

- Use a public endpoint to connect to an ApsaraDB for Redis instance
- Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks

Related API operations

API	Description
DescribeSecurityIps	Queries the IP address whitelists of an ApsaraDB for Redis instance.
ModifySecurityIps	Modifies the IP address whitelists of an ApsaraDB for Redis instance.
DescribeSecurityGroupConfiguration	Queries the security groups that are added as whitelists to an ApsaraDB for Redis instance.
ModifySecurityGroupConfiguration	Modifies the security groups that are added as whitelists to an ApsaraDB for Redis instance.

FAQ

- Q: Why are whitelists automatically created for an ApsaraDB for Redis instance? Can I delete these whitelists?

  A: After you create an ApsaraDB for Redis instance, a default whitelist is automatically created. After you perform specific operations on the instance, more whitelists are automatically created, as described in the following table.

Whitelist name	Source
default	The default whitelist that cannot be deleted.
ali_dms_group	This whitelist is automatically created by Data Management (DMS) when you log on to an ApsaraDB for Redis instance from DMS. For more information, see Log on to an ApsaraDB for Redis instance by using DMS. Do not delete or modify this whitelist. Otherwise, you may be unable to log on to the ApsaraDB for Redis instance from DMS.
hdm_security_ips	This whitelist is automatically created by Database Autonomy Service (DAS) when you use CloudDBA-related features such as cache analysis. For more information, see Offline key analysis. Do not delete or modify this whitelist. Otherwise, the CloudDBA-related features may become unavailable.

- Q: A whitelist contains IP address 127.0.0.1 in addition to client IP addresses. In this case, can these clients connect to the ApsaraDB for Redis instance?

A: Yes, these clients can connect to the ApsaraDB for Redis instance. If only 127.0.0.1 exists in the whitelist, all IP addresses are not allowed to connect to the ApsaraDB for Redis instance.

- Q: Why does the `(error) ERR illegal address` message appear after I use the redis-cli tool to connect to an ApsaraDB for Redis instance?

  A: The IP address of the client where you use the redis-cli tool is not added to a whitelist of the ApsaraDB for Redis instance. You must check the whitelists of the ApsaraDB for Redis instance.

- Q: If the IP address of my client is not added to a whitelist of an ApsaraDB for Redis instance, can I check port connectivity by running the telnet command?

  A: Yes, you can run the telnet command to check port connectivity. The following output is returned after you run the telnet command:

  ```
  Escape character is '^]'.
  Connection closed by foreign host.
  ```

6.4. Configure SSL encryption

This topic describes how to enable SSL encryption for an ApsaraDB for MongoDB instance to enhance link security. After you enable SSL encryption, you must install SSL certificates that are issued by certificate authorities (CAs) on your application. SSL encryption can encrypt connections at the transport layer to increase data security and ensure data integrity.

Prerequisites

- The major version of the instance is Redis 4.0 or 5.0. The instance is a cluster master-replica instance. For more information, see Cluster master-replica instances.

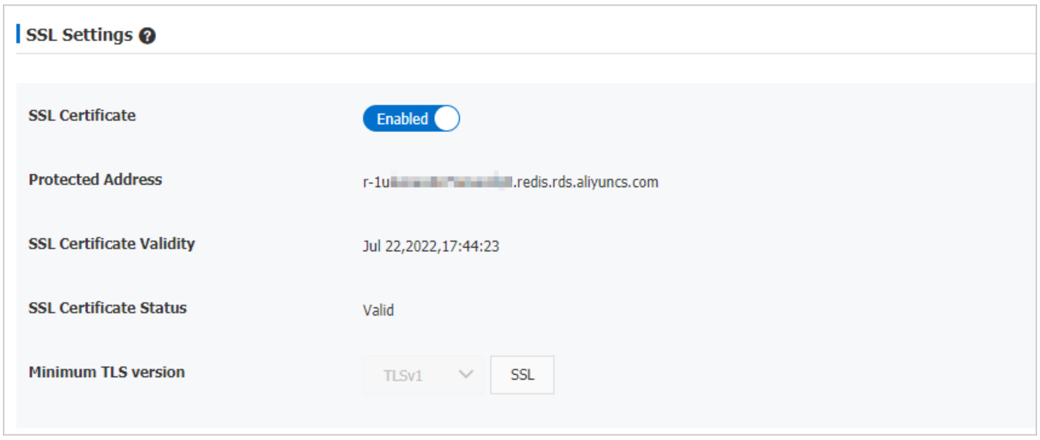
Precautions

- An SSL certificate remains valid for one year. Before the used SSL certificate expires, you must update its validity period. In addition, you must download the required SSL certificate file and configure the certificate again. Otherwise, clients cannot connect to your instance over an encrypted connection.
- SSL encryption may increase the network latency of instances. We recommend that you enable this feature only when encryption needs arise. For example, you can enable SSL encryption if you connect to an ApsaraDB for Redis instance over the Internet.
- The instance restarts after you enable SSL encryption or update the certificate validity period. The instance may experience a transient connection of a few seconds. We recommend that you perform this operation during off-peak hours and make sure that your application can automatically reconnect to the instance.
- After you enable SSL encryption for an instance, both SSL and non-SSL connections are supported.

Procedure

1.
2. In the left-side navigation pane, click **SSL Settings**.
3. Perform one of the following operations.

   Configure SSL encryption



SSL Settings

SSL Certificate	Enabled
Protected Address	r-1u[redacted].redis.rds.aliyuncs.com
SSL Certificate Validity	Jul 22,2022,17:44:23
SSL Certificate Status	Valid
Minimum TLS version	TLSv1 SSL

Operation	Procedure
Enable or disable SSL encryption	Turn on or off **SSL Certificate**.
Modify the earliest TLS version supported by the instance	Click **SSL** next to **Minimum TLS version**, select a TLS version from the drop-down list, and then click **Save**. The default value is TLSv1. Note ○ If the **Minimum TLS version** drop-down list is unavailable, you must update your instance to the latest minor version. For more information, see Update the minor version. ○ This operation is not supported if you use a standard master-replica instance that runs Redis 2.8. For more information, see Standard master-replica instances.
Update the CA certificate	Click **Update Validity** in the upper-right corner of the page and then click **OK**. The CA certificate remains valid for one year. You can click **Update Validity** and then download and configure the CA certificate again. After the CA certificate is updated, it is valid for another year.
Download the CA certificate	In the upper-right corner, click **Download SSL Certificate**.

FAQ

- Q: What do I do if the error message "version not supported" appears?

  A: You must update your instance to the latest minor version. For more information, see Update the minor version.

- Q: What files are included in the downloaded CA certificate?

  A: The downloaded CA certificate is a compressed package that consists of the following files:

  - ApsaraDB-CA-Chain.p7b: imports the CA certificate into the Windows operating system.

  - ApsaraDB-CA-Chain.pem: imports the CA certificate into other operating systems such as Linux or applications.
  - ApsaraDB-CA-Chain.jks: stores truststore certificates in Java and imports the CA certificate chain into Java applications.

SSL connection methods

- Use redis-cli to connect to an ApsaraDB for Redis instance
- Use a client to connect to an ApsaraDB for Redis instance that has SSL encryption enabled

Related API operations

Operation	Description
ModifyInstanceSSL	Modifies SSL encryption configurations for an ApsaraDB for Redis instance.

6.5. Enable TDE

ApsaraDB for Redis provides Transparent Data Encryption (TDE), which can be used to encrypt and decrypt Redis Database (RDB) files. You can enable TDE in the ApsaraDB for Redis console to allow the system to encrypt and decrypt RDB files. This improves data security and compliance.

Prerequisites

- The instance is a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances.
- The minor version of the instance is 1.7.1 or later. For more information about how to view and update the minor version, see Update the minor version.

Context

TDE encrypts RDB files before they are written to disks and decrypts RDB files when they are read to the memory from disks. TDE does not increase the sizes of RDB files. When you use TDE, you do not need to modify your client.

Impacts

You cannot disable TDE after it is enabled. You must evaluate the impacts on your business before you enable TDE. Take note of the following impacts:

- After TDE is enabled for an instance, the instance cannot be migrated across zones. For more information, see Migrate an instance across zones.
- After TDE is enabled for an instance, the offline key analysis feature is not supported for the instance. For more information, see Offline key analysis.
- After TDE is enabled for an instance, the instance cannot be converted into a child instance of a distributed instance. For more information, see Create a distributed instance.
- After TDE is enabled for an instance, instance data cannot be migrated or synchronized by using Data Transmission Service (DTS) or redis-shake. For more information about redis-shake and DTS, see RedisShake and What is DTS?

Precautions

- TDE can be enabled for an instance but not for a key or a database.
- TDE encrypts RDB files that are written to disks, such as *dump.rdb*.
- Key Management Service (KMS) generates and manages the keys used by TDE. For more information about KMS, see What is Key Management Service? ApsaraDB for Redis does not provide keys or certificates required for encryption.

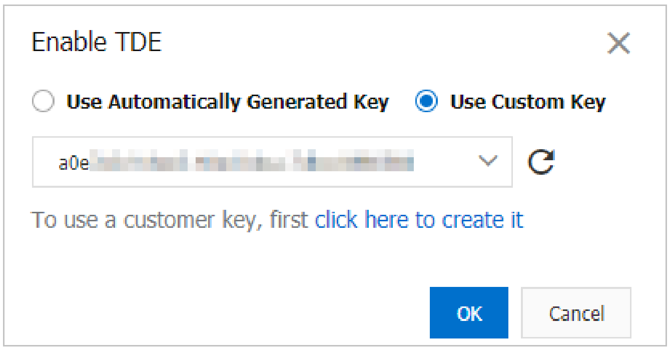
Procedure

1.
2. In the left-side navigation pane, click **TDE Settings**.
3. Turn on the switch next to **TDE Status** to enable TDE.

   > **Note** If an earlier minor version is used, the switch is dimmed. For more information about how to view and update the minor version, see Update the minor version.

4. In the dialog box that appears, select **Use Automatically Generated Key** or **Use Custom Key** and then click **OK**.

   Select key type for enabling TDE



Enable TDE

○ Use Automatically Generated Key ● Use Custom Key

a0e[redacted]

To use a customer key, first click here to create it

> **Note**
>
> - The first time you enable TDE for an instance within your Alibaba Cloud account, follow the instructions on the page to authorize the **AliyunRdsInstanceEncryptionDefaultRole** role. KMS can be used only after the authorization is complete.
> - For more information about how to create a custom key, see Create a CMK.

When the instance state changes from **Modifying TDE** to **Running**, the configurations are complete.

Related API operations

Operation	Description
ModifyInstanceTDE	Enables TDE for an ApsaraDB for Redis instance. You can use automatically generated keys or existing custom keys.
DescribeInstanceTDEStatus	Queries whether TDE is enabled for an ApsaraDB for Redis instance.

Operation	Description
DescribeEncryptionKeyList	Queries the custom keys that are available for an ApsaraDB for Redis instance to use TDE.
DescribeEncryptionKey	Queries the details of a custom key for an ApsaraDB for Redis instance to use TDE.
CheckCloudResourceAuthorized	Queries whether an ApsaraDB for Redis instance has the permissions to use KMS.

FAQ

- Q: How do I decrypt an encrypted RDB file?

  A: RDB files cannot be decrypted. You can restore the file to a new instance. After the restoration is complete, the data is automatically decrypted.

- Q: Why is the data read by clients still displayed in plaintext?

  A: Only RDB files written to disks are encrypted. The data read by clients is read from memory and is not encrypted. That is why it is displayed in plaintext.

6.6. Enable password-free access

ApsaraDB for Redis allows you to enable password-free access for instances that are deployed in a virtual private cloud (VPC). This feature provides a secure and convenient method to connect to an instance. After password-free access is enabled for an instance located in a VPC, clients within the same VPC can access the instance without using a password. Meanwhile, you can still use a username and a password to connect to the instance.

Prerequisites

The instance is deployed in a VPC.

> **Note** If the network type of the instance is classic network, you must change the network type to VPC. For more information, see Change the network type from classic network to VPC.

Precautions

- After you enable password-free access for an instance, the default account is used to connect to the instance. The username of the default account is the same as the instance ID, such as r-bp1zxszhcgatnx****. The default account has read and write permissions on the instance.
- After password-free access is enabled for an instance, the system still prompts for a password when you connect to the instance by using a public endpoint to enhance security.

  > **Note** If you cannot connect to an instance by using a public endpoint, update the instance to the latest minor version. For more information, see Update the minor version.

- By default, the #no_loose_check-whitelist-always parameter of an instance is set to *no*. This way, after password-free access is enabled, clients within the same VPC can directly connect to the instance without the need to add the IP addresses of the clients to a whitelist of the instance. For more information, see Modify parameters of an instance.

Note

- If the `(error) ERR illegal address` error message is returned when you run commands on an instance that has password-free access enabled, the IP address of the client that you are using is not added to a whitelist of the instance.

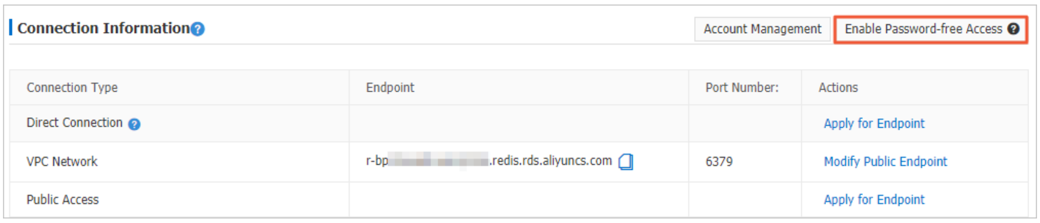
  You can add the IP address to a whitelist of the instance. Alternatively, you can set #no_loose_check-whitelist-always to *no*. This way, the system does not check whether IP addresses are included in instance whitelists.

- The #no_loose_check-whitelist-always parameter cannot be specified for instances that use cloud disks. For more information, see Supported parameters.

Procedure

1.
2. In the upper-right corner of the **Connection Information** section, click **Enable Password-free Access**.

   Enable password-free access

   Connection Information Account Management Enable Password-free Access

Connection Type	Endpoint	Port Number:	Actions
Direct Connection			Apply for Endpoint
VPC Network	r-bp[redacted].redis.rds.aliyuncs.com	6379	Modify Public Endpoint
Public Access			Apply for Endpoint

3. In the panel that appears, read the prompt and click **OK**.

   After you refresh the page, the **Enable Password-free Access** button changes to **Disable Password-free Access**. You can click this button to disable password-free access.

   Note If password-free access is disabled for an instance, clients that use password-free access to connect to the instance can no longer access the instance. To allow these clients to connect to the instance after password-free access is disabled, you must configure the clients to use a username and a password for authentication.

Connection example

The following code shows how to connect to an instance that has password-free access enabled:

Note For information about how to obtain the endpoint and account password of an ApsaraDB for Redis instance, see View endpoints.

○ redis-cli password-free logon ○ Jedis password-free logon

```
redis-cli -h host -p port
// Example: redis -h r-bp10noxlhcoim2****.redis.rds.aliyuncs.com -p 6379
```

```
JedisPoolConfig config = new JedisPoolConfig();
// Maximum number of idle connections allowed. You can set this parameter based on your
needs. Make sure that the specified value does not exceed the maximum number of connections
that the instance supports.
config.setMaxIdle(100);
// Maximum number of connections allowed. You can set this parameter based on your needs.
Make sure that the specified value does not exceed the maximum number of connections that
the instance supports.
config.setMaxTotal(200);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
// Replace the values of the host and port parameters with the endpoint and port number of
the instance respectively. The password parameter is not required.
String host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
int port = 6379;
JedisPool pool = new JedisPool(config, host, port);
Jedis jedis = null;
try
{
    jedis = pool.getResource();
    /// ... do stuff here ... for example
    jedis.set("foo", "bar");
    System.out.println(jedis.get("foo"));
    jedis.zadd("sose", 0, "car");
    jedis.zadd("sose", 0, "bike");
    System.out.println(jedis.zrange("sose", 0, -1));
}
finally
{
    if(jedis != null)
    {
        // Close connections after each API operation is complete. To close a connection,
release the connection to the connection pool instead of destroying the connection.
        jedis.close();
    }
}
// Call only once when you exit.
pool.destroy();
```

Related API operations

Operation	Description
ModifyInstanceVpcAuthMode	Enables or disables password-free access for an ApsaraDB for Redis instance that is deployed in a VPC.

FAQ

- Q: Why does the `WRONGPASS invalid username-password pair` error message appear after I enabled password-free access for an instance deployed in a VPC?

A: This error is returned because you use a Community Edition instance that runs Redis 6.0. If you use the instance and enter a wrong account password, the system returns this error. Enter the correct account password or left the account password field empty.

> ⓘ **Note** The following section describes the password format:
>
> - If you use the default account whose username is the same as the instance ID, you can enter only the password.
> - If you use a custom account, the account password follows the `<user>:<password>` format. Example: `testaccount:Rp829dlwa` .

6.7. Enable the release protection feature

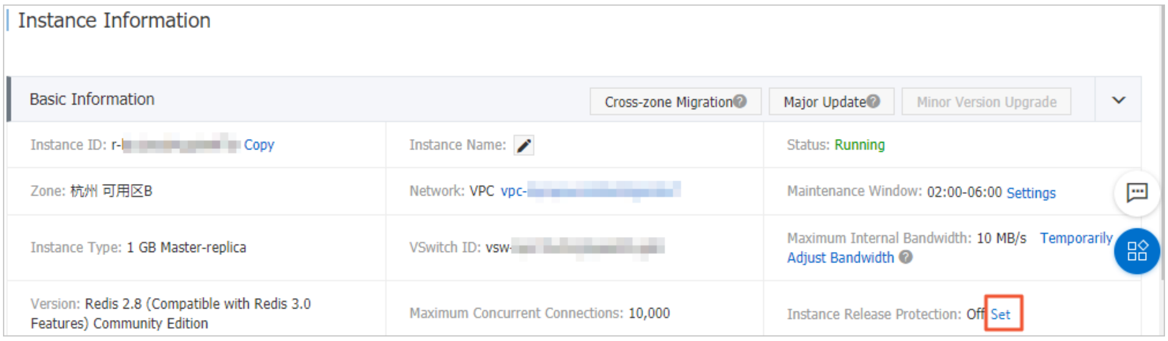
To prevent an instance from being released by accident, ApsaraDB for Redis provides the release protection feature. This topic describes how to enable the feature in the ApsaraDB for Redis console.

Prerequisites

The billing method of the instance for which you want to enable the feature is pay-as-you-go.

Procedure

1.
2. In the **Basic information** section, click **Settings** to the right of **Instance Release Protection**.



Instance Information

Basic Information

Instance ID: r-[redacted] Copy	Instance Name:	Status: Running
Zone: 杭州 可用区B	Network: VPC vpc-[redacted]	Maintenance Window: 02:00-06:00 Settings
Instance Type: 1 GB Master-replica	VSwitch ID: vsw-[redacted]	Maximum Internal Bandwidth: 10 MB/s Temporarily Adjust Bandwidth
Version: Redis 2.8 (Compatible with Redis 3.0 Features) Community Edition	Maximum Concurrent Connections: 10,000	Instance Release Protection: Off Set

3. In the Set Release Protection panel, turn on **Release Protection**.
4. Click **OK**.

> ⓘ **Note** After you enable release protection for an instance, you cannot release the instance in the console or by calling an API operation. To release the instance, you must disable the instance release protection feature first.

7.Connection management

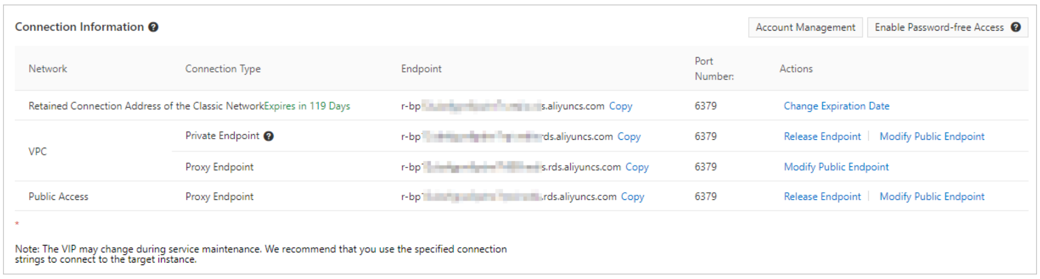
7.1. View endpoints

This topic describes how to view the endpoints of an ApsaraDB for Redis instance in the ApsaraDB for Redis console. The virtual IP address (VIP) of an ApsaraDB for Redis instance may change due to upgrades or maintenance. We recommend that you connect to an ApsaraDB for Redis instance by using an endpoint to ensure service availability. You can view different types of endpoints in the ApsaraDB for Redis console.

Procedure

1.
2. In the **Connection Information** section, view different types of endpoints and port numbers.

View endpoints



Connection Information

Network	Connection Type	Endpoint	Port Number:	Actions
Retained Connection Address of the Classic NetworkExpires in 119 Days		r-bp[redacted]s.aliyuncs.com Copy	6379	Change Expiration Date
VPC	Private Endpoint	r-bp[redacted]ds.aliyuncs.com Copy	6379	Release Endpoint | Modify Public Endpoint
	Proxy Endpoint	r-bp[redacted]s.rds.aliyuncs.com Copy	6379	Modify Public Endpoint
Public Access	Proxy Endpoint	r-bp[redacted].rds.aliyuncs.com Copy	6379	Release Endpoint | Modify Public Endpoint

*

Note: The VIP may change during service maintenance. We recommend that you use the specified connection strings to connect to the target instance.

Endpoint type	Condition	Description
Private endpoint	If your instance is a cluster instance, you can apply for a private endpoint.	For an ApsaraDB for Redis instance that uses the cluster architecture, you can apply for a private endpoint. You can use the endpoint to directly access backend data shards. This connection method is similar to the method that is used to connect to a native Redis cluster. Compared with the the direct connection mode reduces the response time of ApsaraDB for Redis because requests do not need to pass through proxy servers. For more information about how to enable the direct connection mode, see Enable the direct connection mode. **Note** The virtual private cloud (VPC) endpoints of cluster instances and read/write splitting instances are used for the. proxy mode proxy mode

Endpoint type	Condition	Description
VPC endpoint	N/A	A VPC is a private network dedicated to your Alibaba Cloud account. VPCs are logically isolated from each other at Layer 2 to provide higher security and performance. When redis-cli is deployed on an Elastic Compute Service (ECS) instance, you can connect to an ApsaraDB for Redis instance over a VPC to gain higher security and reduce network latency. For more information, see What is ECS?
Classic network endpoint	When you create an ApsaraDB for Redis instance, you select the classic network.	Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by using security groups or whitelists. We recommend that you use a VPC to improve security. For more information about how to change the network type of an ApsaraDB for Redis instance, see Change the network type from classic network to VPC.
Public endpoint	You must apply for the endpoint.	If you want to connect an on-premises host to an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance. **Note** You are not charged for traffic generated when you connect to the ApsaraDB for Redis instance over the Internet. However, the connection may be exposed to security risks. We recommend that you use a VPC to improve security.

Related information

- Connect to an ApsaraDB for Redis instance

7.2. Log on to an ApsaraDB for Redis instance by using DMS

You can connect to an ApsaraDB for Redis instance by using Data Management (DMS) without the need to install clients. DMS allows you to visually manage ApsaraDB for Redis instances.

Context

DMS is an all-in-one data management service that supports multiple relational databases and NoSQL databases. The service offers features such as data management, schema management, user authorization, security audit, data trend analysis, and data tracking. For more information about DMS, see Overview. You can use DMS to manage databases with ease. This enhances data security, improves management efficiency, and maximizes data value.

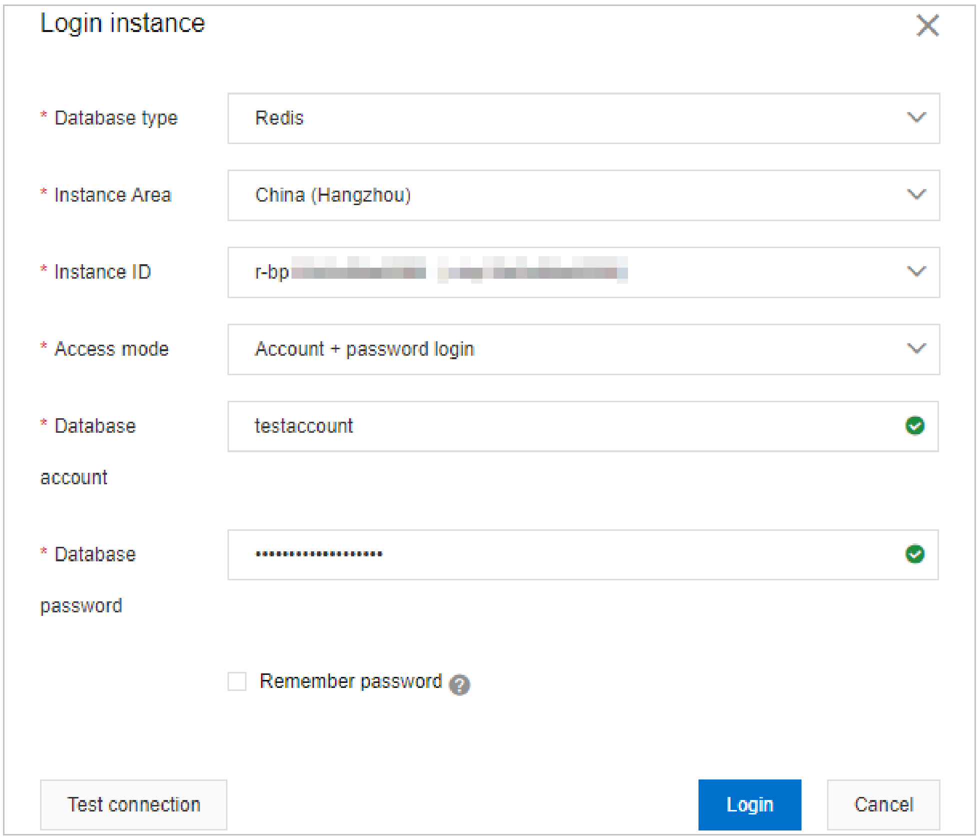
Limits

DMS has limits on Redis commands. For more information, see SQL Console for Redis.

Procedure

1.
2. In the upper-right corner of the page, click **Log into Database**.
3. In the DMS console to which you are redirected, configure the logon mode.

Configure the DMS logon mode



Login instance

* Database type: Redis

* Instance Area: China (Hangzhou)

* Instance ID: r-bp[redacted]

* Access mode: Account + password login

* Database account: testaccount

* Database password: ••••••••••••••••••••

□ Remember password

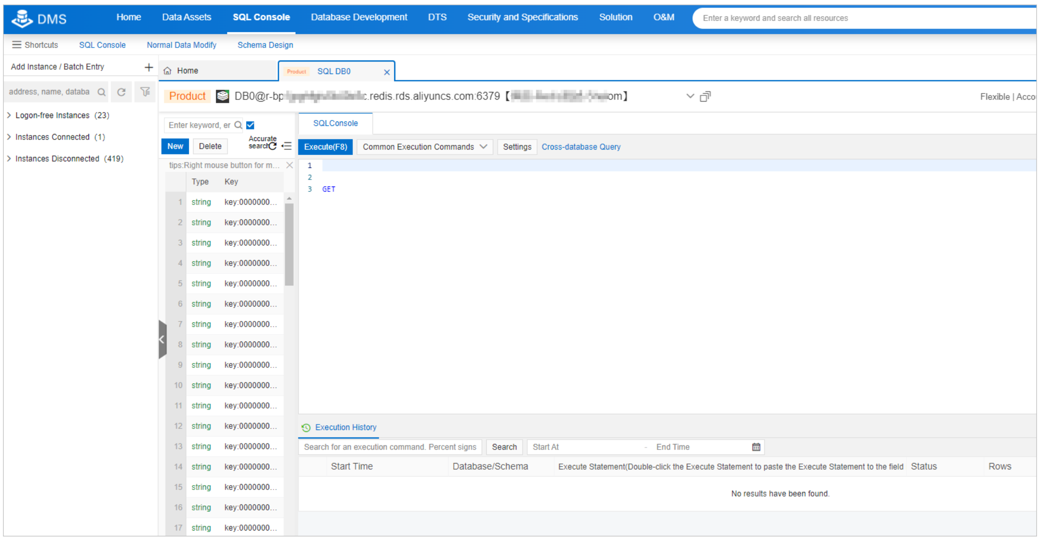
Access mode	Description

Access mode	Description
Account + password login (recommended)	Enter the database account and password. For more information about how to create a database account, see Create and manage database accounts. Note ◦ By default, an ApsaraDB for Redis instance contains a database account named after the instance ID. Example: r-bp10noxlhcoim2****. You can use this account for logon. The account password is set when you create the instance. ◦ If you forget your password, you can reset it. For more information, see Change or reset the password.
noSecret login	If you enable the password-free access for the ApsaraDB for Redis instance, you can log on to the instance without using passwords. For more information, see Enable password-free access.
password login	Use the password that is specified when your instance is created to log on to the ApsaraDB for Redis instance. The password is created for the database account that is named after the instance ID. If you forget the password, you can reset it. For more information, see Change or reset the password.

4. Click **Login**.

> Note If you do not add the IP address of the DMS server to a whitelist of the ApsaraDB for Redis instance, a dialog box appears and prompts you to click **Set IP address whitelists**. The system creates a whitelist named **ali_dms_group** and adds the IP address of the DMS server to this whitelist.

Complete logon in the example


DMS
Home
Data Assets
SQL Console
Database Development
DTS
Security and Specifications
Solution
O&M
Enter a keyword and search all resources
Shortcuts
SQL Console
Normal Data Modify
Schema Design
Add Instance / Batch Entry
address, name, databa
Logon-free Instances (23)
Instances Connected (1)
Instances Disconnected (419)
Home
SQL DB0
Product
DB0@r-bp[redacted].redis.rds.aliyuncs.com:6379 [[redacted]om]
Flexible | Acco
Enter keyword, er
New
Delete
Accurate search
SQLConsole
Execute(F8)
Common Execution Commands
Settings
Cross-database Query
tips:Right mouse button for m...
Type
Key
string
key:0000000...
GET
Execution History
Search for an execution command. Percent signs
Search
Start At
End Time
Start Time
Database/Schema
Execute Statement(Double-click the Execute Statement to paste the Execute Statement to the field
Status
Rows
No results have been found.

5. After your logon, you can run Redis commands on the **SQLConsole** page. For example, you can run the **DBSIZE** command to query the number of keys in the current database.

   For more information about Redis commands supported by ApsaraDB for Redis, see Overview. For more information about all Redis commands and their usage, visit the Redis official website.

What's next

Manage ApsaraDB for Redis instances by using DMS

Related information

- Use redis-cli to connect to an ApsaraDB for Redis instance
- Use a client to connect to an ApsaraDB for Redis instance

7.3. Use redis-cli to connect to an ApsaraDB for Redis instance

redis-cli is a built-in command line interface (CLI) that is native to Redis. You can use redis-cli to connect to an ApsaraDB for Redis instance from an Elastic Compute Service (ECS) instance or on-premises device and manage data.

Type of network used to connect to an ApsaraDB for Redis instance

Network type	Description
VPC (recommended)	A virtual private cloud (VPC) is a private network dedicated to your Alibaba Cloud account. VPCs are logically isolated from each other to provide higher security and performance. When redis-cli is deployed on an ECS instance, you can connect redis-cli to an ApsaraDB for Redis instance over a VPC to gain higher security and reduce network latency. For more information, see What is ECS?

Network type	Description
Internet	If you want to test or manage an ApsaraDB for Redis instance from an on-premises device, you can deploy redis-cli on the device and then connect to the ApsaraDB for Redis instance over the Internet. ? **Note** Although you are not charged to connect to the ApsaraDB for Redis instance over the Internet, the connection may not be secure and incur other risks. We recommend that you use a VPC to ensure better security.

redis-cli

- redis-cli is a built-in CLI that is native to Redis. You can deploy the tool on an ECS instance or an on-premises device by installing Redis.

  > ? **Note** In addition to redis-cli, you can also use Data Management (DMS) to connect to ApsaraDB for Redis databases without the need to install a client. DMS provides a graphic user interface for database management. For more information about DMS, see Overview. For more information about the procedure, see Log on to an ApsaraDB for Redis instance by using DMS.

- The redis-cli version can be different from the major version of the ApsaraDB for Redis instance.

Use redis-cli in Linux

1. Install redis-cli. Skip this step if it is already installed.

   i. Log on to the device on which you want to install redis-cli, such as an ECS instance or an on-premises device.

   ii. Run the following command to download the Redis source code package:

   ```
   wget https://download.redis.io/releases/redis-6.0.9.tar.gz
   ```

   > ? **Note** Redis 6.0.9 is used as an example to demonstrate the operations. You can also install other versions. For more information, visit Download.

   iii. Run the following command to decompress the Redis source code package:

   ```
   tar xzf redis-6.0.9.tar.gz
   ```

   iv. Run the following command to go to the directory to which the Redis source code package is decompressed. Then, compile and install Redis.

   ```
   cd redis-6.0.9&&make
   ```

   > ? **Note** It takes two or three minutes to compile and install Redis.

2. Perform the corresponding operations based on the redis-cli installation location.

<table>
<tr><th>redis-cli installation location</th><th>Operation</th></tr>
<tr><td>ECS instance (recommended)</td><td>

i. Make sure that the ECS instance and the ApsaraDB for Redis instance are deployed in the same VPC. In this case, the same VPC ID is displayed in the Basic Information section of the instances.

> Note
>
> - If the instances are deployed in different VPCs, you can change the VPC to which the ECS instance belongs. For more information, see Change the VPC of an ECS instance.
> - The network types of the ECS instance and the ApsaraDB for Redis instance may be different. For example, the ECS instance belongs to the classic network and the ApsaraDB for Redis instance belongs to a VPC. For information about how to connect to an ApsaraDB for Redis instance from an ECS instance when the instances are deployed in different types of networks, see Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks.

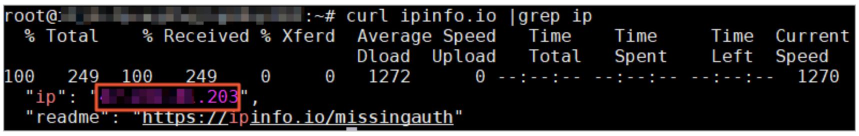
ii. Obtain the internal IP address of the ECS instance. For more information, see Network FAQ.

iii. Add the internal IP address of the ECS instance to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

</td></tr>
<tr><td>On-premises device</td><td>

i. By default, only internal endpoints are available for ApsaraDB for Redis instances. If you want to connect to an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance.

ii. Run the **curl ipinfo.io |grep ip** command on the on-premises device to obtain its public IP address. The following figure shows a sample result.



```
root@[redacted]:~# curl ipinfo.io |grep ip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   249  100   249    0     0   1272      0 --:--:-- --:--:-- --:--:--  1270
  "ip": "[redacted].203",
  "readme": "https://ipinfo.io/missingauth"
```

iii. Add the public IP address of the on-premises device to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

</td></tr>
</table>

3. Obtain the connection information and run the following command to connect to the ApsaraDB for Redis instance:

```
src/redis-cli -h <hostname> -p <port> [-c]
```

Parameter description

Parameter	Description	Method to obtain the parameter value

Parameter	Description	Method to obtain the parameter value
<hostname>	The endpoint of the ApsaraDB for Redis instance.	◦ If you connect an ECS instance to the ApsaraDB for Redis instance over a VPC, obtain the endpoint of the ApsaraDB for Redis instance in the VPC. ◦ If you connect an on-premises device to the ApsaraDB for Redis instance over the Internet, obtain the public endpoint of the ApsaraDB for Redis instance. For more information, see View endpoints.
<port>	The port number of the ApsaraDB for Redis instance.	Use the default port number 6379 or specify a custom port number. For more information, see Change the endpoint or port number of an ApsaraDB for Redis instance.
-c	Enable the cluster mode	Add -c only when you connect to an ApsaraDB for Redis cluster instance by using a private endpoint. For more information, see Cluster master-replica instances and Enable the direct connection mode.

The following sample command is suitable for scenarios where ApsaraDB for Redis instances are connected by using default endpoints, such as endpoints of standard instances and proxy endpoints of cluster instances:

```
src/redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379
```

The following sample command is suitable for scenarios where ApsaraDB for Redis instances are connected by using private endpoints:

```
src/redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379 -c
```

4. Run the following command to verify the password:

```
AUTH <password>
```

> **Note** If password-free access is enabled for an ApsaraDB for Redis instance, you can run Redis commands without performing this step when you connect to the instance over a VPC. For more information, see Enable password-free access.

<password>: the password of a specific account. The password format varies based on the selected account. If you forget your password, you can reset it. For more information, see Change or reset the password.

- If you use the default account whose username is the same as the instance ID, enter only the password.
- If you use a custom account, the format of the password must be `<user>:<password>` . A password in this format can also be used for default account logon. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa` , you must enter `testaccount:Rp829dlwa` as the database password.

Example:

```
AUTH testaccount:Rp829dlwa
```

If the password succeeds, the following result is returned:

```
OK
```

> **Note** For more information about what to do if an error occurs, see Common connection errors.

5. After the verification is complete, run Redis commands. For example, run the **DBSIZE** command to query the number of keys in the current database.

   For more information about Redis commands supported by ApsaraDB for Redis, see Overview. For more information about all Redis commands and their usage, visit the Redis official website.

Use redis-cli in Windows

> **Note** Only 64-bit Windows operating systems are supported.

1. Install redis-cli. Skip this step if it is already installed.
   i. Log on to the device on which you want to install redis-cli, such as an ECS instance or an on-premises device.
   ii. Download the Redis-x64-3.2.100.zip package.

      > **Note** Redis 3.2 (the latest version) is used in this example to demonstrate the operations. You can also install other versions. For more information, visit Microsoft Archive-Redis.

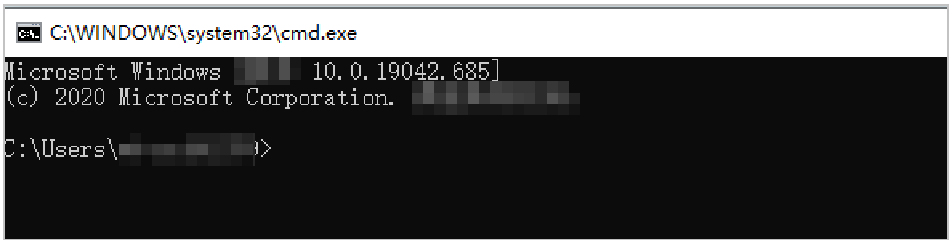
   iii. Decompress the *Redis-x64-3.2.100.zip* package to the directory where you want to install Redis. The *D:\Redis-x64-3.2.100* directory is used in this example.
2. Perform the corresponding operations based on the redis-cli installation location.

redis-cli installation location	Operation
ECS instance (recommended)	i. Make sure that the ECS instance and the ApsaraDB for Redis instance are deployed in the same VPC. In this case, the same VPC ID is displayed in the Basic Information section of the instances. **Note** If the instances are deployed in different VPCs, you can change the VPC to which the ECS instance belongs. For more information, see Change the VPC of an ECS instance. ii. Obtain the internal IP address of the ECS instance. For more information, see Network FAQ. iii. Add the internal IP address of the ECS instance to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

redis-cli installation location	Operation
On-premises device	i. By default, only internal endpoints are available for ApsaraDB for Redis instances. If you want to connect to an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance. ii. On the on-premises device, visit ipinfo to obtain the public IP address of the on-premises device. iii. Add the public IP address of the on-premises device to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

3. On the ECS instance or on-premises device where redis-cli is installed, press Win+R to open the **Run** dialog box. In the Run dialog box, enter *cmd*, and then click **OK**.
   The Windows CLI is opened.

   Windows CLI

   C:\WINDOWS\system32\cmd.exe

   ```
   Microsoft Windows [redacted] 10.0.19042.685]
   (c) 2020 Microsoft Corporation. [redacted]

   C:\Users\[redacted]>
   ```

4. Run the following command to access the directory where redis-cli is installed:

   ```
   cd /d <path>
   ```

   <path>: the full path of the directory to which the *Redis-x64-3.2.100.zip* package is decompressed. *D:\Redis-x64-3.2.100* is used in this example.

   Example:

   ```
   cd /d D:\Redis-x64-3.2.100
   ```

5. Obtain the connection information and run the following command to connect to the ApsaraDB for Redis instance:

   ```
   redis-cli -h <hostname> -p <port> [-c]
   ```

   > **Note** If redis-cli is installed on PowerShell, run the following command to use redis-cli:
   >
   > ```
   > .\redis-cli -h <hostname> -p <port> [-c]
   > ```

   Parameter description

Parameter	Description	Method to obtain the parameter value

Parameter	Description	Method to obtain the parameter value
<hostname>	The endpoint of the ApsaraDB for Redis instance.	○ If you connect an ECS instance to the ApsaraDB for Redis instance over a VPC, obtain the endpoint of the ApsaraDB for Redis instance in the VPC. ○ If you connect an on-premises device to the ApsaraDB for Redis instance over the Internet, obtain the public endpoint of the ApsaraDB for Redis instance. For more information, see View endpoints.
<port>	The port number of the ApsaraDB for Redis instance.	Use the default port number 6379 or specify a custom port number. For more information, see Change the endpoint or port number of an ApsaraDB for Redis instance.
-c	Enable the cluster mode	Add -c only when you connect to an ApsaraDB for Redis cluster instance by using a private endpoint. For more information, see Cluster master-replica instances and Enable the direct connection mode.

The following sample command is suitable for scenarios where ApsaraDB for Redis instances are connected by using default endpoints, such as endpoints of standard instances and proxy endpoints of cluster instances:

```
redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379
```

The following sample command is suitable for scenarios where ApsaraDB for Redis instances are connected by using private endpoints:

```
redis-cli -h r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com -p 6379 -c
```

6. Run the following command to verify the password:

```
AUTH <password>
```

> **Note** If password-free access is enabled for an ApsaraDB for Redis instance, you can run Redis commands without performing this step when you connect to the instance over a VPC. For more information, see Enable password-free access.

<password>: the password of a specific account. The password format varies based on the selected account. If you forget your password, you can reset it. For more information, see Change or reset the password.

- If you use the default account whose username is the same as the instance ID, enter only the password.
- If you use a custom account, the format of the password must be `<user>:<password>`. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa`, you must enter `testaccount:Rp829dlwa` as the database password.

Example:

```
AUTH testaccount:Rp829dlwa
```

If the password succeeds, the following result is returned:

```
OK
```

> ? **Note** For more information about what to do if an error occurs, see Common connection errors.

7. After the verification is complete, run Redis commands. For example, run the **DBSIZE** command to query the number of keys in the current database.

   For more information about Redis commands supported by ApsaraDB for Redis, see Overview. For more information about all Redis commands and their usage, visit the Redis official website.

Common connection errors

> ? **Note** If the minor version of your instance is outdated, the returned error message may be misleading. We recommend that you update your instance to the latest minor version. For more information, see Update the minor version.

Error message	Cause and solution
`(error) ERR illegal address`	A whitelist is not configured as required. For more information, see Step 2 in Use redis-cli in Linux or Use redis-cli in Windows.
`(error) ERR client ip is not in whitelist`	
• `(error) ERR invalid password` • `(error) WRONGPASS invalid username-password pair`	The password is invalid. Make sure that you use the correct password in a valid format. The password format varies based on the selected account. • If you use the default account whose username is the same as the instance ID, you can enter only the password. • If you use a custom account, the format of the password must be `<user>:<password>`. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa`, you must enter `testaccount:Rp829dlwa` as the database password. ? **Note** • If you use a management tool such as Redis Desktop Manager (RDM) to connect to the ApsaraDB for Redis instance, enter a password in the format of `<user>:<password>`. • If you forget your password, you can reset it. For more information, see Change or reset the password.

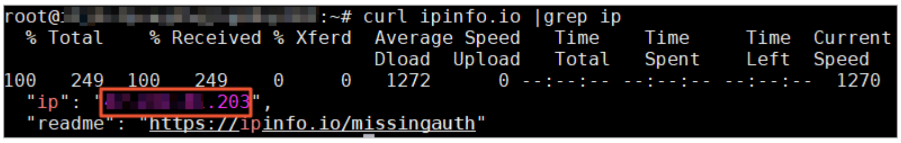
7.4. Use a client to connect to an ApsaraDB for Redis instance

ApsaraDB for Redis is fully compatible with open source Redis. You can connect to an ApsaraDB for Redis instance in the same way as you connect to open source Redis. Therefore, you can use a client that is compatible with the Redis protocol to connect to an ApsaraDB for Redis instance. You can connect to an ApsaraDB for Redis instance by using clients of different programming languages.

Prerequisites

The operations listed in the following table are performed based on the type of host on which the client is deployed.

Host	Operation
ECS instance (recommended)	1. Make sure that the Elastic Compute Service (ECS) instance and the ApsaraDB for Redis instance belong to the same virtual private cloud (VPC). If the VPC IDs of these two instances are the same, the instances belong to the same VPC. Note ◦ If the instances are deployed in different VPCs, you can change the VPC to which the ECS instance belongs. For more information, see Change the VPC of an ECS instance. ◦ The network types of the ECS instance and the ApsaraDB for Redis instance may be different. For example, the ECS instance belongs to the classic network and the ApsaraDB for Redis instance belongs to a VPC. For information about how to connect to an ApsaraDB for Redis instance from an ECS instance when the instances are deployed in different network types, see Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks. 2. Obtain the internal IP address of the ECS instance. For more information, see Network FAQ. 3. Add the internal IP address of the ECS instance to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

<table>
<tr><th>Host</th><th>Operation</th></tr>
<tr><td>On-premises device</td><td>
1. By default, only internal endpoints are available for ApsaraDB for Redis instances. If you want to connect to an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance.
2. Run the <b>curl ipinfo.io |grep ip</b> command on your on-premises device to obtain its public IP address. The following figure shows a sample command output.

<pre>
root@[redacted]:~# curl ipinfo.io |grep ip
  % Total   % Received % Xferd Average Speed   Time   Time   Time Current
                                 Dload Upload   Total   Spent   Left Speed
100   249 100   249   0   0   1272   0 --:--:-- --:--:-- --:--:-- 1270
  "ip": "[redacted].203",
  "readme": "https://ipinfo.io/missingauth"
</pre>
Note If your on-premises device runs a Windows operating system, visit ipinfo to obtain the public IP address.
3. Add the public IP address of your on-premises device to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.
</td></tr>
</table>

Precautions

- By default, cluster or read/write splitting instances run in proxy mode. In this mode, you can access an ApsaraDB for Redis instance by using the endpoint of a proxy node in the instance in the same way as you access an ApsaraDB for Redis standard instance. For more information about cluster and read/write splitting instances, see Cluster master-replica instances or Read/write splitting instances.

  > Note If you use a private endpoint to connect to an ApsaraDB for Redis instance, you can connect to the instance in the same way as you connect to an open source Redis cluster. For more information about private endpoints, see Enable the direct connection mode.

- If password-free access is enabled for an ApsaraDB for Redis instance deployed in a VPC, a client in the same VPC as the instance can connect to the instance without using passwords. For more information, see Enable password-free access.

Obtain connection information

When you use a client to connect to an ApsaraDB for Redis instance, you must obtain the connection information described in the following table and specify the information in the code.

Item	Description
Instance endpoint	ApsaraDB for Redis instances support multiple endpoint types. We recommend that you use VPCs for higher security and lower network latency. For more information, see View endpoints.
Port number	The default port number is 6379. You can also use a custom port number. For more information, see Change the endpoint or port number of an ApsaraDB for Redis instance.

Item	Description
Instance account (this information is optional for specific clients)	By default, an ApsaraDB for Redis instance has a database account that is named after the instance ID. Example: r-bp10noxlhcoim2****. You can create another database account and grant the required permissions to the account. For more information, see Create and manage database accounts.
Password	The password format varies with the selected account: - If you use the default account whose username is the same as the instance ID, enter only the password. - If you use a custom account, enter a password in the `<user>:<password>` format. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa`, you must enter `testaccount:Rp829dlwa` as the database password. Note - If you use a management tool such as Redis Desktop Manager (RDM) to connect to the ApsaraDB for Redis instance, enter a password in the format of `<user>:<password>`. - If you forget your password, you can reset it. For more information, see Change or reset the password.

Common types of clients

For information about all types of clients supported by Redis, see Clients.

- Jedis client
- TairJedis (for ApsaraDB for Redis Enhanced Edition (Tair))
- PhpRedis client
- redis-py client
- C or C++ client
- .NET client
- node-redis client
- C# client StackExchange.Redis
- go-redis client
- Lettuce client (not recommended)

Jedis client

Note We recommend that you select the TairJedis client if you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances.

1. Download and install the Jedis client. For more information, visit Getting started.
2. Select a connection method based on your business requirements.

i. Start the Eclipse client, create a project, and then add the following dependency to the pom file:

```
<dependency>
   <groupId>redis.clients</groupId>
   <artifactId>jedis</artifactId>
   <version>Latest non-RC version</version>
   <type>jar</type>
   <scope>compile</scope>
</dependency>
```

> **Note** For information about the latest version of the Jedis client, go to the Releases page.

ii. Enter the following code in the project based on the Jedis client version and modify the code based on the comments.

> **Note** For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.

○ Jedis client of the latest version ○ Jedis 2.4.0 and earlier ○ A single connection to ApsaraDB for Redis from Jedis (not recommended because the connection cannot be restored if it times out)

```
JedisPoolConfig config = new JedisPoolConfig();
// Specify the maximum number of idle connections based on your business needs. The value cannot exceed the maximum number of connections supported by the ApsaraDB for Redis instance.
config.setMaxIdle(200);
// Specify the maximum number of connections based on your business needs. The value cannot exceed the maximum number of connections supported by the ApsaraDB for Redis instance.
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
// Replace the values of the host and password parameters with the endpoint of the ApsaraDB for Redis instance and the password of the instance account.
String host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
String password = "testaccount:Rp829dlwa";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
try
{
    jedis = pool.getResource();
    /// ... do stuff here ... for example
    jedis.set("foo", "bar");
    System.out.println(jedis.get("foo"));
    jedis.zadd("sose", 0, "car");
    jedis.zadd("sose", 0, "bike");
    System.out.println(jedis.zrange("sose", 0, -1));
}
finally
{
    if(jedis != null)
    {
        jedis.close();
    }
}
/// ... when closing your application:
pool.destroy();
```

```
JedisPoolConfig config = new JedisPoolConfig();
// Specify the maximum number of idle connections based on your business needs. The
value cannot exceed the maximum number of connections supported by the ApsaraDB for
Redis instance.
config.setMaxIdle(200);
// Specify the maximum number of connections based on your business needs. The
value cannot exceed the maximum number of connections supported by the ApsaraDB for
Redis instance.
config.setMaxTotal(300);
config.setTestOnBorrow(false);
config.setTestOnReturn(false);
// Replace the values of the host and password parameters with the endpoint of the
ApsaraDB for Redis instance and the password of the instance account.
String host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
String password = "testaccount:Rp829dlwa";
JedisPool pool = new JedisPool(config, host, 6379, 3000, password);
Jedis jedis = null;
boolean broken = false;
try
{
    jedis = pool.getResource();
    /// ... do stuff here ... for example
    jedis.set("foo", "bar");
    String foobar = jedis.get("foo");
    jedis.zadd("sose", 0, "car");
    jedis.zadd("sose", 0, "bike");
    Set < String > sose = jedis.zrange("sose", 0, -1);
}
catch(Exception e)
{
    broken = true;
}
finally
{
    if(broken)
    {
        pool.returnBrokenResource(jedis);
    }
    else if(jedis != null)
    {
        pool.returnResource(jedis);
    }
}
```

```
import redis.clients.jedis.Jedis;
public class jedistest {
public static void main(String[] args) {
try {
     String host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";// Specify the
endpoint of the ApsaraDB for Redis instance.
     int port = 6379;
     Jedis jedis = new Jedis(host, port);
     //Specify the authentication information.
     jedis.auth("password");//password
     String key = "redis";
     String value = "aliyun-redis";
     //Select a database. Default value: 0.
     jedis.select(1);
     //Specify a key.
     jedis.set(key, value);
     System.out.println("Set Key " + key + " Value: " + value);
     //Obtain the configured key and value.
     String getvalue = jedis.get(key);
     System.out.println("Get Key " + key + " ReturnValue: " + getvalue);
     jedis.quit();
     jedis.close();
}
catch (Exception e) {
 e.printStackTrace();
 }
}
}
```

3. Run the project. Eclipse may return the following result. The result indicates that the client is connected to the ApsaraDB for Redis instance.

```
bar
[bike, car]
```

> **Warning** If specific invalid parameters are set or some features are not properly used, errors may occur. For more information about how to troubleshoot errors, see Common Jedis exceptions in ApsaraDB for Redis.

TairJedis client

TairJedis is an ApsaraDB for Redis client that is developed by Alibaba Cloud based on the Jedis client. TairJedis supports the features of Jedis and ApsaraDB for Redis Enhanced Edition (Tair). For example, TairJedis supports the Tair commands of Tair data structures. For more information, see Extended data structures of ApsaraDB for Redis Enhanced Edition (Tair) and Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

> **Note** For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.

For more information, visit alibabacloud-tairjedis-sdk.

PhpRedis client

1. Download and install the PhpRedis client. For more information, visit phpredis.
2. Enter the following code in a PHP editor and modify the code based on the comments:

> **Note**
>
> - For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
> - If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition (Tair) instances

```
<?php
 /* Replace the values of the host and port parameters with the endpoint and port
number of the ApsaraDB for Redis instance.  */
 $host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
 $port = 6379;
 /* Replace the values of the user and pwd parameters with the username and password of
the instance account. */
 $user = "testaccount";
 $pwd = "Rp829dlwa";
 $redis = new Redis();
 if ($redis->connect($host, $port) == false) {
         die($redis->getLastError());
   }
 if ($redis->auth($pwd) == false) {
         die($redis->getLastError());
  }
  /* You can perform operations on the instance after the connection is established.
For example, you can run the following code to call the set and get methods.  */
 if ($redis->set("foo", "bar") == false) {
         die($redis->getLastError());
 }
 $value = $redis->get("foo");
 echo $value;
 ?>
```

```
<?php
 /* Replace the values of the host and port parameters with the endpoint and port
number of the ApsaraDB for Redis instance.  */
 $host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
 $port = 6379;
 /* Replace the values of the user and pwd parameters with the username and password of
the instance account. */
 $user = "test_username";
 $pwd = "password";
 $redis = new Redis();
 if ($redis->connect($host, $port) == false) {
         die($redis->getLastError());
   }
 if ($redis->auth($pwd) == false) {
         die($redis->getLastError());
  }
  /* You can perform operations on the instance after the connection is established.
For example, you can run the following code to use the TairString data structure.  */
 if ($redis->set("foo", "bar") == false) {
         die($redis->getLastError());
 }
 /* Returns: 1 */
 $redis->rawCommand("CAS", "foo", "bar", "bzz");
 /* Returns: 1 */
 $redis->rawCommand("CAD", "foo", "bzz");
 /* Returns: OK */
 $redis->rawCommand("EXSET", "foo", "200", "VER", "1");
 /* ERR update version is stale */
 $redis->rawCommand("EXSET", "foo", "300", "VER", "10");
 /* Returns : ["OK", " ", VERSION] */
 $redis->rawCommand("EXCAS", "foo", "300", "1");
 ?>
```

3. Run the preceding code to connect to the ApsaraDB for Redis instance.

   For more information, visit phpredis.

> **Note** For more information about how to troubleshoot the `Cannot assign requested address` error when this error is returned to the client, see The "Cannot assign requested address" error occurs when short-lived connections are used to access ApsaraDB for Redis.

redis-py client

1. Download and install the redis-py client. For more information, visit redis-py.
2. Enter the following code in Python 2 and modify the code based on the comments:

Note

- For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
- If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition (Tair) instances

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
import redis
# Replace the values of the host and port parameters with the endpoint and port number
of the ApsaraDB for Redis instance.
host = 'r-bp10noxlhcoim2****.redis.rds.aliyuncs.com'
port = 6379
# Replace the value of the pwd parameter with the password of the instance account.
pwd = 'testaccount:Rp829dlwa'
r = redis.StrictRedis(host=host, port=port, password=pwd)
# You can perform operations on the instance after the connection is established. For
example, you can run the following code to call the set and get methods.
r.set('foo', 'bar');
print r.get('foo')
```

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
import redis
# Replace the values of the host and port parameters with the endpoint and port number
of the ApsaraDB for Redis instance.
host = 'r-bp10noxlhcoim2****.redis.rds.aliyuncs.com'
port = 6379
# Replace the value of the pwd parameter with the password of the instance account.
pwd = 'testaccount:Rp829dlwa'
r = redis.StrictRedis(host=host, port=port, password=pwd)
# You can perform operations on the instance after the connection is established. For
example, you can run the following code to use the TairString data structure.
print(r.execute_command('CAS foo bar bzz'))
print(r.execute_command('CAD foo bzz'))
print(r.execute_command('EXSET foo 200 VER 1'))
try:
    r.execute_command('EXSET foo 300 VER 10')
except:
    print("The attached version is different from the server version, the operation
will fail. ")
print(r.execute_command('EXCAS foo 300 1'))
```

3. Run the preceding code to connect to the ApsaraDB for Redis instance.

Spring Data Redis client

1. Download and install the Spring Data Redis client
2. Enter the following code in a Spring Data Redis editor and modify the code based on the comments:

   Note

   - For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
   - By default, Spring Data Redis 2.0 and later use Lettuce as the driver. If you want to switch over to Jedis, you must exclude the Lettuce dependency and add the Jedis dependency. Example:

     ```
     <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-data-redis</artifactId>
         <exclusions>
             <exclusion>
                 <artifactId>lettuce-core</artifactId>
                 <groupId>io.lettuce</groupId>
             </exclusion>
         </exclusions>
     </dependency>
     <dependency>
         <groupId>redis.clients</groupId>
         <artifactId>jedis</artifactId>
         <version>Latest non-RC version</version>
     </dependency>
     ```

   ○ Spring Data Redis With Jedis (recommended) ○ Spring Data Redis With Lettuce (not recommended)

```
@Bean
     JedisConnectionFactory redisConnectionFactory() {
         RedisStandaloneConfiguration config = new RedisStandaloneConfiguration("host",
port);

         JedisPoolConfig jedisPoolConfig = new JedisPoolConfig();
         // Specify the maximum number of connections based on your business needs. The
value cannot exceed the maximum number of connections supported by the ApsaraDB for
Redis instance.
         jedisPoolConfig.setMaxTotal(30);
         //Specify the maximum number of idle connections based on your business needs.
The value cannot exceed the maximum number of connections supported by the ApsaraDB for
Redis instance.
         jedisPoolConfig.setMaxIdle(20);
         // Close testOn[Borrow|Return] to prevent additional PING commands from being
generated.
         jedisPoolConfig.setTestOnBorrow(false);
         jedisPoolConfig.setTestOnReturn(false);

         JedisClientConfiguration jedisClientConfiguration =
JedisClientConfiguration.builder().usePooling().poolConfig(
             jedisPoolConfig).build();

         return new JedisConnectionFactory(config, jedisClientConfiguration);
     }
```

```
    @Bean
    LettuceConnectionFactory redisConnectionFactory() {
        RedisStandaloneConfiguration config = new RedisStandaloneConfiguration("host",
6379);
        return new LettuceConnectionFactory(config);
    }
```

3. Run the preceding code to connect to the ApsaraDB for Redis instance.

   For more information, visit Spring Data Redis.

C or C++ client

1. Run the following commands to download, compile, and install the C client:

```
git clone https://github.com/redis/hiredis.git
cd hiredis
make
sudo make install
```

2. Enter the following code in a C or C ++ editor and modify the code based on the comments:

Note

- For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
- If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition (Tair) instances

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <hiredis.h>
int main(int argc, char **argv) {
    unsigned int j;
    redisContext *c;
    redisReply *reply;
    if (argc < 4) {
            printf("Usage: example r-bp10noxlhcoim2****.redis.rds.aliyuncs.com 6379
instance_id password\n");
            exit(0);
    }
    const char *hostname = argv[1];
    const int port = atoi(argv[2]);
    const char *instance_id = argv[3];
    const char *password = argv[4];
    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    c = redisConnectWithTimeout(hostname, port, timeout);
    if (c == NULL || c->err) {
    if (c) {
            printf("Connection error: %s\n", c->errstr);
            redisFree(c);
    } else {
        printf("Connection error: can't allocate redis context\n");
    }
    exit(1);
    }
    /* AUTH */
    reply = redisCommand(c, "AUTH %s", password);
    printf("AUTH: %s\n", reply->str);
    freeReplyObject(reply);
    /* PING server */
    reply = redisCommand(c,"PING");
    printf("PING: %s\n", reply->str);
    freeReplyObject(reply);
    /* Set a key */
    reply = redisCommand(c,"SET %s %s", "foo", "hello world");
    printf("SET: %s\n", reply->str);
    freeReplyObject(reply);
    /* Set a key using binary safe API */
    reply = redisCommand(c,"SET %b %b", "bar", (size_t) 3, "hello", (size_t) 5);
    printf("SET (binary API): %s\n", reply->str);
    freeReplyObject(reply);
    /* Try a GET and two INCR */
    reply = redisCommand(c,"GET foo");
    printf("GET foo: %s\n", reply->str);
    freeReplyObject(reply);
    reply = redisCommand(c,"INCR counter");
    printf("INCR counter: %lld\n", reply->integer);
    freeReplyObject(reply);
    /* again ... */
    reply = redisCommand(c,"INCR counter");
    printf("INCR counter: %lld\n", reply->integer);
    freeReplyObject(reply);
    /* Create a list of numbers, from 0 to 9 */
    reply = redisCommand(c,"DEL mylist");
    freeReplyObject(reply);
    for (j = 0; j < 10; j++) {
            char buf[64];
            snprintf(buf,64,"%d",j);
            reply = redisCommand(c,"LPUSH mylist element-%s", buf);
            freeReplyObject(reply);
        }
    /* Let's check what we have inside the list */
    reply = redisCommand(c,"LRANGE mylist 0 -1");
    if (reply->type == REDIS_REPLY_ARRAY) {
            for (j = 0; j < reply->elements; j++) {
            printf("%u) %s\n", j, reply->element[j]->str);
    }
    }
    freeReplyObject(reply);
    /* Disconnects and frees the context */
    redisFree(c);
    return 0;
}
```

```
#include <iostream>
#include <string>
#include <string.h>
#include <hiredis/hiredis.h>
using namespace std;

int main(int argc, char **argv) {
    unsigned int j;
    redisContext *c;
    redisReply *reply;
    if (argc < 3) {
            printf("Usage: example r-bp10noxlhcoim2****.redis.rds.aliyuncs.com 6379
password\n");
            exit(0);
    }
    const char *hostname = argv[1];
    const int port = atoi(argv[2]);
    const char *password = argv[3];
    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    c = redisConnectWithTimeout(hostname, port, timeout);
    if (c == NULL || c->err) {
    if (c) {
            printf("Connection error: %s\n", c->errstr);
            redisFree(c);
    } else {
            printf("Connection error: can't allocate redis context\n");
    }
    exit(1);
    }
    /* AUTH */
    reply = (redisReply *)redisCommand(c, "AUTH %s", password);
    printf("AUTH: %s\n", reply->str);
    freeReplyObject(reply);

    /* PING server */
    reply = (redisReply *)redisCommand(c,"PING");
    printf("PING: %s\n", reply->str);
    freeReplyObject(reply);

    /* The following code provides an example on how to use the TairString data 
structure. */
    reply = (redisReply *)redisCommand(c,"SET foo bar");
    printf("SET: %s\n", reply->str);
    freeReplyObject(reply);

    reply = (redisReply *)redisCommand(c,"CAS foo bar bzz");
    printf("CAS: %lld\n", reply->integer);
    freeReplyObject(reply);

    reply = (redisReply *)redisCommand(c,"CAD foo bzz");
    printf("CAD: %lld\n", reply->integer);
    freeReplyObject(reply);

    /* TairString exstrtype */
    reply = (redisReply *)redisCommand(c,"EXSET foo 200 VER 1");
    printf("EXSET: %s\n", reply->str);
    freeReplyObject(reply);

    /* The attached version is different from the server version, the operation will 
fail */
    reply = (redisReply *)redisCommand(c,"EXSET foo 300 VER 10");
    printf("EXSET: %s\n", reply->str);
    freeReplyObject(reply);

    /* Compare the specified version to update the value, and the update is successful
    when the version in the engine is the same as the specified one */
    reply = (redisReply *)redisCommand(c,"EXCAS foo 300 1");
    if (reply->type == REDIS_REPLY_ARRAY) {
        /* ["OK", "", version], The middle value is an empty string, meaningless when
successful */
        for (j = 0; j < reply->elements; j++) {
            printf("%u) %s\n", j, reply->element[j]->str);
        }
    }
    freeReplyObject(reply);

    /* Disconnects and frees the context */
    redisFree(c);
    return 0;
}
```

3. Compile the code.

```
gcc -o example -g example.c -I /usr/local/include/hiredis -lhiredis
```

4. Perform a test run and connect to the ApsaraDB for Redis instance.

```
 example r-bp10noxlhcoim2****.redis.rds.aliyuncs.com 6379 instance_id password
```

.NET client

Warning If you need to switch over to or select a database from multiple databases in a cluster instance or read/write splitting instance, you must set the cluster_compat_enable parameter to *0* to disable the support for the cluster syntax of open source Redis, and restart the client. Otherwise, the following error message is returned: `Multiple databases are not supported on this server; cannot switch to database` . For more information, see Modify parameters of an instance.

1. Run the following command to download the .NET client:

```
 git clone https://github.com/ServiceStack/ServiceStack.Redis
```

2. Create a .NET project in the .NET client.
3. Add a reference. The referenced file is stored in the library file directory ServiceStack.Redis/lib/tests.
4. Enter the following code in the .NET project and modify the code based on the comments. For more information, visit ServiceStack.Redis.

Note

- For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
- If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition

(Tair) instances

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ServiceStack.Redis;

namespace ServiceStack.Redis.Tests
 {
         class Program
 {
 public static void RedisClientTest()
 {
         // Replace the value of the host parameter with the endpoint of the ApsaraDB
for Redis instance.
         string host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
         // Replace the value of the password parameter with the password of the
instance account.
         string password = "testaccount:Rp829dlwa";
         RedisClient redisClient = new RedisClient(host, 6379, password);
         string key = "test-aliyun";
         string value = "test-aliyun-value";
         redisClient.Set(key, value);
         string listKey = "test-aliyun-list";
         System.Console.WriteLine("set key " + key + " value " + value);
         string getValue =
System.Text.Encoding.Default.GetString(redisClient.Get(key));
         System.Console.WriteLine("get key " + getValue);
         System.Console.Read();
 }
 public static void RedisPoolClientTest()
 {
         string[] testReadWriteHosts = new[] {
         "redis://password@127.0.0.1:6379"/*redis:// Specify a password in the format
of Password@Endpoint:Port number */
 };
 RedisConfig.VerifyMasterConnections = false;// This parameter is required.
 PooledRedisClientManager redisPoolManager = new PooledRedisClientManager(10/*Number of
connection pools*/, 10/*Timeout period of connection pools*/, testReadWriteHosts);
 for (int i = 0; i < 100; i++){
         IRedisClient redisClient = redisPoolManager.GetClient();// Create a client
object.
         RedisNativeClient redisNativeClient = (RedisNativeClient)redisClient;
         redisNativeClient.Client = null;// ApsaraDB for Redis does not support the
CLIENT SETNAME command. Set the client object to null.
 try
 {
         string key = "test-aliyun1111";
         string value = "test-aliyun-value1111";
         redisClient.Set(key, value);
         string listKey = "test-aliyun-list";
         redisClient.AddItemToList(listKey, value);
         System.Console.WriteLine("set key " + key + " value " + value);
```

```
        System.Console.WriteLine("set key " + key + " value " + value);
        string getValue = redisClient.GetValue(key);
        System.Console.WriteLine("get key " + getValue);
        redisClient.Dispose();//
}catch (Exception e)
{
        System.Console.WriteLine(e.Message);
}
}
        System.Console.Read();
}
static void Main(string[] args)
{
        // Use the single-connection mode.
        RedisClientTest();
        // Use the connection pool mode.
        RedisPoolClientTest();
}
}
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ServiceStack.Redis;

namespace NetTestRedis {
class Program {
    public static void RedisClientTest() {
        // Replace the value of the host parameter with the endpoint of the ApsaraDB
for Redis instance.
        string host = "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com";
        // Replace the value of the password parameter with the password of the
instance account.
        string password = "testaccount:Rp829dlwa";
        RedisClient redisClient = new RedisClient(host, 6379, password);
 // The following code provides an example on how to use the TairString data structure.
        System.Console.WriteLine("set : " + redisClient.Custom("set", "foo",
"bal").Text);
        System.Console.WriteLine("CAS : " + redisClient.Custom("CAS", "foo", "bal",
"bzz").Text);
        System.Console.WriteLine("CAD : " + redisClient.Custom("CAD", "foo",
"bzz").Text);
        System.Console.WriteLine("EXSET : " + redisClient.Custom("EXSET", "foo", "200",
"VER", "1").Text);
        try {
            System.Console.WriteLine("EXSET : " + redisClient.Custom("EXSET", "foo",
"300", "VER", "10").Text);
        } catch (Exception ex) {
            Console.WriteLine("ERR : " + ex.ToString());
        }
        var ret = redisClient.Custom("EXCAS", "foo", "300", "1");
        Console.Write("EXCAS : [");
        var values = ret.GetResults();
        // ["OK", "", version], The middle value is an empty string, meaningless when
successful.
        foreach (string item in values) {
            Console.Write(item + " ");
        }
        Console.Write("]");
    }
    static void Main(string[] args) {
        // Use the single-connection mode.
        RedisClientTest();
    }
}
}
```

node-redis client

1. Download and install the node-redis client.

```
npm install hiredis redis
```

2. Enter the following code in the node-redis client and modify the code based on the comments:

> **Note**
>
> - For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
> - If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition (Tair) instances

```
var redis = require("redis"),
// Specify the port number and endpoint of the ApsaraDB for Redis instance.
client = redis.createClient(6379, "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com",
{detect_buffers: true});
// Specify the password of the instance account.
client.auth("testaccount:Rp829dlwa", redis.print)
// Write data to the instance.
client.set("key", "OK");
// Query data from the ApsaraDB for Redis instance. The returned data is of the STRING
type.
client.get("key", function (err, reply) {
console.log(reply.toString()); // print `OK`
});
// If the input parameter is a buffer, the returned value is also a buffer.
client.get(new Buffer("key"), function (err, reply) {
console.log(reply.toString()); // print `<Buffer 4f 4b>`
});
client.quit();
```

```
var redis = require("redis");
// Replace the values of the host and port parameters with the endpoint and port number
of the ApsaraDB for Redis instance.
var client = redis.createClient({host : 'r-bp10noxlhcoim2****.redis.rds.aliyuncs.com',
port : 6379});
// Specify the password of the instance account.
client.auth("testaccount:Rp829dlwa", redis.print)

client.set("foo", "bar");
client.get("foo", function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log(reply.toString()); // print `OK`
    }
});
// If the input parameter is a buffer, the returned value is also a buffer.
client.get(new Buffer("foo"), function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log(reply.toString()); // print `<Buffer 4f 4b>`
    }
});

// The following code provides an example on how to use the TairString data structure.
client.sendCommand('CAS', ['foo', 'bar', 'bzz'], function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log('CAS : %s', reply.toString());
    }
});

client.sendCommand('CAD', ['foo', 'bzz'], function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log('CAD : %s', reply.toString());
    }
});

client.sendCommand('EXSET', ['foo', '200', 'VER', '1'], function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log('EXSET : %s', reply.toString());
    }
});

client.sendCommand('EXSET', ['foo', '300', 'VER', '10'], function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log('EXSET : %s', reply.toString());
    }
});

client.sendCommand('EXCAS', ['foo', '300', '1'], function (err, reply) {
    if (err) {
        console.log(err);
    } else {
        console.log('EXCAS : %s', reply.toString()); // print `<Buffer 4f 4b>`
    }
});

client.quit();
```

3. Run the preceding code to connect to the ApsaraDB for Redis instance.

C# client StackExchange.Redis

Warning If you need to switch or select a database from multiple databases in a cluster instance or read/write splitting instance, you must set the cluster_compat_enable parameter to *0* to disable the support for the cluster syntax of open source Redis, and then restart the client. Otherwise, the following error message is returned: `RedisCommandException: Multiple databases are not supported on this server; cannot switch to database: 1` . For more information, see Modify parameters of an instance.

1. Download and install the StackExchange.Redis client.
2. Enter the following code in the client and modify the code based on the comments:

   Note

   - For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
   - If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).
   - ConfigurationOptions is a core object of the StackExchange.Redis client. ConfigurationOptions is shared and reused by the client. The singleton pattern is recommended. For more information about parameter settings, see Configuration Options.
   - GetDatabase() returns a lightweight object. You can obtain this object from the ConnectionMultiplexer object.

     ```
     redisConn = getRedisConn();
     var db = redisConn.GetDatabase();
     ```

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition (Tair) instances

```
using StackExchange.Redis;
 // Specify the endpoint, port number, and password of the ApsaraDB for Redis instance.
 private static ConfigurationOptions configurationOptions =
ConfigurationOptions.Parse("r-
bp10noxlhcoim2****.redis.rds.aliyuncs.com:6379,password=testaccount:Rp829dlwa,connectTime

  //the lock for singleton
 private static readonly object Locker = new object();
  //singleton
 private static ConnectionMultiplexer redisConn;
 //singleton
 public static ConnectionMultiplexer getRedisConn()
 {
     if (redisConn == null)
     {
         lock (Locker)
         {
             if (redisConn == null || !redisConn.IsConnected)
             {
                 redisConn = ConnectionMultiplexer.Connect(configurationOptions);
             }
         }
     }
     return redisConn;
 }
```

```
using System;
using StackExchange.Redis;

namespace CSharpTestRedis
{
    class Program
    {
        // Specify the endpoint, port number, and password of the ApsaraDB for Redis
instance.
  private static ConfigurationOptions connDCS = ConfigurationOptions.Parse("r-
bp10noxlhcoim2****.redis.rds.aliyuncs.com:6379,password=testaccount:Rp829dlwa");
        //the lock for singleton
        private static readonly object Locker = new object();
        //singleton
        private static ConnectionMultiplexer redisConn;
        //singleton
        public static ConnectionMultiplexer getRedisConn()
        {
            if (redisConn == null)
            {
                lock (Locker)
                {
                    if (redisConn == null || !redisConn.IsConnected)
                    {
                        redisConn = ConnectionMultiplexer.Connect(connDCS);
                    }
                }
            }
            return redisConn;
        }
        static void Main(string[] args)
        {
            redisConn = getRedisConn();
            var db = redisConn.GetDatabase();

            var ret = db.Execute("set", "foo", "bal");
            Console.WriteLine("set " + ret);
            ret = db.Execute("CAS", "foo", "bal", "bzz");
            Console.WriteLine("CAS " + ret);
            ret = db.Execute("CAD", "foo", "bzz");
            Console.WriteLine("CAD " + ret);
            ret = db.Execute("EXSET", "foo", "200", "VER", "1");
            Console.WriteLine("EXSET " + ret);

            try {
                ret = db.Execute("EXSET", "foo", "300", "VER", "10");
                Console.WriteLine("EXSET " + ret);
            } catch (Exception ex) {
                Console.WriteLine("ERR : " + ex.ToString());
            }
            // ["OK", "", version], The middle value is an empty string, meaningless
when successful.
            db.Execute("EXCAS", "foo", "300", "1");
            Console.WriteLine("END");
        }
    }
}
```

3. Use one of the following common data structures to perform operations. These data structures are slightly different from those of native Redis APIs.

   ○ String ○ Hash ○ List ○ Set ○ Sorted Set

```
//set get
string strKey = "hello";
string strValue = "world";
bool setResult = db.StringSet(strKey, strValue);
Console.WriteLine("set " + strKey + " " + strValue + ", result is " + setResult);
//incr
string counterKey = "counter";
long counterValue = db.StringIncrement(counterKey);
Console.WriteLine("incr " + counterKey + ", result is " + counterValue);
//expire
db.KeyExpire(strKey, new TimeSpan(0, 0, 5));
Thread.Sleep(5 * 1000);
Console.WriteLine("expire " + strKey + ", after 5 seconds, value is " +
db.StringGet(strKey));
//mset mget
KeyValuePair<RedisKey, RedisValue> kv1 = new KeyValuePair<RedisKey, RedisValue>("key1",
"value1");
KeyValuePair<RedisKey, RedisValue> kv2 = new KeyValuePair<RedisKey, RedisValue>("key2",
"value2");
db.StringSet(new KeyValuePair<RedisKey, RedisValue>[] {kv1,kv2});
RedisValue[] values = db.StringGet(new RedisKey[] {kv1.Key, kv2.Key});
Console.WriteLine("mget " + kv1.Key.ToString() + " " + kv2.Key.ToString() + ", result
is " + values[0] + "&&" + values[1]);
```

```
string hashKey = "myhash";
//hset
db.HashSet(hashKey,"f1","v1");
db.HashSet(hashKey,"f2", "v2");
HashEntry[] values = db.HashGetAll(hashKey);
//hgetall
Console.Write("hgetall " + hashKey + ", result is");
for (int i = 0; i < values.Length;i++)
{
  HashEntry hashEntry = values[i];
  Console.Write(" " + hashEntry.Name.ToString() + " " + hashEntry.Value.ToString());
}
Console.WriteLine();
```

```
//list key
string listKey = "myList";
//rpush
db.ListRightPush(listKey, "a");
db.ListRightPush(listKey, "b");
db.ListRightPush(listKey, "c");
//lrange
RedisValue[] values = db.ListRange(listKey, 0, -1);
Console.Write("lrange " + listKey + " 0 -1, result is ");
for (int i = 0; i < values.Length; i++)
{
 Console.Write(values[i] + " ");
}
Console.WriteLine();
```

```
//set key
string setKey = "mySet";
//sadd
db.SetAdd(setKey, "a");
db.SetAdd(setKey, "b");
db.SetAdd(setKey, "c");
//sismember
bool isContains = db.SetContains(setKey, "a");
Console.WriteLine("set " + setKey + " contains a is " + isContains );
```

```
string sortedSetKey = "myZset";
//sadd
db.SortedSetAdd(sortedSetKey, "xiaoming", 85);
db.SortedSetAdd(sortedSetKey, "xiaohong", 100);
db.SortedSetAdd(sortedSetKey, "xiaofei", 62);
db.SortedSetAdd(sortedSetKey, "xiaotang", 73);
//zrevrangebyscore
RedisValue[] names = db.SortedSetRangeByRank(sortedSetKey, 0, 2, Order.Ascending);
Console.Write("zrevrangebyscore " + sortedSetKey + " 0 2, result is ");
for (int i = 0; i < names.Length; i++)
{
  Console.Write(names[i] + " ");
}
Console.WriteLine();
```

go-redis client

Note

- For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.
- If you use a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair), you can click the preceding **Code for ApsaraDB for Redis Enhanced Edition (Tair) instances** tab to view the example on how to use the TairString data structure. For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about the data structures, see Commands supported by extended data structures of ApsaraDB for Redis Enhanced Edition (Tair).

○ Code for regular ApsaraDB for Redis instances ○ Code for ApsaraDB for Redis Enhanced Edition (Tair) instances

```
package main

import (
 "github.com/go-redis/redis"
 "fmt"
)

func ExampleClient() {
 client := redis.NewClient(&redis.Options{
        // Specify the endpoint and port number of the ApsaraDB for Redis instance.
  Addr:     "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com:6379",
        // Specify the password of the instance account.
  Password: "testaccount:Rp829dlwa",
  DB:       0,  // use default DB
 })
    // The following code provides an example on how to call the set and get methods.
 err := client.Set("foo", "bar", 0).Err()
 if err != nil {
  panic(err)
 }

 val, err := client.Get("foo").Result()
 if err != nil {
  panic(err)
 }
 fmt.Println("set : foo -> ", val)
}

func main() {
 ExampleClient()
}
```

```
package main

import (
 "github.com/go-redis/redis"
 "fmt"
)

func ExampleClient() {
 client := redis.NewClient(&redis.Options{
        // Specify the endpoint and port number of the ApsaraDB for Redis instance.
  Addr:     "r-bp10noxlhcoim2****.redis.rds.aliyuncs.com:6379",
        // Specify the password of the instance account.
  Password: "testaccount:Rp829dlwa", // no password set
  DB:       0,  // use default DB
 })

 err := client.Set("foo", "bar", 0).Err()
 if err != nil {
  panic(err)
 }

 val, err := client.Get("foo").Result()
 if err != nil {
  panic(err)
 }
 fmt.Println("set : foo -&gt; ", val)

 // The following code provides an example on how to use the TairString data structure.
 res, err := client.Do("CAS", "foo", "bar", "bzz").Result()
 fmt.Println("CAS : ", res)

 res, err = client.Do("CAD", "foo", "bzz").Result()
 fmt.Println("CAD : ", res)

 res, err = client.Do("EXSET", "foo", "200", "VER", "1").Result()
 fmt.Println("EXSET : ", res)

 res, err = client.Do("EXSET", "foo", "300", "VER", "10").Result()
 if err != nil {
  fmt.Println(err)
 }
 fmt.Println("EXSET : ", res)

 res, err = client.Do("EXCAS", "foo", "300", "1").Result()
 fmt.Println("EXCAS : ", res)
}

func main() {
 ExampleClient()
}
```

Lettuce client

The Lettuce client supports synchronous and asynchronous communication based on comprehensive Redis APIs. The Lettuce client does not automatically reconnect to an instance after multiple requests time out. If failures occur in ApsaraDB for Redis and cause failovers for proxy nodes or data nodes, a connection timeout may occur. This may result in the failure to reconnect to ApsaraDB for Redis. To prevent such risks, we recommend that you use a Jedis client to connect to ApsaraDB for Redis instances. For more information, see Jedis client.

> ? **Note** For information about how to obtain the endpoint of the ApsaraDB for Redis instance and the password of the instance account, see Obtain connection information.

For more information, visit lettuce-core.

Related information

- Troubleshooting for connection issues in ApsaraDB for Redis
- Retry mechanisms for Redis clients
- Alert settings
- View monitoring data

7.5. Use a client to connect to an ApsaraDB for Redis instance that has SSL encryption enabled

When you connect to an ApsaraDB for Redis instance by using a client, you can enable the SSL encryption feature to enhance data security and ensure data integrity. You can connect to an ApsaraDB for Redis instance by using clients of different programming languages that are compatible with the Redis protocol. This topic describes sample code of common programming languages.

Prerequisites

SSL encryption is enabled for an ApsaraDB for Redis instance. For more information, see Configure SSL encryption.

Precautions

- By default, cluster or read/write splitting instances run in proxy mode. In this mode, you can access an ApsaraDB for Redis instance by using the endpoint of a proxy node in the instance in the same way as you access an ApsaraDB for Redis standard instance. For more information about cluster and read/write splitting instances, see Cluster master-replica instances or Read/write splitting instances.

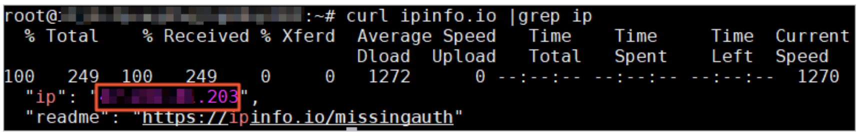
  > ? **Note** If you use a private endpoint to connect to an ApsaraDB for Redis instance, you can connect to the instance in the same way as you connect to an open source Redis cluster. For more information about private endpoints, see Enable the direct connection mode.

- If password-free access is enabled for an ApsaraDB for Redis instance deployed in a VPC, a client in the same VPC as the instance can connect to the instance without using passwords. For more information, see Enable password-free access.

Preparations

1. Perform the following operations based on the type of host on which a client is deployed.

Host	Operation
ECS instance (recommended)	i. Make sure that the Elastic Compute Service (ECS) instance and the ApsaraDB for Redis instance belong to the same virtual private cloud (VPC). If the VPC IDs of these two instances are the same, the instances belong to the same VPC. **Note** ■ If the instances are deployed in different VPCs, you can change the VPC to which the ECS instance belongs. For more information, see Change the VPC of an ECS instance. ■ The network types of the ECS instance and the ApsaraDB for Redis instance may be different. For example, the ECS instance belongs to the classic network and the ApsaraDB for Redis instance belongs to a VPC. For information about how to connect to an ApsaraDB for Redis instance from an ECS instance when the instances are deployed in different network types, see Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks. ii. Obtain the internal IP address of the ECS instance. For more information, see Network FAQ. iii. Add the internal IP address of the ECS instance to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

On-premises device	i. By default, only internal endpoints are available for ApsaraDB for Redis instances. If you want to connect to an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance.
	ii. Run the **curl ipinfo.io |grep ip** command on your on-premises device to obtain its public IP address. The following figure shows a sample command output.
	
	`root@[redacted]:~# curl ipinfo.io |grep ip`
	`  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current`
	`                                 Dload  Upload   Total   Spent    Left  Speed`
	`100   249  100   249    0     0   1272      0 --:--:-- --:--:-- --:--:--  1270`
	`  "ip": "[redacted].203",`
	`  "readme": "https://ipinfo.io/missingauth"`
	Note If your on-premises device runs a Windows operating system, visit ipinfo to obtain the public IP address.
	iii. Add the public IP address of your on-premises device to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

2. Obtain the following information and use the information in client code of different programming languages.

Item	Description

Item	Description
Instance endpoint	ApsaraDB for Redis instances support multiple endpoint types. We recommend that you use VPCs for higher security and lower network latency. For more information, see View endpoints.
Port number	The default port number is 6379. You can also use a custom port number. For more information, see Change the endpoint or port number of an ApsaraDB for Redis instance.
Instance account (this information is optional for specific clients)	By default, an ApsaraDB for Redis instance has a database account that is named after the instance ID. Example: r-bp10noxlhcoim2****. You can create another database account and grant the required permissions to the account. For more information, see Create and manage database accounts.
Password	The password format varies with the selected account: ○ If you use the default account whose username is the same as the instance ID, enter only the password. ○ If you use a custom account, enter a password in the `<user>:<password>` format. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa`, you must enter `testaccount:Rp829dlwa` as the database password. Note ○ If you use a management tool such as Redis Desktop Manager (RDM) to connect to the ApsaraDB for Redis instance, enter a password in the format of `<user>:<password>`. ○ If you forget your password, you can reset it. For more information, see Change or reset the password.

3. Download the certificate authority (CA) certificate. For more information, see Configure SSL encryption.

Java

The following sample code uses the Jedis 3.6.0 client. We recommend that you use the latest version of the client. For more information, visit Jedis.

> Note You must modify your code based on comments. For information about how to obtain the endpoint, port number, and password of an ApsaraDB for Redis instance, see Step 2 of the Preparations section.

```
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;
public class JedisSSLTest {
    private static SSLSocketFactory createTrustStoreSSLSocketFactory(String jksFile) throws
Exception {
        KeyStore trustStore = KeyStore.getInstance("jks");
        InputStream inputStream = null;
        try {
            inputStream = new FileInputStream(jksFile);
            trustStore.load(inputStream, null);
        } finally {
            inputStream.close();
        }
        TrustManagerFactory trustManagerFactory = TrustManagerFactory.getInstance("PKIX");
        trustManagerFactory.init(trustStore);
        TrustManager[] trustManagers = trustManagerFactory.getTrustManagers();
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, trustManagers, new SecureRandom());
        return sslContext.getSocketFactory();
    }
    public static void main(String[] args) throws Exception {
        // ApsaraDB-CA-Chain.jks is the name of the CA certificate file.
        final SSLSocketFactory sslSocketFactory = createTrustStoreSSLSocketFactory("ApsaraD
B-CA-Chain.jks");
        // The endpoint, port number, timeout period, and password of the instance are incl
uded in the configurations of a connection pool.
        JedisPool pool = new JedisPool(new GenericObjectPoolConfig(), "r-bp1zxszhcgatnx****
.redis.rds.aliyuncs.com",
            6379, 2000, "redistest:Test1234", 0, true, sslSocketFactory, null, null);
        try (Jedis jedis = pool.getResource()) {
            jedis.set("key", "value");
            System.out.println(jedis.get("key"));
        }
    }
}
```

Python

The following sample code uses the redis-py client. We recommend that you use the latest version of the client. For more information, visit redis-py.

Note You must modify your code based on comments. For information about how to obtain the endpoint, port number, and password of an ApsaraDB for Redis instance, see Step 2 of the Preparations section.

◉ Regular connection ○ Pool connection

```
#!/bin/python
import redis

# Specify connection information. Replace the values of host, port, and password with the
endpoint, port number, and password of the instance, respectively.
# ApsaraDB-CA-Chain.pem is the name of the CA certificate file.
client = redis.Redis(host="r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com", port=6379,
                     password="redistest:Test1234", ssl=True,
                    ssl_cert_reqs="required", ssl_ca_certs="ApsaraDB-CA-Chain.pem")

client.set("hello", "world")
print client.get("hello")
```

```
#!/bin/python
import redis

# Specify a connection pool. Replace the values of host, port, and password with the
endpoint, port number, and password of the instance, respectively.
# ApsaraDB-CA-Chain.pem is the name of the CA certificate file.
pool = redis.ConnectionPool(connection_class=redis.connection.SSLConnection,
max_connections=100,
                            host="r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com", port=6379,
password="redistest:Test1234",
                            ssl_cert_reqs=True, ssl_ca_certs="ApsaraDB-CA-Chain.pem")
client = redis.Redis(connection_pool=pool)
client.set("hi", "redis")
print client.get("hi")
```

PHP

The following sample code uses the Predis client. We recommend that you use the latest version of the client. For more information, visit Predis. If you use the PhpRedis client, you can reference SSL/TLS with certification file to connect to an instance. For more information about PhpRedis, visit PhpRedis.

Note You must modify your code based on comments. For information about how to obtain the endpoint, port number, and password of an ApsaraDB for Redis instance, see Step 2 of the Preparations section.

```
<?php
require __DIR__.'/predis/autoload.php';
/* Specify connection information. Replace the values of host, port, and password with the
endpoint, port number, and password of the instance, respectively.
ApsaraDB-CA-Chain.pem is the name of the CA certificate file. */
$client = new Predis\Client([
    'scheme' => 'tls',
    'host'   => 'r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com',
    'port'   => 6379,
    'password' => 'redistest:Test1234',
    'ssl'    => ['cafile' => 'ApsaraDB-CA-Chain.pem', 'verify_peer' => true],
]);
/* Replace the endpoint and the port number in the following sample code. */
//$client = new Predis\Client('tls://r-bp1zxszhcgatnx****.redis.rds.aliyuncs.com:6379?ssl[c
afile]=ApsaraDB-CA-Chain.pem&ssl[verify_peer]=1');
$client->set("hello", "world");
print $client->get("hello")."\n";
?>
```

C#

The following sample code uses the StackExchange.Redis client. We recommend that you use the latest version of the client. For more information, visit StackExchange.Redis.

Note You must modify your code based on comments. For information about how to obtain the endpoint, port number, and password of an ApsaraDB for Redis instance, see Step 2 of the Preparations section.

```
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;
using StackExchange.Redis;
namespace SSLTest
{
    class Program
    {
        private static bool CheckServerCertificate(object sender, X509Certificate certificate,
            X509Chain chain, SslPolicyErrors sslPolicyErrors)
        {
            var ca = new X509Certificate2(
                "/your path/ApsaraDB-CA-Chain/ApsaraDB-CA-Chain.pem");
            return chain.ChainElements
                .Cast<X509ChainElement>()
                .Any(x => x.Certificate.Thumbprint == ca.Thumbprint);
        }
        static void Main(string[] args)
        {
          // Specify connection information. Replace the values of host, port, and password with the endpoint, port number, and password of the instance, respectively.
          // ApsaraDB-CA-Chain.pem is the name of the CA certificate file.
            ConfigurationOptions config = new ConfigurationOptions()
            {
                EndPoints = {"r-bp10q23zyfriodu*****.redis.rds.aliyuncs.com:6379"},
                Password = "redistest:Test1234",
                Ssl = true,
            };
            config.CertificateValidation += CheckServerCertificate;
            using (var conn = ConnectionMultiplexer.Connect(config))
            {
                Console.WriteLine("connected");
                var db = conn.GetDatabase();
                db.StringSet("hello", "world");
                Console.WriteLine(db.StringGet("hello"));
            }
        }
    }
}
```

Related information

- Use a client to connect to an ApsaraDB for Redis instance
- Retry mechanisms for Redis clients
- Use redis-cli to connect to an ApsaraDB for Redis instance
- Log on to an ApsaraDB for Redis instance by using DMS

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

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

Prerequisites

- The direct connection mode is enabled for the cluster instance. For more information, see Enable the direct connection mode.
- The client IP address is added to a whitelist of the ApsaraDB for Redis cluster instance. For more information, see Step 2: Configure whitelists.
- A client that supports Redis Cluster is used, such as Jedis and PhpRedis.

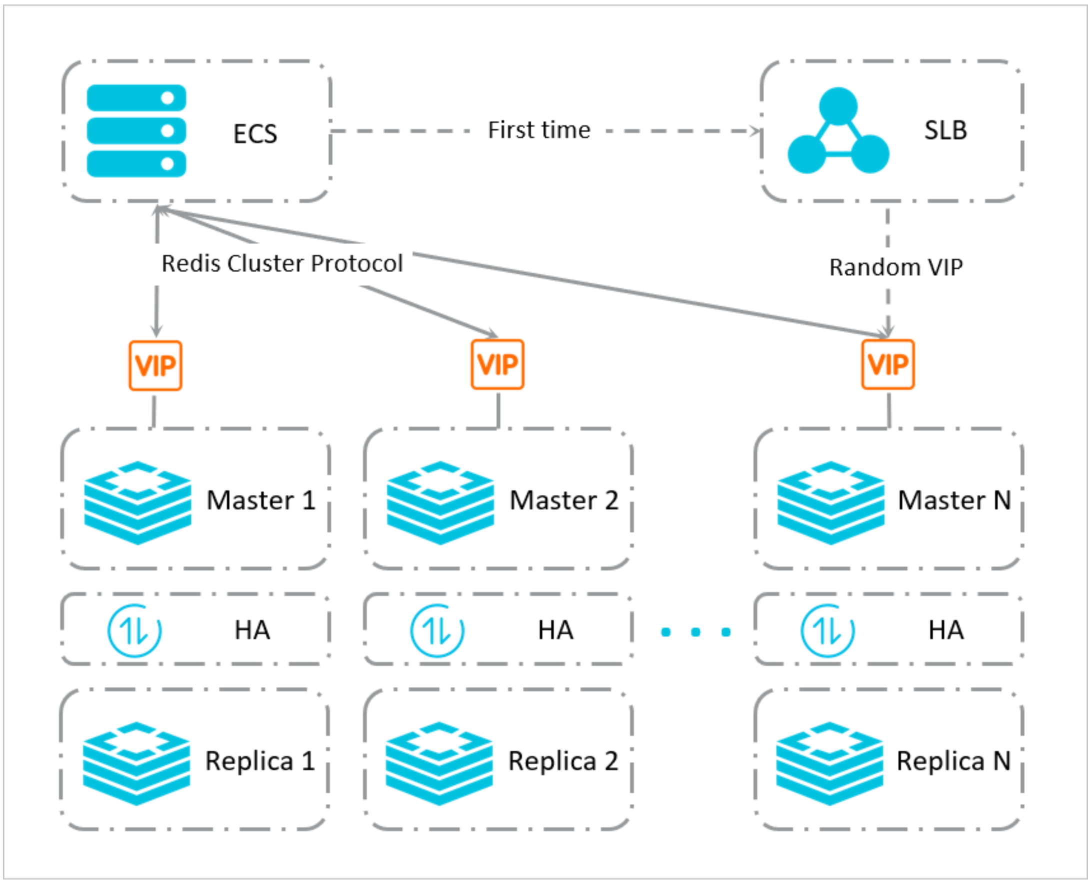
  > Note
  >
  > - If you use a client that does not support Redis Cluster, you may fail to obtain data because the client cannot redirect your request to the correct shard.
  > - Jedis uses the JedisCluster class to support Redis Cluster. For more information, see Class JedisCluster.
  > - You can obtain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

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

Context

When you enable the direct connection mode, ApsaraDB for Redis allocates a virtual IP (VIP) address to the master node of each data shard in the ApsaraDB for Redis cluster instance. 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 instance. The client can use this VIP address to manage the data of the ApsaraDB for Redis cluster instance over the Redis Cluster protocol. The following figure shows the service architecture of an ApsaraDB for Redis cluster instance in the direct connection mode.

Architecture of a cluster instance in direct connection mode



Precautions

- ApsaraDB for Redis instances of different architectures provide different support for native Redis commands. For example, the cluster architecture does not support the **SWAPDB** command, and has restrictions on Lua scripts. For more information,see Limits on commands supported by cluster instances.
- In direct connection mode, when you change the configurations of an instance, slot migration is performed during the process. In this case, the client may prompt error messages such as `MOVED` and `TRYAGAIN` when the client accesses the slots being migrated. For more information about how to change the configurations of an instance, see Change the configurations of an instance. To ensure the successful execution of the request, configure a retry mechanism for the client. For more information, see Retry mechanisms for Redis clients.
- In direct connection mode, you can use the **SELECT** command to switch databases. However, specific Redis Cluster clients do not support the **SELECT** command, such as StackExchange.redis. If you use StackExchange.redis, you can only use database 0.
- Private endpoints can be used to access ApsaraDB for Redis cluster instances only over the Alibaba Cloud internal network. When you use the private endpoint of a cluster instance to access the instance, password-free access and account and password authentication are supported.

Sample code for connecting Jedis to a cluster instance

Note For more information about how to use Jedis, visit GitHub.

○ Use the default connection pool ○ Use a custom 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[]){

         // Specify the private endpoint that you applied for the cluster instance.
         String host = "r-bp1xxxxxxxxxxxx.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();
     }
 }
```

```
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();

     public static void main(String args[]){
         JedisPoolConfig config = new JedisPoolConfig();
         // Specify the maximum number of idle connections. In direct connection mode, the
client directly connects to a shard of a cluster instance. Therefore, the following
requirement must be met: Number of clients × Value of MaxTotal < Maximum number of
connections to a single shard.
         // Set the maximum number of connections to a shard of an ApsaraDB for Redis
Community Edition instance to 10000 and set the maximum number of connections to a shard of
an ApsaraDB for Redis Enhanced Edition (Tair) instance to 30000.
         config.setMaxTotal(30);
         // Specify the maximum number of idle connections based on your business needs.
         config.setMaxIdle(20);
         config.setTestOnBorrow(false);
         config.setTestOnReturn(false);

         // Specify the private endpoint that you applied for the cluster instance.
         String host = "r-bp1xxxxxxxxxxxx.redis.rds.aliyuncs.com";
         int port = 6379;
         // Specify 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);
     }
 }
```

Sample code for connecting PhpRedis to a cluster instance

> **Note** For more information about how to use PhpRedis, visit GitHub.

○ Sample code for connecting PhpRedis to the cluster instance

```
<?php
 // Specify the private endpoint and the port number to connect to the cluster instance.
 $array = ['r-bp1xxxxxxxxxxxx.redis.rds.aliyuncs.com:6379'];
 // Specify 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);

 // Display 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;
 ?>
```

Sample code for connecting Spring Data Redis to a cluster instance

Note For more information about how to use Spring Data Redis, visit Spring.

○ Spring Data Redis With Jedis ○ Spring Data Redis With Lettuce (Not recommended)

```
@Bean
     JedisConnectionFactory redisConnectionFactory() {
         List<String> clusterNodes = Arrays.asList("host1:port1", "host2:port2",
"host3:port3");
         RedisClusterConfiguration redisClusterConfiguration = new
RedisClusterConfiguration(clusterNodes);
         redisClusterConfiguration.setPassword("xxx");

         JedisPoolConfig jedisPoolConfig = new JedisPoolConfig();
         // Specify the maximum number of idle connections. In direct connection mode, the
client directly connects to a shard of a cluster instance. Therefore, the following
requirement must be met: Number of clients × Value of MaxTotal < Maximum number of
connections to a single shard.
         // Set the maximum number of connections to a shard of an ApsaraDB for Redis
Community Edition instance to 10000 and set the maximum number of connections to a shard of
an ApsaraDB for Redis Enhanced Edition (Tair) instance to 30000.
         jedisPoolConfig.setMaxTotal(30);
         // Specify the maximum number of idle connections based on your business needs.
         jedisPoolConfig.setMaxIdle(20);
         // Disable testOn[Borrow|Return] to prevent generating additional ping commands.
         jedisPoolConfig.setTestOnBorrow(false);
         jedisPoolConfig.setTestOnReturn(false);

         return new JedisConnectionFactory(redisClusterConfiguration, jedisPoolConfig);
     }
```

```
@Bean
     public LettuceConnectionFactory redisConnectionFactory() {
         List<String> clusterNodes = Arrays.asList("host1:port1", "host2:port2",
"host3:port3");
         RedisClusterConfiguration redisClusterConfiguration = new
RedisClusterConfiguration(clusterNodes);
         redisClusterConfiguration.setPassword("xxx");

         ClusterTopologyRefreshOptions topologyRefreshOptions =
ClusterTopologyRefreshOptions.builder()
             .enablePeriodicRefresh(Duration.ofSeconds(15))                    // Specify
the interval at which the topology of the cluster instance is refreshed. We recommend that
you set the interval to 15 seconds. If you set it to a smaller value, a large number of
calls are made to the nodes of the cluster instance. This degrades the cluster instance
performance.
             .dynamicRefreshSources(false)                                     // Specify
whether to use the IP addresses of the nodes obtained from the topology as nodes to be
called to refresh the topology of the cluster instance. If you connect to an ApsaraDB for
Redis instance, you must set the parameter to false.
             .enableAllAdaptiveRefreshTriggers()                               // Enable
all triggers to adaptively refresh the topology. After adaptive refresh is enabled, the
topology of the cluster instance is automatically refreshed each time a runtime event such
as a MOVED redirection occurs on the cluster instance.
             .adaptiveRefreshTriggersTimeout(Duration.ofSeconds(15)).build();   // Specify
the time period during which the topology of the cluster instance can be refreshed only
once. This prevents the topology from being frequently refreshed.

         LettuceClientConfiguration lettuceClientConfiguration =
LettuceClientConfiguration.builder().
             clientOptions(ClusterClientOptions.builder()
                 .validateClusterNodeMembership(false)
                 .topologyRefreshOptions(topologyRefreshOptions).build()).build();

         return new LettuceConnectionFactory(redisClusterConfiguration,
lettuceClientConfiguration);
     }
```

Sample code for connecting Lettuce to a cluster instance

The Lettuce client supports synchronous and asynchronous communication by using Redis APIs. The Lettuce client does not automatically reconnect to an instance after multiple requests time out. If failures occur in ApsaraDB for Redis and cause failovers for proxy nodes or data nodes, a connection timeout may occur. This may result in the failure to reconnect to ApsaraDB for Redis. To prevent this issue, we recommend that you use other clients.

Note For more information about how to use Lettuce, visit GitHub.

○ Code for connecting Lettuce to an cluster instance

```
public class ClusterDemo {
    public static void main(String[] args) throws Exception {
        String host = "r-bp1xxxxxxxxxxxx.redis.rds.aliyuncs.com";
        int port = 30001;
        String password = "xxxx";

        RedisURI redisURI = RedisURI.Builder.redis(host)
            .withPort(port)
            .withPassword(password)
            .build();

        ClusterTopologyRefreshOptions refreshOptions =
ClusterTopologyRefreshOptions.builder()
            .enablePeriodicRefresh(Duration.ofSeconds(15))                     // Specify
the interval at which the topology of the cluster instance is refreshed. We recommend that
you set the interval to 15 seconds. If you set it to a smaller value, a large number of
calls are made to the nodes of the cluster instance. This degrades the cluster instance
performance.
            .dynamicRefreshSources(false)                                      // Specify
whether to use the IP addresses of the nodes obtained from the topology as nodes to be
called to refresh the topology of the cluster instance. If you connect to an ApsaraDB for
Redis instance, you must set the parameter to false.
            .enableAllAdaptiveRefreshTriggers()                                // Enable
all triggers to adaptively refresh the topology. After adaptive refresh is enabled, the
topology of the cluster instance is automatically refreshed each time a runtime event such
as a MOVED redirection occurs on the cluster instance.
            .adaptiveRefreshTriggersTimeout(Duration.ofSeconds(15)).build();   // Specify
the time period during which the topology of the cluster instance can be refreshed only
once. This prevents the topology from being frequently refreshed.

        RedisClusterClient redisClient = RedisClusterClient.create(redisURI);
        redisClient.setOptions(ClusterClientOptions.builder()
            .socketOptions(SocketOptions.builder()
                .keepAlive(true) // Set the keepAlive parameter to true.
                .build())
            .validateClusterNodeMembership(false)
            .topologyRefreshOptions(refreshOptions).build());

        StatefulRedisClusterConnection<String, String> connection = redisClient.connect();
        connection.sync().set("key", "value");
    }
 }
```

FAQ

- Q: What do I do if the `Connection to xxx not allowed. This partition is not known in the cluster view.` error occurs when I use Lettuce to connect to an ApsaraDB for Redis cluster instance?

  A: Specify the refreshOptions parameter. For more information, see Sample code for connecting Lettuce to a cluster instance.

7.7. Use the Sentinel-compatible mode to connect to an ApsaraDB for Redis instance

ApsaraDB for Redis provides the Sentinel-compatible mode to increase compatibility and reduce code modifications. After you enable this mode, clients can connect to an ApsaraDB for Redis instance in the same manner as they would connect to the native Redis Sentinel.

ApsaraDB for Redis uses the high-availability (HA) component developed by Alibaba Cloud instead of Redis Sentinel. For more information, see Features.

Overview of Redis Sentinel

Redis Sentinel provides open source Redis with features such as master and replica monitoring, fault alerting, and automatic failover. Redis Sentinel is used in many business scenarios that involve self-managed Redis databases and require high reliability. To facilitate the migration of Redis databases to the cloud in such scenarios, Alibaba Cloud provides the Sentinel-compatible mode.

After you enable the Sentinel-compatible mode, you can run the Sentinel commands described in the following table.

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

> **Note** For more information about Sentinel commands that are supported by different engine versions, see Sentinel command group.

Prerequisites

- The engine version of your ApsaraDB for Redis instance is 4.0 or 5.0.
- The ApsaraDB for Redis instance is deployed in a virtual private cloud (VPC).

  > **Note** If the ApsaraDB for Redis instance runs in the classic network, switch the network type to VPC. For more information, see Change the network type from classic network to VPC.

- The internal IP address of your Elastic Compute Service (ECS) instance or the public IP address of your on-premises host is added to an IP address whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.

Procedure

1.
2. In the left-side navigation pane of the instance details page, click **System Parameters**.
3. On the System Parameters page, find the #no_loose_sentinel-enabled parameter and click **Modify** in the **Actions** column.
4. In the dialog box that appears, set the parameter to *yes* and click **OK**.

> **Note** A value of `no` indicates that the Sentinel-compatible mode is disabled. This is the default value. A value of `yes` indicates that the Sentinel-compatible mode is enabled. For more information about the supported parameters, see Supported parameters.

Connect to an ApsaraDB for Redis instance in Sentinel-compatible mode

After the Sentinel-compatible mode is enabled, you can use one of the following methods to connect to the ApsaraDB for Redis instance:

If password-free access is enabled for the ApsaraDB for Redis instance, you can connect to the instance in Sentinel-compatible mode. Otherwise, you must configure authentication information when you connect to the instance.

- **Connect to an ApsaraDB for Redis instance in Sentinel-compatible mode without using passwords**

  > **Note** For more information about how to enable password-free access, see Enable password-free access.

  The following sample code shows how to connect to an ApsaraDB for Redis instance by using Spring Data Redis:

```
    @Bean
    public JedisConnectionFactory connectionFactory() {
        RedisSentinelConfiguration sentinelConfig = new RedisSentinelConfiguration()
                .master("original-master-name")
                .sentinel(original-sentinel-1-host, original-sentinel-1-port)
                .sentinel(original-sentinel-2-host, original-sentinel-2-port);
        JedisPoolConfig poolConfig = new JedisPoolConfig();
        ...
        JedisConnectionFactory connectionFactory = new JedisConnectionFactory(sentinelCon
fig, poolConfig);
        return connectionFactory;
    }
```

  The following section describes the parameters:

  - master-name: the name of the master node. You can specify a custom name. Example: testmaster.
  - sentinel-host: the VPC endpoint that is used to connect to the ApsaraDB for Redis instance.
  - sentinel-port: the port number that is used to connect to the ApsaraDB for Redis instance. The default port number is 6379.

The following sample code shows how to connect to an ApsaraDB for Redis instance in Sentinel-compatible mode without using passwords:

```
    @Bean
    public JedisConnectionFactory connectionFactory() {
        RedisSentinelConfiguration sentinelConfig = new RedisSentinelConfiguration()
                .master("any-name")
                .sentinel("r-********.redis.rds.aliyuncs.com", 6379);
        JedisPoolConfig poolConfig = new JedisPoolConfig();
        ...
        JedisConnectionFactory connectionFactory = new JedisConnectionFactory(sentinelCon
fig, poolConfig);
        return connectionFactory;
    }
```

- **Connect to an ApsaraDB for Redis instance in Sentinel-compatible mode by using passwords**

  A Java client of the earliest version is used in this example. The client must meet the following requirements:

  - Jedis 3.6.0 or later is used.
  - Lettuce 5.3.0.RELEASE or later is used.
  - Spring Data Redis 2.5.1 or later is used. The spring.redis.sentinel.password parameter is specified for the Spring Data Redis client.

  > **Note** We recommend that you upgrade the clients to the latest stable versions. For more information about the latest versions, visit What's New in Maven.

  The following sample code shows how to connect to an ApsaraDB for Redis instance by using passwords:

```
        String masterName = "original-master-name";
        Set<String> sentinels = new HashSet<>();
        sentinels.add("original-sentinel-1-host:original-sentinel-1-port");
        sentinels.add("original-sentinel-2-host:original-sentinel-2-port");
        GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
        String dbPassword = "original-db-password";
        String sentinelPassword = "original-sentinel-password";
        JedisSentinelPool jedisSentinelPool =
                new JedisSentinelPool(masterName, sentinels, poolConfig,
                        2000, 2000, dbPassword,
                        0, null, 2000, 2000,
                        sentinelPassword, null);
```

  The following section describes the parameters:

  - masterName: the name of the master node. You can specify a custom name. Example: testmaster.
  - sentinels.add: the VPC endpoint and port number that are used to connect to the ApsaraDB for Redis instance. Format: `r-********.redis.rds.aliyuncs.com:6379` .

- dbPassword/sentinelPassword: the password of the account that is used to connect to the ApsaraDB for Redis instance. The password format varies based on the account that you select. If you forget your password, you can reset it. For more information about how to reset a password, see Change or reset the password.

  > ? Note
  >
  > - If you use the default account whose username is the same as the instance ID, you can enter only the password.
  > - If you use a custom account, the format of the password must be `<user>:<password>`. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa`, you must enter `testaccount:Rp829dlwa` as the database password.

The following sample code shows how to connect to an ApsaraDB for Redis instance in Sentinel-compatible mode by using passwords:

```
String masterName = "any-name";
Set<String> sentinels = new HashSet<>();
sentinels.add("r-********.redis.rds.aliyuncs.com:6379");
GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();
String dbPassword = "testaccount:Rp829dlwa";
String sentinelPassword = "testaccount:Rp829dlwa";
JedisSentinelPool jedisSentinelPool =
        new JedisSentinelPool(masterName, sentinels, poolConfig,
                2000, 2000, dbPassword,
                0, null, 2000, 2000,
                sentinelPassword, null);
```

FAQ

- Q: What do I do if the `NOAUTH Authentication required` error message appears when I switch from the native Redis Sentinel mode to the Sentinel-compatible mode provided by ApsaraDB for Redis?

  A: You can upgrade your client, modify some code to add a password for Sentinel authentication, and then try again. For more information, see the **Connect to an ApsaraDB for Redis instance in Sentinel-compatible mode by using passwords** section of this topic.

7.8. Use a public endpoint to connect to an ApsaraDB for Redis instance

If you want to test or manage an ApsaraDB for Redis instance that is deployed on an on-premises machine, you can apply for a public endpoint for the instance and connect to the instance over the Internet.

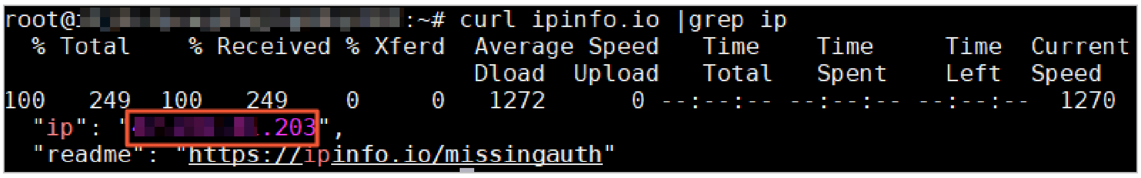
Precautions

No traffic fees are charged when you connect to an ApsaraDB for Redis instance over the Internet. However, this increases network latency, compromises the performance of the ApsaraDB for Redis service, and results in security risks. We recommend that you connect to ApsaraDB for Redis instances though VPC for lower network latency and higher security.

Procedure

1. Select the following methods to obtain the public IP address of the on-premises machine based on the operating system:
   - Linux operating system

     Run the **curl ipinfo.io |grep ip** command on the on-premises machine to obtain the public IP address. The returned result is shown in the following figure.

     ```
     root@[redacted]:~# curl ipinfo.io |grep ip
       % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                      Dload  Upload   Total   Spent    Left  Speed
     100   249  100   249    0     0   1272      0 --:--:-- --:--:-- --:--:--  1270
       "ip": "[redacted].203",
       "readme": "https://ipinfo.io/missingauth"
     ```

   - Windows operating system

     On the on-premises machine, visit ipinfo to obtain the public IP address of the on-premises machine.
2. Add the public IP address of the on-premises machine to the whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.
3. Apply for a public endpoint for the ApsaraDB for Redis instance. For more information, see .
4. Use the redis-cli tool or a Redis client program to connect to the ApsaraDB for Redis instance based on the obtained public endpoint.
   - Use redis-cli to connect to an ApsaraDB for Redis instance
   - Use a client to connect to an ApsaraDB for Redis instance

Related information

- Troubleshooting for connection issues in ApsaraDB for Redis

7.9. Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks

This topic describes methods to connect an Elastic Compute Service (ECS) instance to an ApsaraDB for Redis instance when they are deployed in different types of networks. For example, when an ECS instance is configured in the classic network and an ApsaraDB for Redis instance resides in a virtual private cloud (VPC), you can connect them by using the methods described in this topic.

Prerequisites

The ECS instance and the ApsaraDB for Redis instance are deployed in the same region. For more information about ECS instances, see What is ECS?

Network types

Network type	Description
VPC (recommended)	A VPC is a private network dedicated to your Alibaba Cloud account. VPCs are logically isolated from each other at Layer 2 to provide higher security and performance. If a Redis client is deployed on an ECS instance, you can connect the client to an ApsaraDB for Redis instance over a VPC for higher security and lower network latency.
Classic network	Cloud services on the classic network are not isolated. Unauthorized access can be blocked only by using security groups or whitelists.

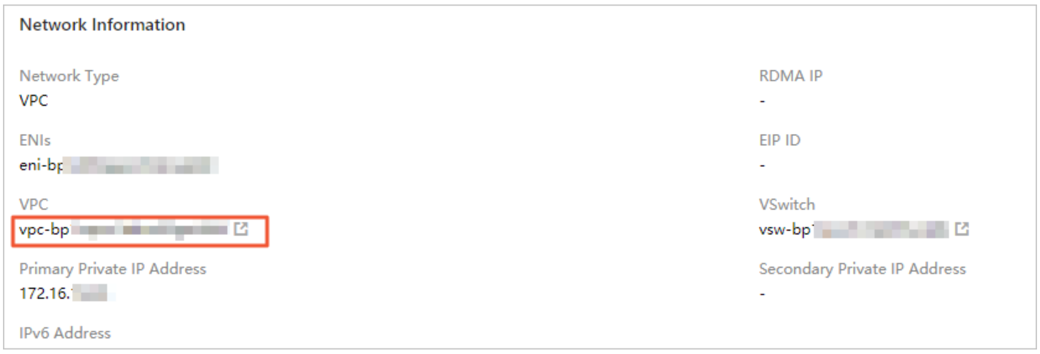
For more information about internal network types, see Network types.

> **Note** In addition to connections over an internal network, ApsaraDB for Redis instances also support connections over the Internet. For more information, see Use a public endpoint to connect to an ApsaraDB for Redis instance.

Connect an ECS instance located in a VPC to an ApsaraDB for Redis instance located in the classic network

1.
2.
3.
4. On the Instances page, click the ID of the ECS instance that you want to use.
5. In the **Network Information** section, obtain the ID of the VPC to which the ECS instance belongs.

Obtain the VPC ID of the ECS instance



Network Information

Network Type
VPC

RDMA IP
-

ENIs
eni-bp[redacted]

EIP ID
-

VPC
vpc-bp[redacted]

VSwitch
vsw-bp[redacted]

Primary Private IP Address
172.16.[redacted]

Secondary Private IP Address
-

IPv6 Address

6. Switch the ApsaraDB for Redis instance to the VPC to which the ECS instance belongs. For more information, see Change the network type from classic network to VPC.

> **Note** The network type can be switched from classic network to VPC but not from VPC to classic network.

Connect an ECS instance located in the classic network to an ApsaraDB for Redis instance located in a VPC

1.
2. In the **Basic Information** section, obtain the ID of the VPC to which the ApsaraDB for Redis instance belongs.

3. Select one of the following methods to achieve network interconnection based on your business requirements:
    - Migrate Switch the ECS instance to the VPC to which the ApsaraDB for Redis instance belongs. We recommend that you use this method. For more information, see Migrate an ECS instance from a classic network to a VPC.
    - Use ClassicLink to connect the ECS instance to the VPC to which the ApsaraDB for Redis instance belongs. For more information about ClassicLink, see Overview. For more information about the method procedure, see Connect a classic network to a VPC.

      > Note
      >
      > - When you configure security group rules of ClassicLink, you must set the protocol to *Custom TCP*. The port range is *6379/6379*.
      > - The ClassicLink-based interconnection is a temporary solution. To achieve high-speed connection in the production environment, we recommend that you migrate the ECS and ApsaraDB for Redis instances to the same VPC.

What's next

Select one of the following connection methods based on your business requirements:

- Use redis-cli to connect to an ApsaraDB for Redis instance
- Use a client to connect to an ApsaraDB for Redis instance

8.Data management

8.1. Manage ApsaraDB for Redis instances by using DMS

Data Management (DMS) provides multiple extended features and allows you to manage ApsaraDB for Redis instances by using commands or interfaces.

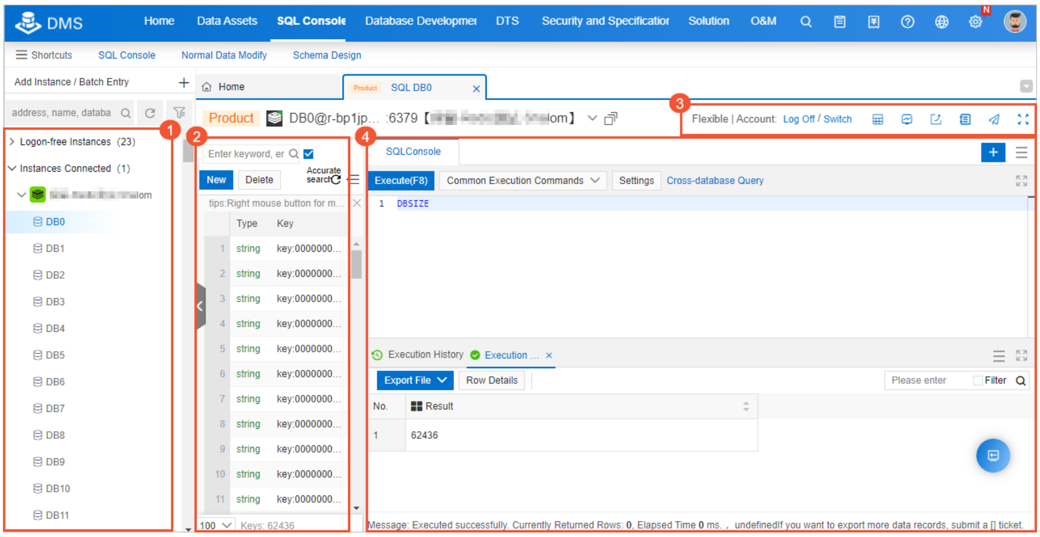
Context

DMS is an all-in-one data management service that supports multiple relational databases and NoSQL databases. The service offers features such as data management, schema management, user authorization, security audit, data trend analysis, and data tracking. For more information about DMS, see Overview. You can use DMS to manage databases with ease. This enhances data security, improves management efficiency, and maximizes data value.

Procedure

1. Log on to the ApsaraDB for Redis console by using DMS. For more information, see Log on to an ApsaraDB for Redis instance by using DMS.
2. Perform section-specific operations based on your business requirements.

   DMS interface



	Type	Key
1	string	key:0000000...
2	string	key:0000000...
3	string	key:0000000...
4	string	key:0000000...
5	string	key:0000000...
6	string	key:0000000...
7	string	key:0000000...
8	string	key:0000000...
9	string	key:0000000...
10	string	key:0000000...
11	string	key:0000000...

```
1  DBSIZE
```

No.	Result
1	62436

Message: Executed successfully. Currently Returned Rows: 0, Elapsed Time 0 ms. , undefinedIf you want to export more data records, submit a [] ticket.

No.	Section	Description
①	Database selection section	Double-click database names to switch ApsaraDB for Redis databases.

No.	Section	Description
②	Visual operation section	Manage keys in ApsaraDB for Redis databases: ○ Add keys: Click **New**. On the right-side tab, specify the key name, data type, time to live (TTL) or timeout period, and key value, and then click **Submit**. ○ Delete keys: In the key list, select the keys that you want to delete, and then click **Delete**. You can hold down the Shift key and click to select multiple keys at a time. ○ Search for keys: Enter the key name in the search box and click the 🔍 icon. ○ Modify keys: In the key list, double-click the key that you want to modify, enter a key value and TTL on the right-side tab, click **Submit**, and then click **Execute**. **Note** You cannot modify key names and data types.
③	Extended feature section	In this section, shortcuts to extended features are provided. You can use these features by clicking the following icons: ○ (DAS shortcut): provides real-time performance and instance sessions. The feature allows you to obtain performance information about ApsaraDB for Redis databases in real time. For more information, see View real-time performance metrics and View sessions of an ApsaraDB for Redis instance. ○ (operation audit): stores all data queries and change records. The feature allows you to query information about performed operations, such as the user that performed the operation, and the time when the operation was performed. ○ (sharing): allows you to share the console and database commands with relevant staff. For more information, see Share the SQLConsole tab or a ticket.
③	CLI section	Enter the command that you want to run such as **DBSIZE**, and then click **Execute** to view the output and history information about the execution. **Notice** DMS has limits on Redis commands. For more information, see SQL Console for Redis.

8.2. Delete data

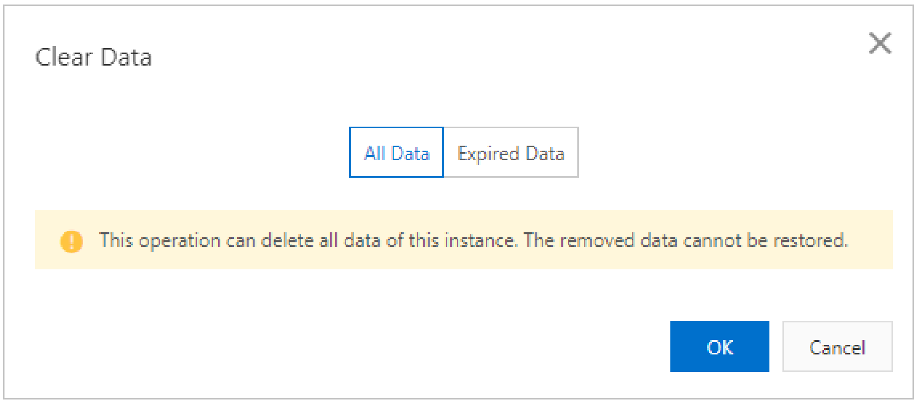
You can delete all data or expired data of an ApsaraDB for Redis instance in the ApsaraDB for Redis console.

Note Even if you disable the **FLUSHALL** command on the **Parameter Settings** page, the **Clear Data** feature can still be used.

Procedure

1.
2. On the **Instance Information** page, click **Clear Data** in the upper-right corner.
3. In the dialog box that appears, select the data that you want to delete.

Delete data



Clear Data

This operation can delete all data of this instance. The removed data cannot be restored.

- **All Data**: runs the **FLUSHALL** command to delete all data of the instance. Deleted data cannot be restored.
- **Expired Data**: runs the **SCAN** command to batch delete all expired data of the instance. Deleted data cannot be restored. You can select **Update Now** or Update During Maintenance.

Warning Data deletion immediately takes effect and deleted data cannot be restored. This may affect your business. Proceed with caution. We recommend that you back up the data of an ApsaraDB for Redis instance before you delete data. For more information, see Automatic or manual backup.

4. Click **OK**.

Note If you select **All Data**, you can specify whether to back up the data after you click OK.

Related API operations

Operation	Description
FlushInstance	Deletes all data of an ApsaraDB for Redis instance.
FlushExpireKeys	Deletes expired keys of an ApsaraDB for Redis instance.

9.Performance monitoring

9.1. Monitoring metrics

9.1.1. View monitoring data

ApsaraDB for Redis supports a variety of performance metrics. You can query the monitoring data of an ApsaraDB for Redis instance during a specified period of time in the previous month. This help you gain insights into the status of the ApsaraDB for Redis instance and troubleshoot issues.

Metric granularity

To avoid adversely affecting the operation of the instance and ensure the effect of trend charts, metrics are collected at different frequencies and within different time ranges.

Supported time range	Data collection interval
10 minutes or less	5 seconds
More than 10 minutes and less than or equal to one day	1 minute
More than one day and less than or equal to seven days	1 hour

> Note
>
> - ApsaraDB for Redis allows you to query the monitoring data of the previous month. The maximum time range that you can specify for a query is seven days. For example, if the current day is March 10, 2021, you can query the monitoring data from February 10 to March 10. If you set the start time to February 10, the latest end time is February 17.
> - If a cluster or read/write splitting instance has 32 or more data nodes, the time range that you can specify for all data nodes on the **All** tab cannot exceed 1 hour. The monitoring data is collected at intervals of 1 minute.

Procedure

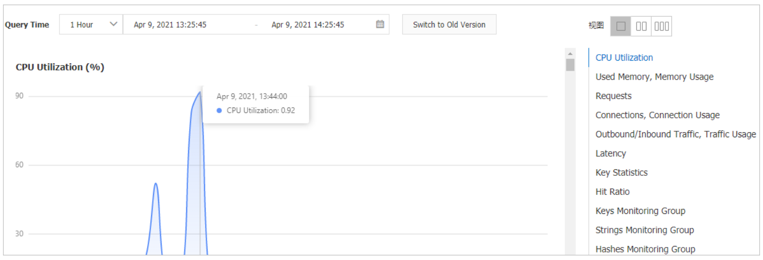
1.
2. In the left-side navigation pane, click **Performance Monitor**.
3. Perform the following operations based on the architecture of the ApsaraDB for Redis instance.

   > Note You can query the monitoring data of the previous month. The maximum time range that you can specify for a query is seven days. For example, if the current day is March 10, 2021, you can query the monitoring data from February 10 to March 10. If you set the start time to February 10, the latest end time is February 17.

<table>
<tr><th>Architecture</th><th>Procedure</th></tr>
<tr><td>Standard</td><td>

Specify a time range.

View the monitoring data of a standard instance

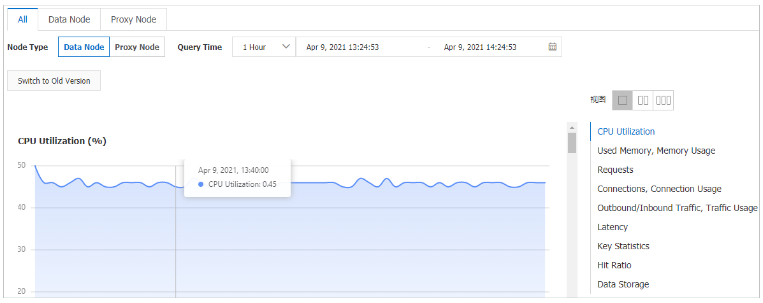


</td></tr>
<tr><td>Cluster
Read/write splitting</td><td>

Select a node type and specify a time range. Click one of the following tabs:

- **All**: displays the average values of metrics for all data nodes or proxy nodes. The following figure shows the average CPU utilization of all data nodes.
- **Data Node** or **Proxy Node**: displays the trend chart of a specified node.

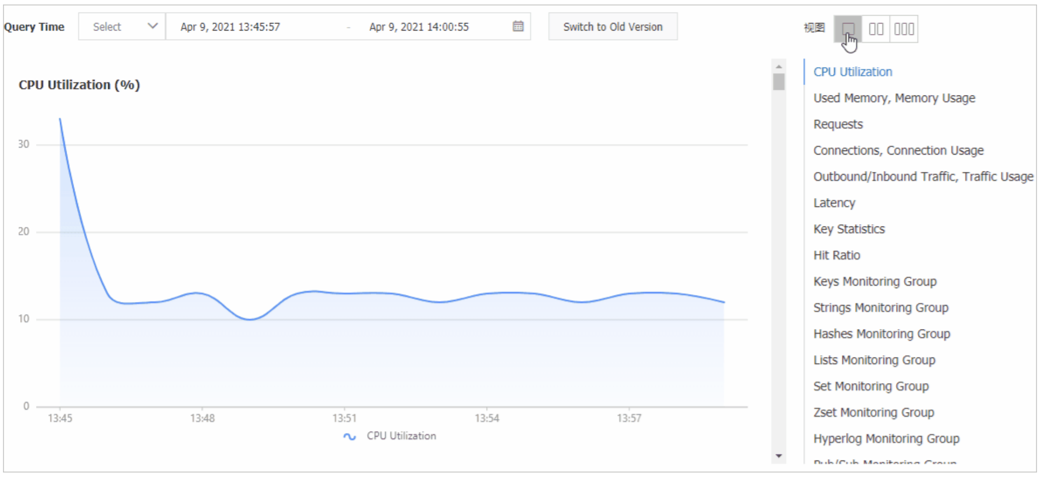
View the monitoring data of a cluster instance or read/write splitting instance



</td></tr>
</table>

4. To view the trend of a metric within a specified time period, you can perform the following operations on the current page.

Trend chart of a metric



Note For more information about metrics, see Metrics.

Related API operations

Operation	Description
DescribeHistoryMonitorValues	Queries the monitoring data of an ApsaraDB for Redis instance. **Note** When you call this operation, you must use the MonitorKeys parameter to specify the metrics that you want to monitor and then retrieve the corresponding monitoring data. For more information about the MonitorKeys parameter, see MonitorKeys.

9.1.2. Metrics

This topic describes the metrics that are collected in real time to monitor the status of ApsaraDB for Redis instances and the limits of these metrics in use.

Query monitoring data

For more information, see View monitoring data.

Metric groups for data nodes

Metrics in basic metric groups and commands-related metric groups are collected for data nodes of an ApsaraDB for Redis instance. For cluster instances and read/write splitting instances, some metrics are displayed in theFor example, if the CPU utilization data of all data nodes is aggregated, the average CPU utilization of data nodes in the current instance is displayed. aggregate views

> **Note** The metrics for read replicas of read/write splitting instances are not aggregated.

Basic metric groups

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
CPU Utilization	**CPU Utilization**	**CpuUsage**	✔☺	%	N/A
Used Memory, Memory	**Memory Usage**	**memoryUsage**	✔☺	%	The engine version of the ApsaraDB for Redis instance must be Redis 4.0 or later. For more information about how to upgrade the engine version, see Upgrade the major version.
	Used Memory	**UsedMemory**	✔☺	Bytes	The amount of used memory, which includes the memory consumed by data and cache.

Usage Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
	Memory Usage of Lua Scripts	**UsedMemoryLua**	▯	Bytes	N/A
Requests	**Total QPS**	**TotalQps**	✔☺	Counts/s	The total number of requests per second, which includes read and write requests.
	Read QPS	**GetQps**	✔☺	Counts/s	Indicates the number of reads per second and the number of writes per second. Note The engine version of the ApsaraDB for Redis instance must be Redis 4.0 or later and the latest minor version must be used. For more information about how to upgrade the minor version, see Upgrade the major version and Update the minor version.
	Write QPS	**PutQps**	✔☺	Counts/s	

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
	Connection Usage	**Connection Usage**	✔☺	%	Connection usage = Number of used connections/Number of supported connections. For more information about connections that are supported by an ApsaraDB for Redis instance, see Overview.

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
Connections, Connection Usage	**Used Connections**	**ConnCount (UsedConnection)**	✔	Counts	**Note** • If clients connect to a cluster instance in direct connect mode, you must pay attention to this metric. • If clients connect to a cluster instance or read/write splitting instance through a proxy node, you can ignore this metric. In this case, pay attention to metrics for the proxy node. For more information, see Metric groups for proxy nodes.

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
Outbound /Inbound Traffic Rate	**Inbound Traffic Rate**	**IntranetIn**	✔☺	KBps	N/A
	Outbound Traffic Rate	**IntranetOut**	✔☺	KBps	
Outbound /Inbound Traffic Usage	**Inbound Traffic Usage**	**IntranetInRatio**	✔☺	%	N/A
	Outbound Traffic Usage	**IntranetOut Ratio**	✔☺	%	
Network latency	**Average Latency**	**AvgRt**	✔☺	us	The average response time of all commands. This is the average time that is measured from the time when a data node receives a command request to the time when the data node returns the response.
Key Statistics	**Total Keys**	**Keys**	✔☺	Counts	The total number of primary keys that are stored in the instance.
	Total Keys with TTLs	**Expires**	✔☺	Counts	This metric indicates the instantaneous value of the total number of keys when the data is collected.
	Total Purged Keys	**ExpiredKeys**	✔☺	Counts	N/A
	Total Evicted Keys	**EvictedKeys**	✔☺	Counts	
	Purged Keys Per Second	**ExpiredKeys PerSecond**	✔☺	Counts/s	
	Evicted Keys Per Second	**EvictedKeys PerSecond**	✔☺	Counts/s	

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
Hit Ratio	Hit Ratio	hit_rate	✔☺	%	The hit ratio of keys. Hit ratio = Total key hits/(Total key hits + Total key misses). Note If you use ApsaraDB for Redis 2.8, the metric is not collected. You must upgrade the minor version of the instance. For more information, see Update the minor version.
Hits and Misses	Hits Per Second	hit	✔☺	Counts	N/A
	Misses Per Second	miss (Miss Count)	✔☺	Counts	
Disk Information	Total Disk Usage	ins_size	✔☺	MB	Only ApsaraDB for Redis instances of Enterprise Edition (Hybrid-storage instances (phased out)) support this metric.
	Disk Usage of Data Files	DataSize	✔☺	MB	The size of data files on the persistent disk, which includes append-only files (AOF) and Redis Database (RDB) files. Note Only ApsaraDB for Redis instances of Enterprise Edition (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
	Disk Usage of Log Files	**LogSize**	✔☺	MB	The size of log files on the persistent disk, which includes the operational logs of ApsaraDB for Redis instances. **Note** Only ApsaraDB for Redis instances of Enterprise Edition (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.
Statistics of Keys on Disk and in Memory	**Keys on Disk**	**swapped_key**	✔☺	Counts	Only ApsaraDB for Redis instances of Enterprise Edition (Hybrid-storage instances (phased out)) support this metric.
	Keys in Memory	**inmem_keys**	✔☺	Counts	

Commands-related metric groups

Metric group	Support foraggregate views	Unit	Description and limit
Key Monitoring Group	□	Counts/s	Information about key-value commands, such as the number of times DEL and EXITS are executed.
String Monitoring Group	□	Counts/s	Information about string commands, such as the number of times APPEND and MGET are executed.
Hash Monitoring Group	□	Counts/s	Information about hash-related commands, such as the number of times HGET and HDEL are executed.
List Monitoring Group	□	Counts/s	Information about list commands, such as the number of times BLPOP and BRPOP are executed.
Set Monitoring Group	□	Counts/s	Information about set commands, such as the number of times SADD and SCARD are executed.
Zset Monitoring Group	□	Counts/s	Information about zset commands, such as the number of times ZADD and ZCARD are executed.

Metric group	Support foraggregate views	Unit	Description and limit
HyperLog Monitoring Group	▯	Counts/s	Information about HyperLogLog commands, such as the number of times PFADD and PFCOUNT are executed.
Pub/Sub Monitoring Group	▯	Counts/s	Information about publication and subscription commands, such as the number of times PUBLISH and SUBSCRIBE are executed.
Transaction Monitoring Group	▯	Counts/s	Information about transaction commands, such as the number of times WATCH, MULTI, and EXEC are executed.
Lua Script Monitoring Group	▯	Counts/s	Information about commands for Lua scripts, such as the number of times EVAL and SCRIPT are executed.
Tairdoc Monitoring Group	▯	Counts/s	Information about TairDoc commands, such as the number of times JSON.SET and JSON.GET are executed. **Note** Only ApsaraDB for Redis instances of Enterprise Edition (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.
TairHash Monitoring Group	▯	Counts/s	Information about TairHash commands, such as the number of times EXHSET and EXHMSET are executed. **Note** Only ApsaraDB for Redis instances of Enterprise Edition (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.
TairString Monitoring Group	▯	Counts/s	Information about TairString commands, such as the number of times EXSET and EXGET are executed. **Note** Only ApsaraDB for Redis instances of Enterprise Edition (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.

Metric group	Support foraggregate views	Unit	Description and limit
TairGis Monitoring Group	▯	Counts/s	Information about TairGis commands, such as the number of times GIS.ADD, GIS.GET, and GIS.DEL are executed. **Note** Only ApsaraDB for Redis instances of Enterprise Edition (Tair) (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.
TairBloom Monitoring Group	▯	Counts/s	Information about TairBloom commands, such as the number of times BF.RESERVE and BF.ADD are executed. **Note** Only ApsaraDB for Redis instances of Enterprise Edition (Performance-enhanced instances) and (Hybrid-storage instances (phased out)) support this metric.

Metric groups for proxy nodes

Metrics for proxy nodes of cluster and read/write splitting instances are collected. Some metrics are displayed in theFor example, if the CPU utilization data of all proxy nodes is aggregated, the average CPU utilization of proxy nodes in the current instance is displayed. aggregate views

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
CPU	**CPU Utilization**	**CpuUsage**	✔☺	%	N/A
	Total QPS	**TotalQps**	✔☺	Counts/s	The total number of requests per second, which includes read and write requests.

Requests Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
	Read QPS	**GetQps**	✔☺	Counts/s	Indicates the number of reads per second and the number of writes per second. **Note** To collect this metric, you must upgrade the minor version of the ApsaraDB for Redis instance to the latest version. For more information, see Update the minor version.
	Write QPS	**PutQps**	✔☺	Counts/s	
Connections	**Connections Usage**	**Connection Usage**	✔☺	%	Connection usage = Number of used connections/Number of supported connections. For more information about connections that are supported by an ApsaraDB for Redis instance, see Overview.
	Used Connections	**UsedConnection**	✔☺	Counts	

Metric group	New metric name	Original metric name	Support foraggregate views	Unit	Description and limit
Outbound /Inbound Traffic	**Inbound Traffic Rate**	**IntranetIn**	✔☺	KBps	N/A
	Outbound Traffic Rate	**IntranetOut**	✔☺	KBps	
Size of Requests and Responses	**Average Bytes Per Request**	**AvgRequestSize**	▯	Byte	N/A Note • If clients connect to a cluster instance or read/write splitting instance through a proxy node, you must pay attention to this metric. • If clients connect to a cluster instance of ApsaraDB for Redis in direct connect mode, you can ignore this metric. In this case, pay attention to the metrics for data nodes. For more information, see Metric groups for data nodes.
	Average Bytes Per Response	**AvgResponseSize**	▯	Byte	
	Maximum Bytes Per Request	**MaxRequestSize**	▯	Byte	
	Maximum Bytes Per Response	**MaxResponseSize**	▯	Byte	
Latency	**Average Latency**	**AvgRt**	✔☺	us	The average response time of all commands. This is the average time that is measured from the time when the proxy node receives a command to the time when the proxy node returns the response.

9.1.3. Customize metrics (previous version)

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

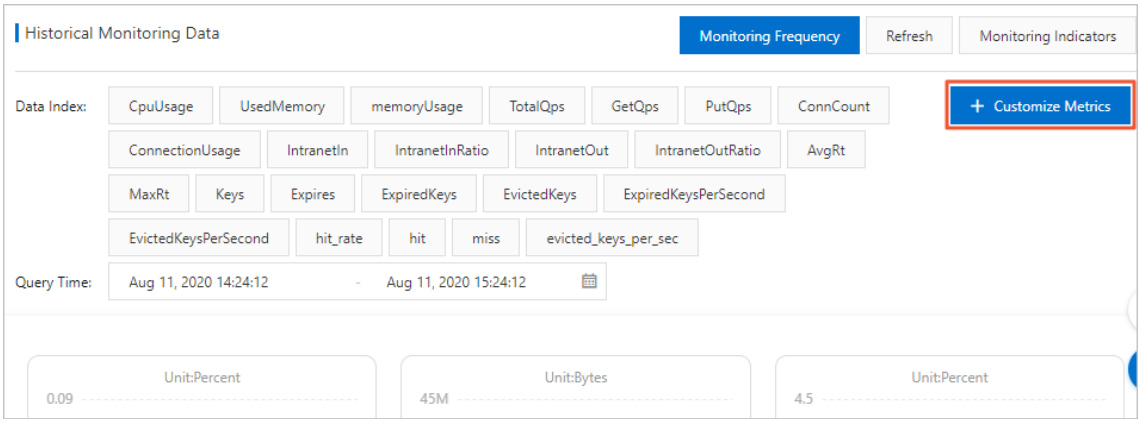
Context

Notice The Performance Monitor page has been upgraded to improve user experience. On the new Performance Monitor page, you are not allowed to select the metrics to be displayed and all metrics are by default displayed on the new page. For more information, see Announcement: performance monitoring upgraded.

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

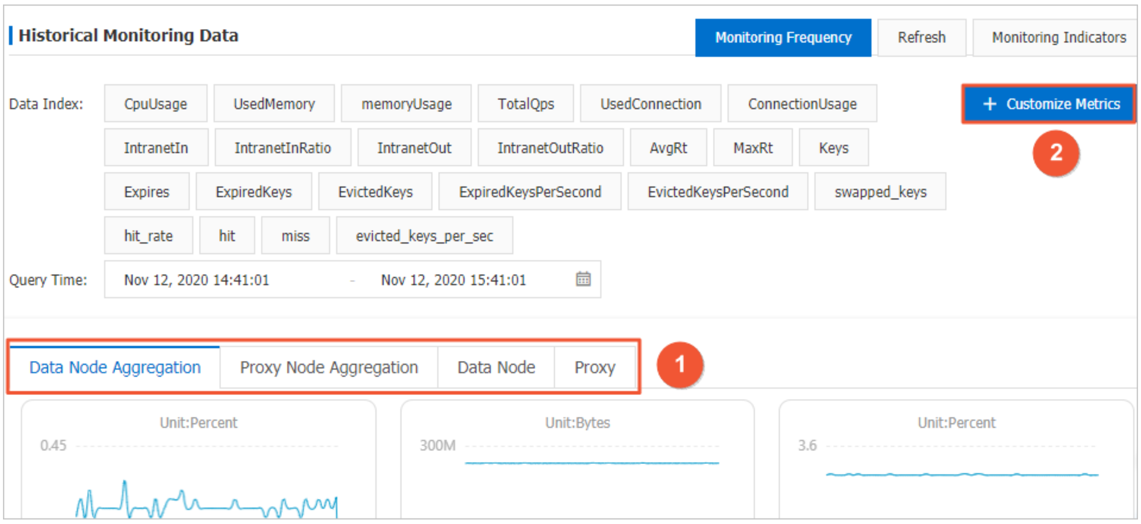
Procedure

1.
2. In the left-side navigation pane, click **Performance Monitor**.
3. Perform the following operations based on the architecture of the instance:
   - Standard master-replica instances: In the upper-right corner of the page, click **Customize Metrics**.



Query Time: Aug 11, 2020 14:24:12 - Aug 11, 2020 15:24:12

   - Cluster instances or read/write splitting instances: Click the tab of the node type to be displayed and click **Customize Metrics** in the upper-right corner of the page.



Query Time: Nov 12, 2020 14:41:01 - Nov 12, 2020 15:41:01

4. In the dialog box that appears, select a monitoring group.
5. Click **OK**.

> Note
>
> - For more information about monitoring groups, see Metrics.
> - If the node type for a cluster instance or read/write splitting instance is **Data Node** or **Proxy**, you can select a node from the **Screening Node** drop-down list.

9.1.4. Modify the data collection interval (previous version)

You can specify the interval at which monitoring data is collected in the ApsaraDB for Redis console.

Context

Notice The Performance Monitor page has been upgraded to improve user experience. On the new Performance Monitor page, you cannot specify the interval at which monitoring data is collected and all monitoring data is by default collected at intervals of seconds. For more information, see Announcement: performance monitoring upgraded.

You can set the data collection interval to **5 Seconds** or **60 Seconds**. The default data collection interval is **60 Seconds**. In most cases, the default settings are sufficient. If you want to collect some metrics at a higher frequency with low latency, you can perform the following operations to change the data collection interval into **5 Seconds**. Monitoring data does not occupy storage of an instance and the instance is not affected when monitoring data is collected.

For more information about the relationship between the data collection interval and monitoring data, see the methods that are used to collect metrics in Metrics.

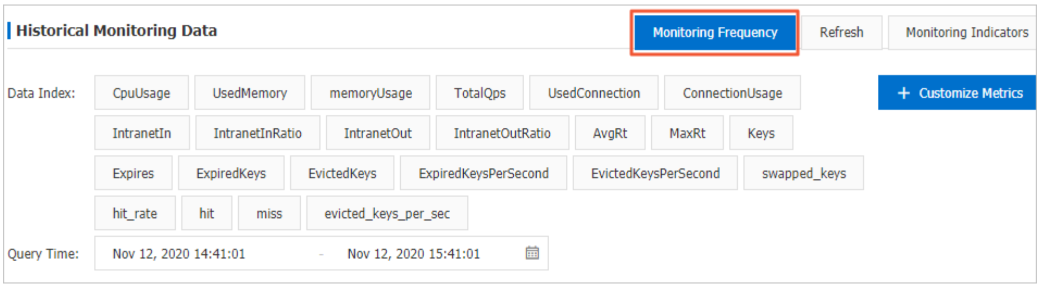
Precautions

If the data collection interval is set to **5 Seconds**, the time range of a query cannot exceed 10 minutes.

Procedure

1.
2. In the left-side navigation pane, click **Performance Monitor**.
3. In the upper-right corner of the page, click **Monitoring Interval**.

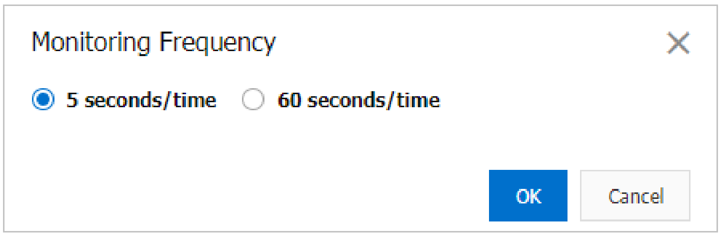
Performance Monitor page



Query Time: Nov 12, 2020 14:41:01 - Nov 12, 2020 15:41:01

4. In the dialog box that appears, select an interval and click **OK**.

Specify the data collection interval



● 5 seconds/time ○ 60 seconds/time

9.2. Performance of queries

9.2.1. Latency insight

CloudDBA provides the latency insight feature to collect millisecond-level latency statistics of all commands that are run and custom events that are executed on ApsaraDB for Redis databases. Latency insight enables you to troubleshoot anomalies and performance issues of ApsaraDB for Redis databases.

Features

Redis 2.8.13 introduced a new feature called latency monitoring to help users identify and troubleshoot possible latency issues. The latency monitoring feature allows you to collect data generated only within the last 160 seconds and access only events that have the highest latency within each second.

As such, ApsaraDB for Redis provides the advanced latency insight feature. With latency insight, up to 27 events and execution durations of all commands can be recorded, and all latency statistics within the last three days can be saved. For more information about the events, see the "Common events" section of this topic. Latency insight provides the following benefits:

- Persistent: supports data persistence and latency spike tracing.
- High-precision: allows full events to be monitored within milliseconds.
- High-performance: supports asynchronous implementations with minimal impact on performance.
- Real-time: supports real-time data queries and aggregation operations.
- Multidimensional: provides comprehensive latency data that allows you to analyze an instance based on events, time, and latency.

Prerequisites

The ApsaraDB for Redis instance uses one of the following minor versions. For information about how to update a minor version, see Update the minor version.

- Minor version 1.6.9 or later if the instance is a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances. If you want to collect statistics about Tair module commands, update the minor version to 1.7.28 or later.
- Minor version 5.1.4 or later if the instance is a Community Edition instance that uses the 5.0 major version.
- Minor version 0.1.15 or later if the instance is a Community Edition instance that uses the 6.0 major version.

Procedure

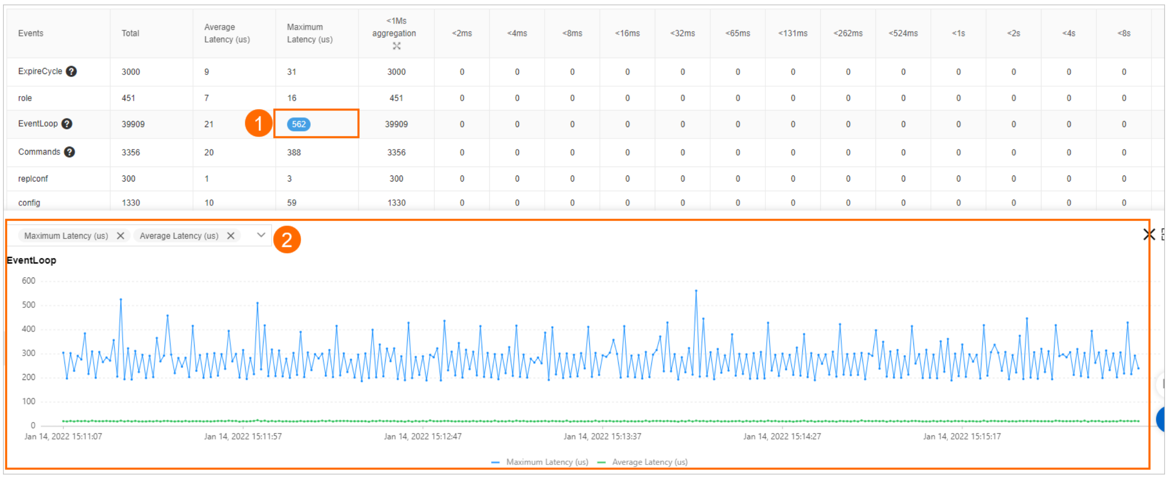
1.
2. In the left-side navigation pane, choose **CloudDBA > Latency Insight**.
3. On the page that appears, specify the time range to query and then click **Search**. The default time range is the last 5 minutes.

   > **Note** Only data of the last three days can be queried, and the time range to query must span within one hour.

4. Click the name of an event or a number corresponding to an event in the table. Then, a chart appears and shows the trend of the event-matched metric.

   You can also specify the metrics that you want to view on the chart by selecting the metric names from the drop-down list above the chart.

   > **Note** Only commands or events that take longer than the specified amount of time to run or execute are recorded and displayed.



Events	Total	Average Latency (us)	Maximum Latency (us)	<1Ms aggregation ⤢	<2ms	<4ms	<8ms	<16ms	<32ms	<65ms	<131ms	<262ms	<524ms	<1s	<2s	<4s	<8s
ExpireCycle ?	3000	9	31	3000	0	0	0	0	0	0	0	0	0	0	0	0	0
role	451	7	16	451	0	0	0	0	0	0	0	0	0	0	0	0	0
EventLoop ?	39909	21	562	39909	0	0	0	0	0	0	0	0	0	0	0	0	0
Commands ?	3356	20	388	3356	0	0	0	0	0	0	0	0	0	0	0	0	0
replconf	300	1	3	300	0	0	0	0	0	0	0	0	0	0	0	0	0
config	1330	10	59	1330	0	0	0	0	0	0	0	0	0	0	0	0	0

Metric	Description
Event	The name of the event. Example values: ExpireCycle, EventLoop, Ping, Scan, Commands, and Info. For more information, see the "Common events" section of this topic.
Total	The total number of occurrences of the event.
Average Latency (μs)	The average latency of the event. Unit: μs.
Maximum Latency (μs)	The maximum latency of the event. Unit: μs.
Aggregation of Instances (Latency < 1ms)	The number of occurrences of the event whose latency is lower than 1 ms. You can click the ⤢ icon to view finer-grained statistics, including the number of occurrences of the event whose latency is lower than 1 μs, 2 μs, 4 μs, 8 μs, 16 μs, 32 μs, 64 μs, 128 μs, 256 μs, and 512 μs. **Note** Counting method: The number of occurrences of the event whose latency is from 0 μs to 1 μs is counted and presented under the <1μs category, and the number of occurrences of the event whose latency is from 1 μs to 2 μs is counted and presented under the <2μs category. Other categories follow the same pattern.

Metric	Description
<2ms <4ms ... >33s	The number of occurrences of the event whose latency is greater than or equal to 1 ms. **Note** Counting method: The number of occurrences of the event whose latency is from 1 ms to 2 ms is counted and presented under the <2ms category, and the number of occurrences of the event whose latency is higher than 33s is counted and presented under the >33s category. Other categories follow the same pattern.

Common events

Category	Name	Threshold	Description
Memory eviction	EvictionDel	30ms	The amount of time it takes to evict a key.
	EvictionLazyFree	30ms	The amount of time it takes to evict a key by using the Lazyfree feature.
	EvictionCycle	30ms	The amount of time it takes to perform an eviction.
Memory defragmentation	ActiveDefragCycle	100ms	The amount of time it takes to defragment memory.
Rehash	Rehash	100ms	The amount of time it takes to perform a rehash.
Data structure upgrade	ZipListConvertHash	30ms	The amount of time it takes to convert a ziplist to a dictionary by means of hash encoding.
	IntsetConvertSet	30ms	The amount of time it takes to convert an intset to a set by means of set encoding.
	ZipListConvertZset	30ms	The amount of time it takes to convert a ziplist to a skiplist by means of ziplist encoding.
Append-only file (AOF)	AofWriteAlone	30ms	The uptime during an AOF write
	AofWrite	30ms	The amount of time it takes to perform each AOF write. An AOF write can be of the AofWriteAlone, AofWriteActiveChild, or AofWritePendingFsync type.
	AofFsyncAlways	30ms	The amount of time it takes to perform a fsync operation on an AOF when the appendfsync option is set to 1.
	AofFstat	30ms	The amount of time it takes to obtain status information about an AOF.
	AofRename	30ms	The amount of time it takes to rename an AOF.

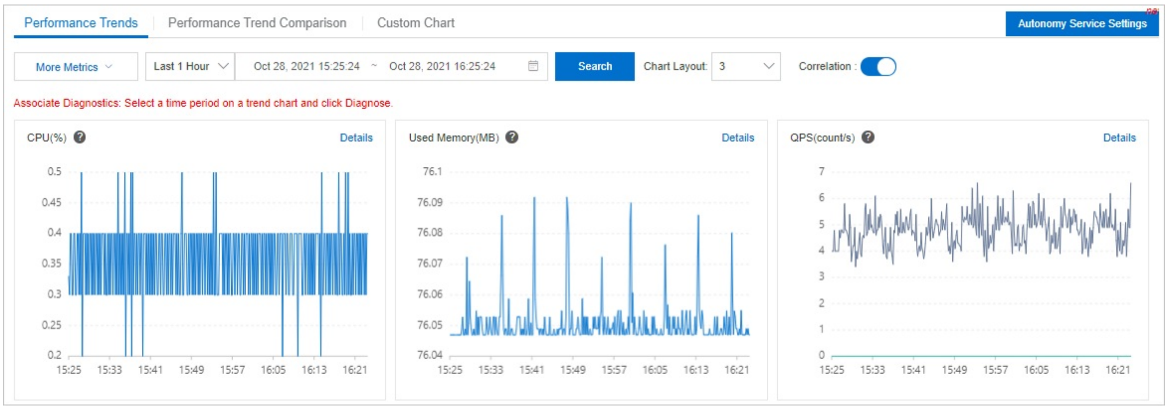
Category	Name	Threshold	Description
	AofReWriteDiffWrite	30ms	The amount of time consumed by an incremental AOF write performed by a parent process after its child process rewrites an AOF.
	AofWriteActiveChild	30ms	The amount of time it takes to perform an AOF write when other child processes are in progress.
	AofWritePendingFsync	30ms	The amount of time it takes to perform an AOF write when a fsync operation is in progress.
Redis database (RDB) file	RdbUnlinkTempFile	50ms	The amount of time it takes to delete a temporary RDB file after a bgsave child process is terminated.
Others	Commands	30ms	The amount of time it takes to run a command that is not tagged with fast.
	FastCommand	30ms	The amount of time it takes to run a command that is tagged with fast, such as GET or EXISTS.
	EventLoop	50ms	The amount of time it takes to have a main event loop running.
	Fork	100ms	The amount of time recorded in a parent process after the parent process is forked.
	Transaction	50ms	The actual amount of time consumed by a transaction.
	PipeLine	50ms	The amount of time consumed by a multi-threaded pipeline.
	ExpireCycle	30ms	The amount of time consumed by a regular deletion of an expired key.
	SlotRdbsUnlinkTempFile	30ms	The amount of time it takes to delete a temporary RDB file from a slot after a bgsave child process is terminated.
	LoadSlotRdb	100ms	The amount of time it takes to load an RDB file from a slot.
	SlotreplTargetcron	50ms	The amount of time it takes to load an RDB file from a slot to a temporary database and then migrate the file to a destination database by using a child process.

9.2.2. Performance trends

CloudDBA provides the performance trends feature that allows you to monitor the basic performance metrics and corresponding performance trends of an ApsaraDB for Redis instance within a specified period of time. The performance trends include the CPU utilization, memory usage, queries per second (QPS), total number of connections, response time, network traffic, and key hit ratio.

Procedure

1.
2. In the left-side navigation pane, choose **CloudDBA > Performance Trends**.
3. Use the following methods to view performance trends:



Last 1 Hour Oct 28, 2021 15:25:24 ~ Oct 28, 2021 16:25:24

Chart Layout: 3

Correlation: on

Associate Diagnostics: Select a time period on a trend chart and click Diagnose.

- **Performance Trends**

  On the **Performance Trends** tab, specify a time range, select more metrics, and then click **Search**.

  > Note
  >
  > - You can select a specific node for a read/write splitting or cluster instance.
  > - By default, **Correlation** is enabled. If you move the pointer over the CPU chart to view the CPU metric of the instance at 09:00, other charts also display other metrics of the instance at 09:00.
  > - Click ? in the upper-left corner of each metric chart to view the definition of the metric and click **Details** in the upper-right corner of each metric chart to obtain a larger view of the metric chart.

- **Performance Trend Comparison**

  To compare the performance trends within two periods of time, click the **Performance Trend Comparison** tab, specify two periods of time, select more metrics, and then click **Search**.

- **Custom Chart**

  The preceding two methods display the basic performance metrics of an ApsaraDB for Redis instance. If you want to display only some basic metrics, you can configure custom performance trend charts. For more information, see Add a custom performance trend chart.

9.2.3. Add a custom performance trend chart

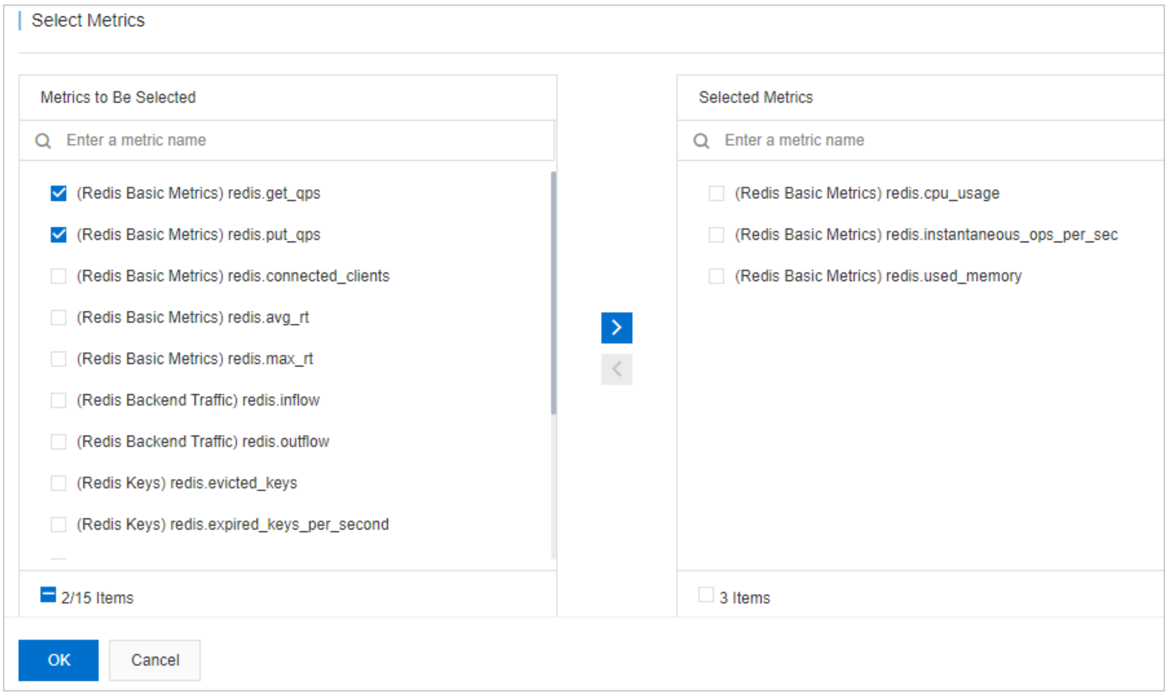
The default performance trends tab displays the basic performance metrics of an ApsaraDB for Redis instance. You can add a custom chart that contains only specific basic performance metrics to analyze the performance trends of the instance. This topic describes how to add a custom performance trend chart to a dashboard for an ApsaraDB for Redis instance.

Procedure

1.
2. In the left-side navigation pane, choose **CloudDBA > Performance Trends**.
3. Click the **Custom Chart** tab.
4. Choose **Operate Dashboard > Create Monitoring Dashboard**. Enter a name for the dashboard and click **OK**.

   > **Note** If no dashboards have been created for the instance, click **Add Monitoring Dashboard**.

5. Click **Add Node and Metrics** or **Add Monitoring Chart**.
6. Select the nodes that you want to view and move them to the **Selected Nodes** section, select the metrics that you want to view and move them to the **Selected Metrics** section, and then click **OK**.



Select Metrics

Metrics to Be Selected

Enter a metric name

☑ (Redis Basic Metrics) redis.get_qps
☑ (Redis Basic Metrics) redis.put_qps
□ (Redis Basic Metrics) redis.connected_clients
□ (Redis Basic Metrics) redis.avg_rt
□ (Redis Basic Metrics) redis.max_rt
□ (Redis Backend Traffic) redis.inflow
□ (Redis Backend Traffic) redis.outflow
□ (Redis Keys) redis.evicted_keys
□ (Redis Keys) redis.expired_keys_per_second

2/15 Items

Selected Metrics

Enter a metric name

□ (Redis Basic Metrics) redis.cpu_usage
□ (Redis Basic Metrics) redis.instantaneous_ops_per_sec
□ (Redis Basic Metrics) redis.used_memory

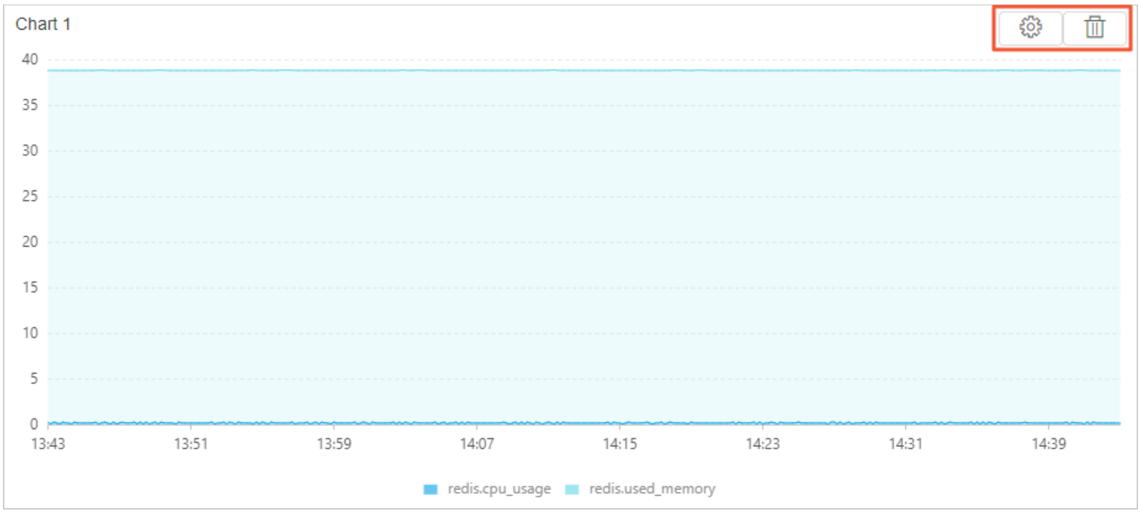
□ 3 Items

7. (Optional) Manage the dashboard.
   - View the dashboard

     Select the monitoring dashboard that you want to view, specify a time range, and then click **Search**.

   - Modify the dashboard

     Click one of the icons as shown in the following figure to modify or delete the dashboard.



- Delete the dashboard

  Choose **Operate Dashboard > Delete Monitoring Dashboard**.

9.2.4. View performance metrics in real time

CloudDBA allows you to view the performance metrics of an ApsaraDB for Redis instance in real time. The performance metrics include CPU utilization, memory usage, queries per second (QPS), network traffic, servers, keys, clients, and connections and more.

Procedure

1.
2. In the left-side navigation pane, choose **CloudDBA > Real-time Performance**.
3. You can view **Global Real-time Node Performance** and **Real-time Performance**.
   - **Global Real-time Node Performance**

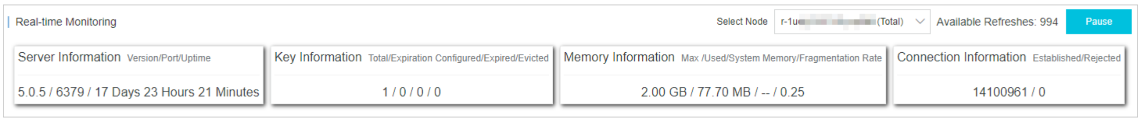
     The Global Real-time Node Performance tab displays metrics such as CPU utilization, memory usage, QPS, network traffic, and keys of each node in the instance in real time.

     > **Note** Only read/write splitting instances and cluster instances support this feature.

   - **Real-time Performance**

     > **Note**
     >
     > - For a read/write splitting instance or cluster instance, you can select a specific node that you want to view.
     > - To allow you to view performance changes in real time, the metrics are automatically refreshed every 5 seconds. The available refreshes are displayed in the upper-right corner. To stop updating the performance metrics, you can click **Pause**.

     In the upper part of the page, performance metrics are displayed in real time. The metrics include servers, keys, memory usage, clients, and connections.



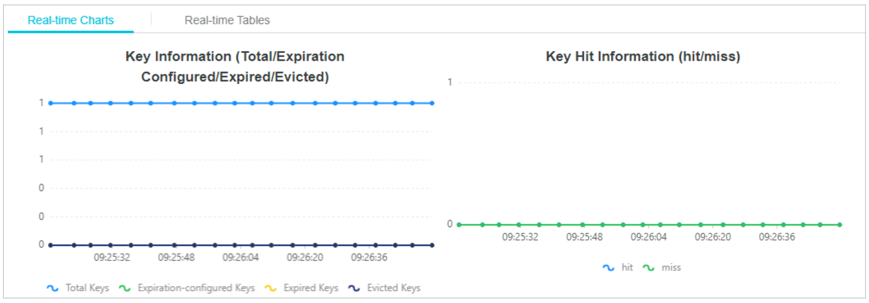
Real-time Monitoring

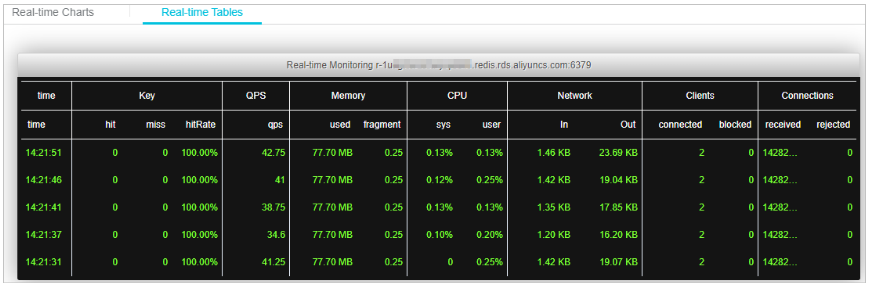
Select Node r-1ue[redacted](Total) Available Refreshes: 994 Pause

Server Information Version/Port/Uptime	Key Information Total/Expiration Configured/Expired/Evicted	Memory Information Max /Used/System Memory/Fragmentation Rate	Connection Information Established/Rejected
5.0.5 / 6379 / 17 Days 23 Hours 21 Minutes	1 / 0 / 0 / 0	2.00 GB / 77.70 MB / -- / 0.25	14100961 / 0

Detailed performance metrics are displayed in **Real-time Charts** and **Real-time Tables**.

<table>
<tr><th>Pattern</th><th>Description</th></tr>
<tr><td><b>Real-time Charts</b></td><td>Displays the real-time performance metrics of the instance in curve charts, such as keys, key hit information, key hit ratio, CPU utilization, memory usage, QPS, and network traffic.


</td></tr>
<tr><td><b>Real-time Tables</b></td><td>Displays the instance performance metrics in tables, including key hit information, key hit ratio, QPS, memory usage, CPU utilization, network traffic, clients, and connections. The table can display up to 999 records. A new record is added to the table every 5 seconds.



Real-time Charts Real-time Tables

Real-time Monitoring r-1u[redacted].redis.rds.aliyuncs.com:6379

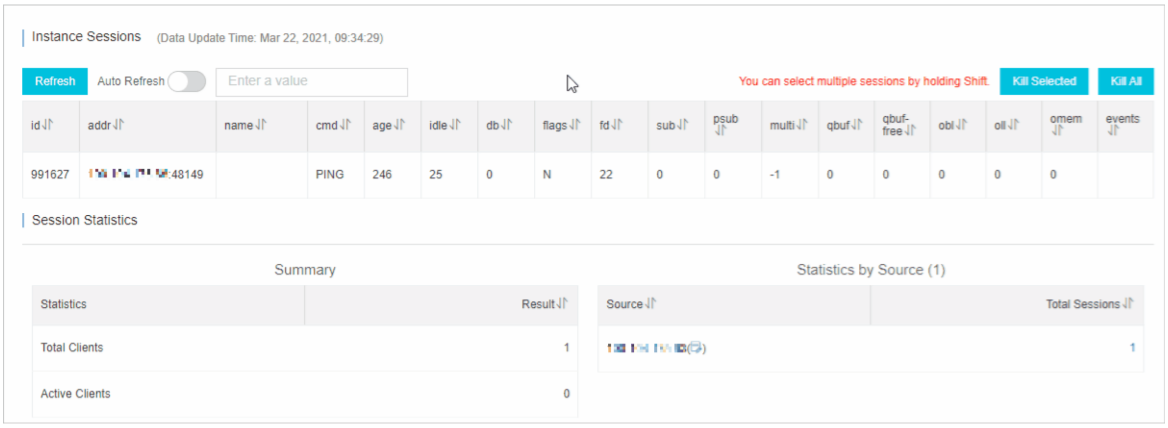
<table>
<tr><th>time</th><th colspan="3">Key</th><th>QPS</th><th colspan="2">Memory</th><th colspan="2">CPU</th><th colspan="2">Network</th><th colspan="2">Clients</th><th colspan="2">Connections</th></tr>
<tr><th>time</th><th>hit</th><th>miss</th><th>hitRate</th><th>qps</th><th>used</th><th>fragment</th><th>sys</th><th>user</th><th>In</th><th>Out</th><th>connected</th><th>blocked</th><th>received</th><th>rejected</th></tr>
<tr><td>14:21:51</td><td>0</td><td>0</td><td>100.00%</td><td>42.75</td><td>77.70 MB</td><td>0.25</td><td>0.13%</td><td>0.13%</td><td>1.46 KB</td><td>23.69 KB</td><td>2</td><td>0</td><td>14282...</td><td>0</td></tr>
<tr><td>14:21:46</td><td>0</td><td>0</td><td>100.00%</td><td>41</td><td>77.70 MB</td><td>0.25</td><td>0.12%</td><td>0.25%</td><td>1.42 KB</td><td>19.04 KB</td><td>2</td><td>0</td><td>14282...</td><td>0</td></tr>
<tr><td>14:21:41</td><td>0</td><td>0</td><td>100.00%</td><td>38.75</td><td>77.70 MB</td><td>0.25</td><td>0.13%</td><td>0.13%</td><td>1.35 KB</td><td>17.85 KB</td><td>2</td><td>0</td><td>14282...</td><td>0</td></tr>
<tr><td>14:21:37</td><td>0</td><td>0</td><td>100.00%</td><td>34.6</td><td>77.70 MB</td><td>0.25</td><td>0.10%</td><td>0.20%</td><td>1.20 KB</td><td>16.20 KB</td><td>2</td><td>0</td><td>14282...</td><td>0</td></tr>
<tr><td>14:21:31</td><td>0</td><td>0</td><td>100.00%</td><td>41.25</td><td>77.70 MB</td><td>0.25</td><td>0</td><td>0.25%</td><td>1.42 KB</td><td>19.07 KB</td><td>2</td><td>0</td><td>14282...</td><td>0</td></tr>
</table></td></tr>
</table>

9.2.5. Instance sessions

Instance sessions allow you to view the statistics about sessions between an ApsaraDB for Redis instance and clients in real time. The statistics include clients, commands that are run, and connection durations. You can also close abnormal sessions based on your business requirements.

Procedure

1.
2. In the left-side navigation pane, choose **CloudDBA** > **Instance Sessions**.
3. You can view or close the **Instance Sessions** for the instance and view the **Session Statistics** of the instance.



Instance Sessions (Data Update Time: Mar 22, 2021, 09:34:29)

Refresh | Auto Refresh | Enter a value | You can select multiple sessions by holding Shift. | Kill Selected | Kill All

id	addr	name	cmd	age	idle	db	flags	fd	sub	psub	multi	qbuf	qbuf-free	obl	oll	omem	events
991627	[redacted]:48149		PING	246	25	0	N	22	0	0	-1	0	0	0	0	0	

Session Statistics

Summary

Statistics	Result
Total Clients	1
Active Clients	0

Statistics by Source (1)

Source	Total Sessions
[redacted]	1

- View sessions

  By default, the details of all sessions are displayed. You can move the pointer over a specific parameter name to view its description.

  > Note
  >
  > - You can enter keywords in the search box to filter sessions.
  > - To refresh instance sessions, click **Refresh** in the upper-left corner or enable **Auto Refresh** to automatically refresh the page every 30 seconds.

- Close sessions

  To close a specific session, press the Shift key, select the session, and then click **Kill Selected** in the upper-right corner. To close all sessions, click **Kill All** in the upper-right corner.

  > Warning To prevent unexpected consequences, we recommend that you do not close system-level sessions.

- View session statistics

  Session statistics record the total number of clients, the number of active clients, and source IP addresses (also called IP addresses of clients) of instance sessions.

  > Note In the Statistics by Source table, click the icon on the right of a source IP address to modify its alias. In the Total Sessions column, click a value to view the details about a source IP address.

9.2.6. Slow queries

Slow queries decrease the stability of ApsaraDB for Redis instances. To monitor and analyze slow queries, you can view the details about slow query logs in CloudDBA.

Procedure

1.
2. In the left-side navigation pane, choose **CloudDBA > Slow Queries**.
3. On the slow queries page, view **Slow Log Trend** and **Slow Log Details**.

- Slow Log Trend

  In the **Slow Log Trend** section, you can view the **number of slow query logs** and CPU utilization for the instance in a specified period. You can click a specific point in time to view the slog query log details.

  > **Note** For master-replica cluster and read/write splitting instances, the following information is displayed: the slow query log details for **data nodes** and **proxy nodes** and the **number of slow queries** of each node.

- Slow Log Details

  By default, the details of all slow queries are displayed in the following columns: **Query Started At**, **Database Name**, **Slow Query Statement**, **Elapsed**, and **Host Address**. You can click **Set** to filter the columns.

  > **Note**
  >
  > - You can click **Export Slow Log** in the upper-right corner to export slow query logs to a local storage for analysis.
  > - By default, the **Host Address** parameter for master-replica cluster instances and read/write splitting instances displays the IP address of proxy nodes. To obtain the IP address of a specific client, set ptod_enabled to `1` in **System Parameters**. For more information, see Modify parameters of an instance.

9.3. Diagnose instances

9.3.1. Create a diagnostic report

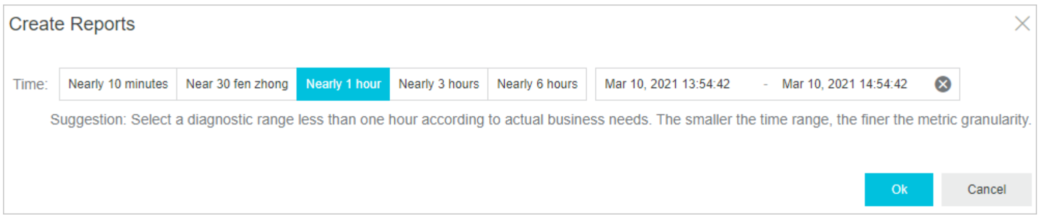
ApsaraDB for Redis integrates the diagnostics feature of Database Autonomy Service (DAS). ApsaraDB for Redis allows you to create a diagnostic report to evaluate the status of an ApsaraDB for Redis instance within a specified period of time. Diagnostic reports collect metrics such as performance level, skewed request distribution, and slow logs to help you identify exceptions in the instance.

Procedure

1.
2. In the left-side navigation pane, choose **CloudDBA > Diagnostic reports**.
3. Click **Create Reports**.
4. In the dialog box that appears, specify a time range for which you want to create a diagnostic report and click **OK**.

   > **Note** The time range cannot exceed one day. If you specify a shorter time range for diagnostics, more detailed statistics are collected in the diagnostic report.

   Create a diagnostic report



Create Reports

Time: Nearly 10 minutes | Near 30 fen zhong | **Nearly 1 hour** | Nearly 3 hours | Nearly 6 hours | Mar 10, 2021 13:54:42 - Mar 10, 2021 14:54:42

Suggestion: Select a diagnostic range less than one hour according to actual business needs. The smaller the time range, the finer the metric granularity.

5. You can view the progress of the diagnostics task by refreshing the Diagnostic Report page. After the diagnostic report is created, click **View Report** in the **Actions** column.

> **Note** For more information about how to interpret a diagnostic report, see Analyze a diagnostic report.

9.3.2. Analyze a diagnostic report

Diagnostic reports help you evaluate the operational conditions of an ApsaraDB for Redis instance and identify anomalies on the instance based on statistics such as performance level, skewed request distribution, and slow logs.

Prerequisites

Create a diagnostic report

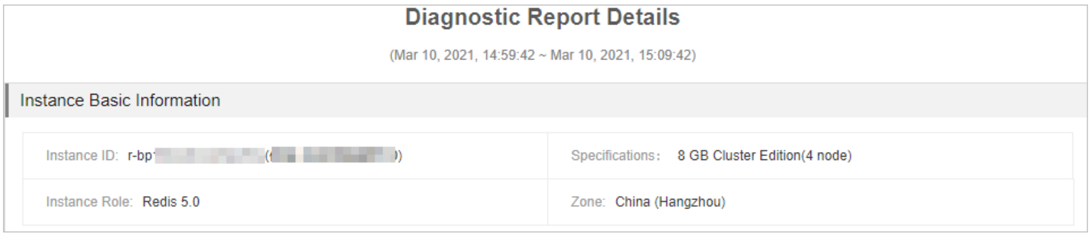
Components of a diagnostic report

- Basic instance information: displays basic information of an instance such as the instance ID, instance type, engine version, and the zone in which the instance is deployed.
- Summary: displays the score of the instance health status and describes the reasons why points are deducted.
- Performance level: displays the statistics and states of key performance metrics related to the instance.
- TOP 10 nodes that receive the greatest number of slow queries: displays the top 10 data nodes that receive the greatest number of slow queries and provides information about the slow queries.

Basic instance information

This section displays the instance ID, instance type, engine version, and the region in which the instance is deployed.

Basic instance information



Diagnostic Report Details

(Mar 10, 2021, 14:59:42 ~ Mar 10, 2021, 15:09:42)

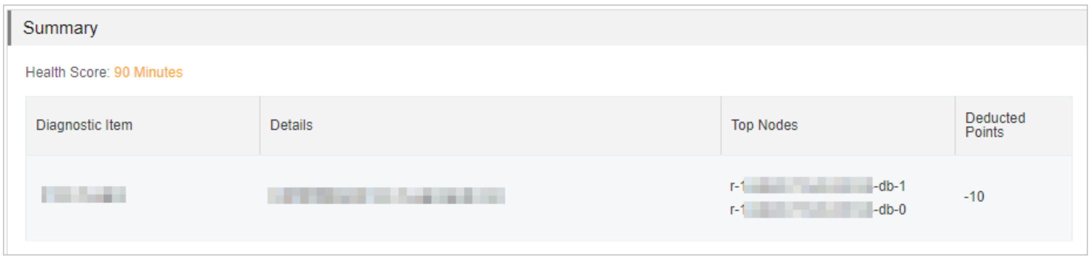
Instance Basic Information

Instance ID: r-bp[redacted]([redacted])	Specifications: 8 GB Cluster Edition(4 node)
Instance Role: Redis 5.0	Zone: China (Hangzhou)

Summary

This section displays the diagnostic results and the score of the instance health status. The highest score is 100. If your instance achieves a score lower than 100, you can check the diagnostic items and details.

Summary



Summary

Health Score: 90 Minutes

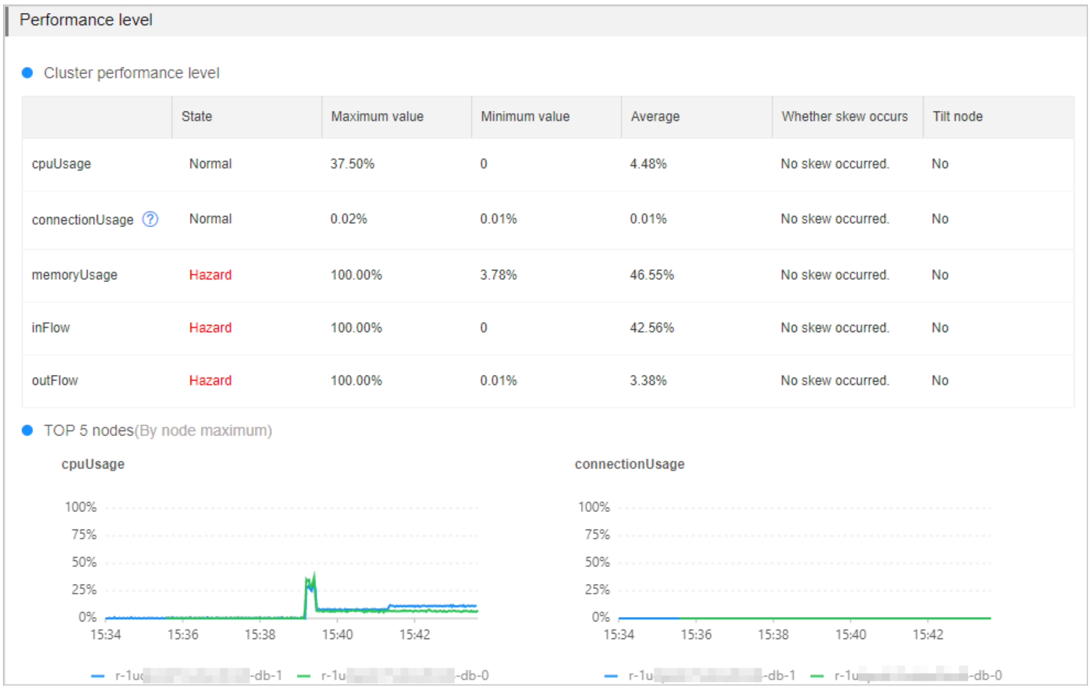
Diagnostic Item	Details	Top Nodes	Deducted Points
[redacted]	[redacted]	r-1[redacted]-db-1 r-1[redacted]-db-0	-10

Performance level

This section displays the statistics and states of key performance metrics related to the instance. You must pay attention to performance metrics that are in the **Hazard** state.

> **Note** If your instance runs in a cluster architecture or a read/write splitting architecture, you must check whether the performance metrics are skewed and check for skewed data nodes. For more information about the cluster and read/write splitting architecture, see Cluster master-replica instances and Read/write splitting instances. In addition, we recommend that you focus on the data nodes with higher loads based on the curve charts of each performance metric in the **Top 5 Nodes** section.

Performance level



Performance level

● Cluster performance level

	State	Maximum value	Minimum value	Average	Whether skew occurs	Tilt node
cpuUsage	Normal	37.50%	0	4.48%	No skew occurred.	No
connectionUsage	Normal	0.02%	0.01%	0.01%	No skew occurred.	No
memoryUsage	Hazard	100.00%	3.78%	46.55%	No skew occurred.	No
inFlow	Hazard	100.00%	0	42.56%	No skew occurred.	No
outFlow	Hazard	100.00%	0.01%	3.38%	No skew occurred.	No

● TOP 5 nodes(By node maximum)

Performance metric	Threshold	Impact	Possible cause and troubleshooting method

Performance metric	Threshold	Impact	Possible cause and troubleshooting method
CPU Utilization	60%	When an ApsaraDB for Redis instance has high CPU utilization, the throughput of the instance and the response time to clients are affected. In some cases, the clients may be unable to respond.	Possible causes: - The instance runs commands that require high time complexity. - Hotkeys exist. - Connections are frequently established. For more information about how to troubleshoot these issues, see Troubleshoot high CPU utilization on an ApsaraDB for Redis instance.
Memory Usage	80%	When the memory usage of an ApsaraDB for Redis instance continuously increases, response time increases, queries per second (QPS) becomes unstable, and keys may be frequently evicted. This affects your business.	Possible causes: - The memory is exhausted. - A great number of large keys exist. For more information about how to troubleshoot these issues, see Troubleshoot the high memory usage of an ApsaraDB for Redis instance.
Connections Usage of data nodes	80%	When the number of connections to a data node reaches the upper limit, connection requests may time out or fail. **Note** - This metric is collected when clients connect to an ApsaraDB for Redis cluster instance in direct connection mode. For more information about the direct connection mode, see Enable the direct connection mode. - This metric is not collected when clients connect to an ApsaraDB for Redis cluster instance or read/write splitting instance by using proxy nodes. In this case, you must monitor the number of connections on the proxy nodes. For more information, see View monitoring data.	Possible causes: - User traffic spikes. - Idle connections are not released for an extended period of time. For more information about how to troubleshoot these issues, see Instance sessions.

Performance metric	Threshold	Impact	Possible cause and troubleshooting method
Inbound Traffic	80%	When the inbound or outbound traffic exceeds the maximum bandwidth provided by the instance type, the performance of clients is affected.	Possible causes: ● Workloads spike. ● Large keys are frequently read or written. For more information about troubleshoot these issues, see Troubleshoot high traffic usage on an ApsaraDB for Redis instance.
Outbound Traffic	80%		

If your instance runs in the cluster architecture or read/write splitting architecture, the system measures the overall access performance of the instance based on the preceding performance metrics and displays the result in the diagnostic report. The following table describes the criteria used to determine skewed requests, possible causes of skewed requests, and troubleshooting methods.

Note If the diagnostic report indicates that the instance has skewed requests for a specific performance metric, you must check the nodes to which the skewed requests are directed.

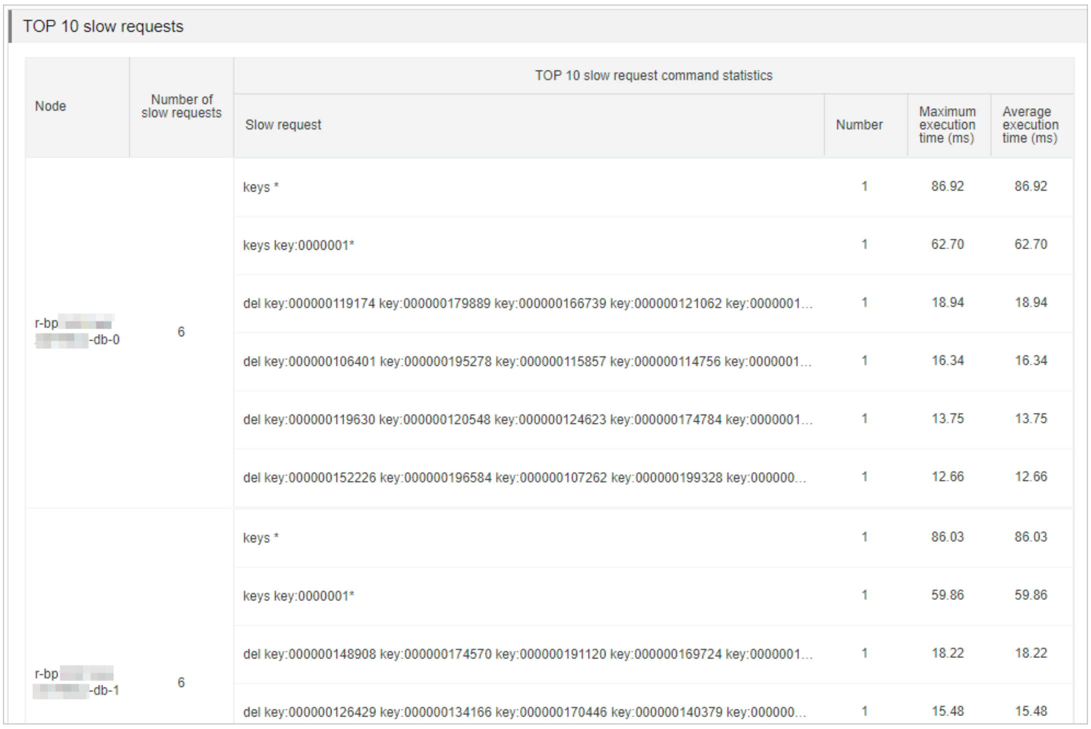
Criteria	Possible cause	Troubleshooting method
The following conditions are met: • Peak values of performance metrics for all data nodes of an ApsaraDB for Redis instance are greater than the following thresholds: ○ CPU utilization: 10%. ○ Memory usage: 20%. ○ Inbound and outbound traffic: 5 Mbit/s. ○ Connection usage: 5%. • The balance score is greater than 1.3, which is calculated by using the following formula: max{average performance values of all data nodes}/median performance value of all data nodes. For example, an ApsaraDB for Redis instance contains four data nodes and the average CPU utilization of the four nodes is 10%, 30%, 50%, and 60%. Then, the median value is 40% and the result is 1.5 from 60%/40%. The calculated value 1.5 is greater than 1.3. Therefore, the system considers the CPU utilization of the instance skewed.	• A data node has excessive large keys. • A data node has hotkeys. • The hash tags are improperly configured. **Note** If keys are configured with the same hash tag, the keys are stored on the same data node. If a large number of keys are configured with the same hash tag, the node is overwhelmed by these keys.	• Offline key analysis • Use the real-time key statistics feature

TOP 10 nodes that receive the greatest number of slow queries

This section displays the top 10 data nodes that receive the greatest number of slow queries and statistics about the slow queries. The statistics include the following slow logs:

- The slow logs of data nodes that are stored in the system audit logs. These slow logs are retained only for four days.
- The slow logs that are stored on the data node. Only the most recent 1,024 log entries are retained. You can use redis-cli to connect to the instance and run the **SLOWLOG GET** command to view these slow logs.

Slow query analysis



TOP 10 slow requests

Node	Number of slow requests	TOP 10 slow request command statistics			
		Slow request	Number	Maximum execution time (ms)	Average execution time (ms)
r-bp[redacted]-db-0	6	keys *	1	86.92	86.92
		keys key:0000001*	1	62.70	62.70
		del key:000000119174 key:000000179889 key:000000166739 key:000000121062 key:0000001...	1	18.94	18.94
		del key:000000106401 key:000000195278 key:000000115857 key:000000114756 key:0000001...	1	16.34	16.34
		del key:000000119630 key:000000120548 key:000000124623 key:000000174784 key:0000001...	1	13.75	13.75
		del key:000000152226 key:000000196584 key:000000107262 key:000000199328 key:000000...	1	12.66	12.66
r-bp[redacted]-db-1	6	keys *	1	86.03	86.03
		keys key:0000001*	1	59.86	59.86
		del key:000000148908 key:000000174570 key:000000191120 key:000000169724 key:0000001...	1	18.22	18.22
		del key:000000126429 key:000000134166 key:000000170446 key:000000140379 key:000000...	1	15.48	15.48

You can analyze the slow queries and determine whether improper commands exist. This way, you can find the solutions to different issues.

Cause	Solution
Commands that have a time complexity of O(N) or consume a large amount of CPU resources, such as **keys ***.	Evaluate and disable commands that cause a high risk and consume a large amount of CPU resources, such as **FLUSHALL**, **KEYS**, and **HGETALL**. For more information, see Disable high-risk commands.
Large keys that are frequently read from and written to the data nodes.	Analyze and evaluate the large keys. For more information, see Offline key analysis. Then, split these large keys based on your business requirements.

9.4. Alert settings

ApsaraDB for Redis has been integrated with CloudMonitor. This topic describes how to configure alert rules for important metrics of an ApsaraDB for Redis instance in the CloudMonitor console. This way, you can handle exceptions of metrics or instances at the earliest opportunity.

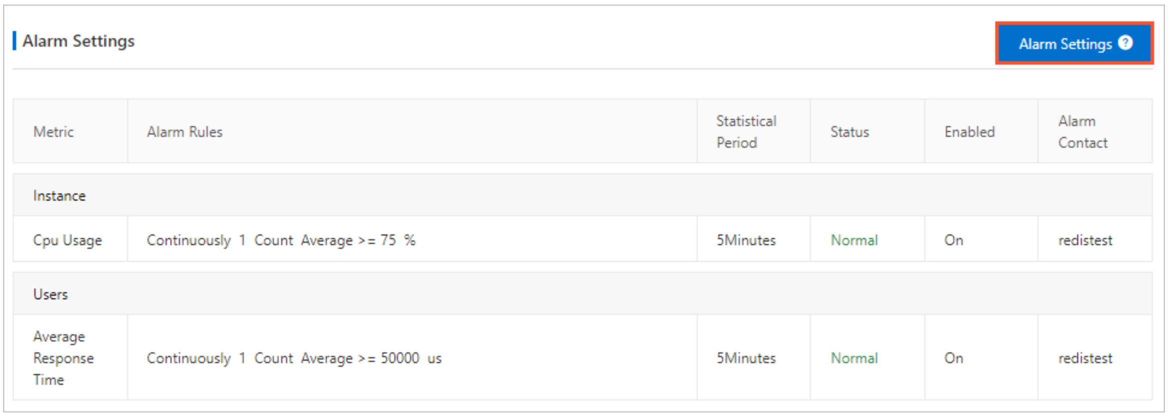
Background information

CloudMonitor is a service that can be used to monitor Alibaba Cloud resources and Internet applications. The service is an all-in-one enterprise-grade monitoring service that works out-of-the-box. For more information, see What is CloudMonitor? You can create alert rules and specify the metrics based on which alerts are set. When the alert rules of a specified metric are triggered, alerts are generated and sent to alert contacts in an alert contact group.

Note CloudMonitor sends alerts to alert contacts in alert contact groups. Before you add an alert contact to an alert contact group, you must create the alert contact or alert contact group. For more information, see Create an alert contact or alert contact group.

Procedure

1. Log on to the ApsaraDB for Redis console and go to the Instances page. In the top navigation bar, select the region in which the instance is deployed. Then, find the instance and click the instance ID.
2. In the left-side navigation pane, click **Alarm Settings**.
3. On the **Alarm Settings** page, view monitoring metrics of the current instance. You can also click **Alert Settings** in the upper-right corner to go to the CloudMonitor console to add or manage alert rules.

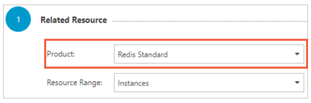


Alarm Settings

Metric	Alarm Rules	Statistical Period	Status	Enabled	Alarm Contact
Instance					
Cpu Usage	Continuously 1 Count Average >= 75 %	5Minutes	Normal	On	redistest
Users					
Average Response Time	Continuously 1 Count Average >= 50000 us	5Minutes	Normal	On	redistest

The following table describes the alert types that are supported by CloudMonitor.

Alert type	Instance type	Metric	Description	References

Alert type	Instance type	Metric	Description	References
Threshold-triggered alerts	When you create an threshold-triggered alert, you must specify the instance type to be associated with the alert. The following instance types are supported:  1 Related Resource Product: Redis Standard Resource Range: Instances ○ ApsaraDB for Redis (standard architecture) ○ ApsaraDB for Redis (cluster architecture) ○ ApsaraDB for Redis (read/write splitting architecture) ○ ApsaraDB for Redis Enhanced Edition (Tair) storage-optimized master-replica instances ○ ApsaraDB for Redis Enhanced Edition (Tair) persistent memory-optimized standard instances ○ ApsaraDB for Redis Enhanced Edition (Tair) persistent memory-optimized cluster instances	Metrics vary based on instance types. Cluster instances and read/write splitting instances separately support metrics for data nodes and proxy nodes. For more information, see ApsaraDB for Redis (standard architecture), ApsaraDB for Redis (cluster architecture), and ApsaraDB for Redis (read/write splitting architecture).	When the value of a metric exceeds the specified threshold, the system sends an alert. For example, if the CPU utilization of an ApsaraDB for Redis instance exceeds the 90% threshold, the system sends an alert. These alerts enable you to understand and respond to exceptions in a timely manner. In most cases, workloads are sensitive to fluctuations in the CPU utilization, memory usage, and network traffic of an ApsaraDB for Redis instance. We recommend that you specify alert thresholds for the main metrics. The following metrics and thresholds are provided for your reference: ○ CPU utilization: greater than 60%. ○ Memory usage: greater than 80%. ○ Inbound bandwidth usage and outbound bandwidth usage: greater than 80%. ○ Disk usage: greater than 80%. This threshold applies only to ApsaraDB for Redis Enhanced Edition (Tair) storage-optimized master-replica instances.	Create a threshold-triggered alert rule

Alert type	Instance type	Metric	Description	References
Event-triggered alerts	N/A	○ InstanceMaintenance (proactive O&M events) ○ Exceptions ○ Maintenance	If an ApsaraDB for Redis instance fails, performs a master-replica failover, or runs a proactive O&M task such as an instance migration, the system sends an alert. This allows you to resolve issues at the earliest opportunity.	Subscribe to event notifications

Related information

- Query and manage pending events
- Causes and impacts of master-replica switchovers

10.Log management

10.1. Audit logs (new version)

10.1.1. Enable the new audit log feature

ApsaraDB for Redis provides the new version of the audit log feature. The new version is integrated with Log Service and allows you to query, analyze online, and export log data. This helps you gain insights into the security and performance of ApsaraDB for Redis instances.

Prerequisites

- The instance is an ApsaraDB for Redis Community Edition instance or a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances.
- The engine version of the instance is Redis 4.0 or later, and the latest minor version is used. For more information about how to update the minor version and upgrade the engine version of an instance, see Upgrade the major version and Update the minor version.
- The **AliyunLogFullAccess** permission is granted to the Resource Access Management (RAM) user that is used to enable the new audit log feature. This requirement must be met if you want to enable the feature by using the credentials of a RAM user. For more information, see Grant permissions to a RAM user.
- The **AliyunLogFullAccess** or **AliyunLogReadOnlyAccess** permission is granted to a RAM user. This requirement must be met if you want to use the RAM user to access audit logs. For more information, see Grant permissions to a RAM user.

  All permissions or read-only permissions on Log Service are granted to the RAM user. This requirement must be met if you want to create Logstore-level custom policies. For more information, see Create a custom policy. The following code displays policy content.

  - All permissions:

    ```
    {
      "Version": "1",
      "Statement": [
        {
          "Action": "log:*",
          "Resource": "acs:log:*:*:project/nosql-*",
          "Effect": "Allow"
        }
      ]
    }
    ```

- Read-only permissions:

```
{
  "Version": "1",
  "Statement": [
    {
        "Action": [
          "log:Get*",
          "log:List*"
        ],
      "Resource": "acs:log:*:*:project/nosql-*",
      "Effect": "Allow"
    }
  ]
}
```

Scenarios

ApsaraDB for Redis integrates the features of Log Service to provide an audit log feature that is stable, flexible, simple, and efficient. This feature can be used in the scenarios described in the table. For more information about Log Service, see What is Log Service?

Scenario	Description
Operation audit	Helps security auditors check information such as operator identity or data modification time to identify internal risks such as permission abuse and execution of invalid commands.
Security compliance	Assists business systems in meeting the audit requirements in security compliance.

Precautions

- After you enable the audit log feature for an instance, ApsaraDB for Redis audits the write operations that were performed on the instance and logs the audit information. During the process, the instance may encounter a performance degrade of 5% to 15% and a specific amount of latency jitter. The performance decrease and the latency jitter vary based on the amount of data that is written or audited.

  Notice

  - Your application may write a large amount of data to an instance. For example, your application frequently runs the **INCR** command to count. To prevent a performance decrease in such a scenario, we recommend that you enable the audit log feature only for troubleshooting issues or auditing instance security.
  - The number of read operations is often large. If the audit information of a large number of read operations is recorded, the instance performance may deteriorate. To prevent this issue, ApsaraDB for Redis records audit information only for write operations.

- The specified log retention period for an instance is applicable to the instance and all the other instances that reside in the same region as the instance. Other settings of the instance are applied only to the instance. For example, if you enable the audit log feature for an instance, the audit log feature takes effect only on the instance.

Billing

You are charged for the audit log feature based on storage usage and log retention period. The price varies based on regions that you select. For more information, see Billable items and prices.

> **Note** The free trial version of the audit log feature has been phased out on June 11, 2021. For more information, see [Notice] Official version of the audit log feature for ApsaraDB for Redis released.

Procedure

1.
2. In the left-side navigation pane, choose **Logs > Audit Log**.
3. Specify a log retention period.

   > **Note** This configuration is applicable to the instance and all instances that reside in the same region as the instance. Audit logs are billed based on storage usage and log retention period. Valid values for the log retention period are 1 to 365. Unit: days.

4. Click **Estimate Fees and Enable Audit Logs**.
5. In the dialog box that appears, estimate log fees, read the prompt, and then click **Enable**.

   > **Note** The audit log feature depends on Log Service. If Log Service is not activated for your Alibaba Cloud account, you are prompted for activating Log Service. For more information, see What is Log Service?

Related API operations

Operation	Description
ModifyAuditLogConfig	Enables or disables the audit log feature and specifies a retention period for audit logs.
DescribeAuditLogConfig	Queries the audit log configurations of an ApsaraDB for Redis instance. The configurations include whether the audit log feature is enabled and the retention period of audit logs.
DescribeAuditRecords	Queries the audit logs of an ApsaraDB for Redis instance.

FAQ

- How do I disable the audit log feature for an instance?

  Log on to the ApsaraDB for Redis console and go to the **Audit Log** page of the instance. In the upper-right corner of the page, click **Service Settings**. Then, you can disable the audit log feature.

- How do I download all audit logs?

  For more information, see 下载日志. To download all audit logs, take note of the following items:

    - To download all audit logs, you must specify the **redis_audit_log_standard** Logstore and specify the project name in the following format: `nosql-{`*ID of your Alibaba Cloud account*`}-{`*Region*`}`. Example: **nosql-17649847257****-cn-hangzhou**.
    - To download all audit logs, you must select **Download All Logs with Cloud Shell** or **Download All Logs Using Command Line Tool**. If you select **Download Log in Current Page**, you can download only the audit logs that are displayed on the current page.

- Why does the audit log feature support write operations but not read operations?

  In most scenarios, the number of read operations is larger than the number of write operations. The auditing for read operations can cause a serious performance degrade. In addition, a large number of audit logs need to be generated and stored for read operations. As such, ApsaraDB for Redis may discard specific audit logs to ensure service stability. Due to these issues, the audit log feature does not support read operations.

- If I specify different log retention periods for two instances in the same region that have the new audit log feature enabled, which log retention period is applied to all the instances in the region?

  The last log retention period that you specify is applied.

- Why do I find audit logs whose client IP addresses are not the IP address of the client on which my application runs?

  The audit logs record write operations of the control class. You can filter out this type of information.

Related information

- View audit logs
- Download audit logs
- Subscribe to audit log reports
- Overview of real-time consumption

10.1.2. View audit logs

You can view audit logs within a specific period of time and filter audit logs that match specific conditions.

Prerequisites

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

Background information

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

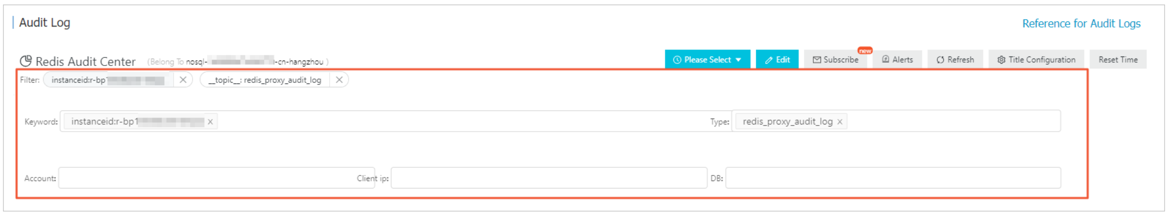
View audit logs

1.
2. In the left-side navigation pane, choose **Logs > Audit Log**.
3. On the **Audit Log** page, check the audit log details of the instance.

Filter the audit logs of an instance

ApsaraDB for Redis allows you to view the audit logs that meet specified filter conditions.

1.
2. In the left-side navigation pane, choose **Logs > Audit Log**.
3. On the **Audit Log** page, you can specify conditions to filter audit logs.



Filter: instanceid:r-bp1[redacted] __topic__: redis_proxy_audit_log

Keyword: instanceid:r-bp1[redacted]

Type: redis_proxy_audit_log

Account:

Client ip:

DB:

Filter conditions

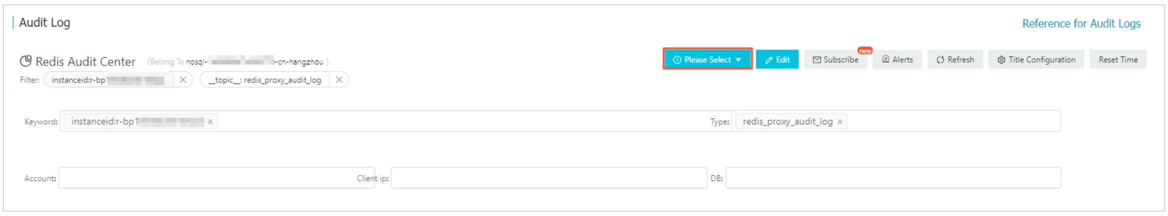
Filter condition	Description
Keyword	The keywords that are included in the audit logs you want to view. A keyword can be a client IP address, a command, a username, or other extended information. Note ○ The Keyword field supports exact match. You must enter complete information in the Keyword field. Examples: ■ If you want to specify an IP address as a keyword, you must enter a complete IP address such as 192.168.1.1, not a partial IP address such as 192.168 or 1.1. ■ If you want to specify a command as a keyword, you must enter a complete command such as AUTH or auth, not a partial command such as au. ○ If a keyword contains a colon (:), you must enclose the keyword in a pair of double quotation marks (""). Example: *"userId:1"*.
Type	The type of audit logs. Valid values: ○ redis_audit_log: the audit logs of data shards. ○ redis_proxy_audit_log: the audit logs of proxy nodes.
Account	The account used to connect to the instance. Default value: null. For more information about accounts, see Create and manage database accounts.
Client IP	The client IP address used to connect to the instance.
DB	The database of which you want to query the audit logs.

View the audit logs of an instance over a specified time range

You can use the time picker to specify a time range to query.

1.

2. In the left-side navigation pane, choose **Logs > Audit Log**.
3. On the **Audit Log** page, click **Please Select**.



Audit Log

Reference for Audit Logs

Redis Audit Center (Belong To nosql-[redacted]-cn-hangzhou)

Filter: instanceid:r-bp[redacted] × __topic__: redis_proxy_audit_log ×

Keyword: instanceid:r-bp1[redacted] ×

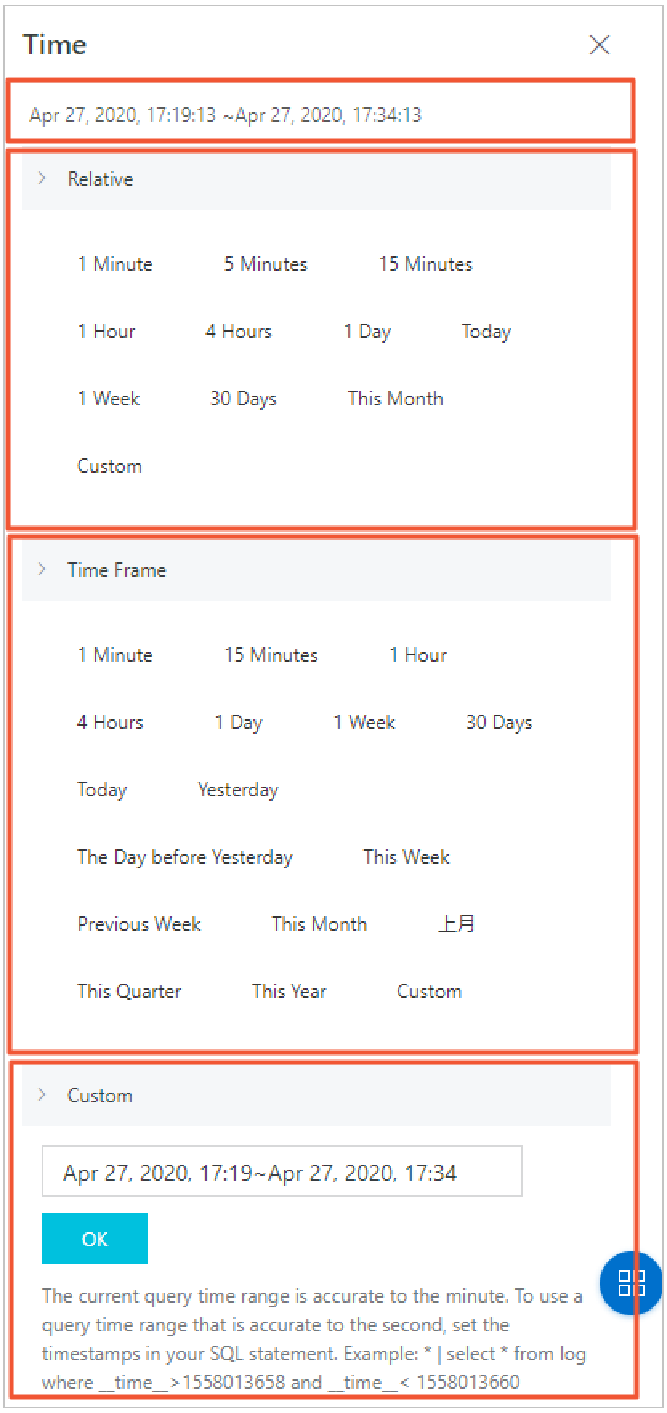
Type: redis_proxy_audit_log ×

Account:

Client ip:

DB:

4. In the Time panel, specify a time range to query audit logs.



Time

Apr 27, 2020, 17:19:13 ~Apr 27, 2020, 17:34:13

Relative

1 Minute 5 Minutes 15 Minutes

1 Hour 4 Hours 1 Day Today

1 Week 30 Days This Month

Custom

Time Frame

1 Minute 15 Minutes 1 Hour

4 Hours 1 Day 1 Week 30 Days

Today Yesterday

The Day before Yesterday This Week

Previous Week This Month 上月

This Quarter This Year Custom

Custom

Apr 27, 2020, 17:19~Apr 27, 2020, 17:34

OK

The current query time range is accurate to the minute. To use a query time range that is accurate to the second, set the timestamps in your SQL statement. Example: * | select * from log where __time__>1558013658 and __time__< 1558013660

Time picker description

Section No.	Section name	Description
①	Time details	When you move the pointer over a time option in the Relative section or Time Frame section, the time details section displays the time range that matches the selected time option.
②	Relative	Select a time range relative to the current point in time. When you move the pointer over a time option in this section, the time details section displays the time range that matches the selected time option.
③	Time Frame	Select a time range that is accurate to the minute, hour, week, or day. When you move the pointer over a time option in this section, the time details section displays the time range that matches the selected time option.
④	Custom	Specify a custom time range. After you click **OK**, the custom time range is applied.

Related API operations

Operation	Description
ModifyAuditLogConfig	Enables or disables the audit log feature and specifies a retention period for audit logs.
DescribeAuditLogConfig	Queries the audit log configurations of an ApsaraDB for Redis instance. The configurations include whether the audit log feature is enabled and the retention period of audit logs.
DescribeAuditRecords	Queries the audit logs of an ApsaraDB for Redis instance.

FAQ

- Can I view more than 2,000 audit log entries?

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

10.1.3. Query historical hotkeys

In an ApsaraDB for Redis instance, the keys that are frequently accessed are called hotkeys. Improper management of hotkeys may cause congestion and degrade service performance. You can use the audit log feature to query and analyze the historical hotkeys. This allows you to further optimize your instance.

Prerequisites

The audit log feature is enabled for the ApsaraDB for Redis instance. For more information about audit logs, see Enable the new audit log feature.

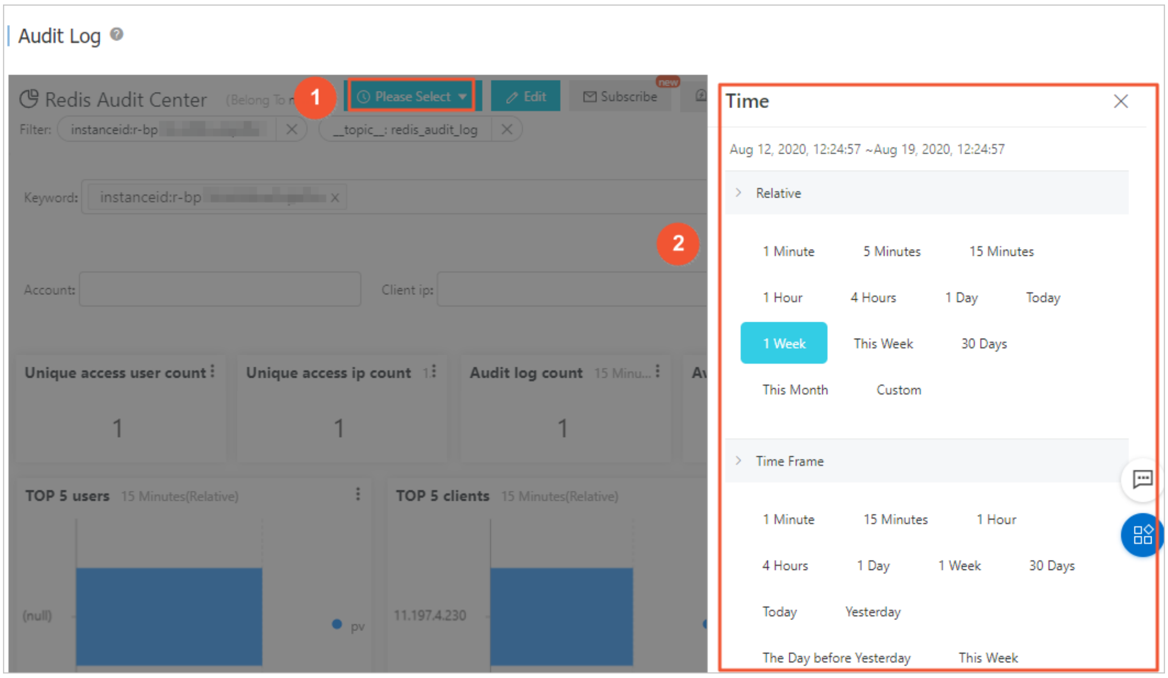
Context

ApsaraDB for Redis uses efficient sorting and statistical algorithms based on the Least Frequently Used (LFU) cache to identify hotkeys in an instance.

> Note If the number of queries per second (QPS) of a key is greater than 3,000, the key is considered to be a hotkey.

Procedure

1.
2. In the left-side navigation pane, choose **Logs > Audit Log**.
3. On the **Audit Log** page, click **Please Select** in the upper-right corner. In the Time panel, specify a time range that you want to query. In this example, **1 Week** is selected, which indicates the last week.



Filter: instanceid:r-bp[redacted] __topic__: redis_audit_log

Keyword: instanceid:r-bp[redacted]

Account:

Client ip:

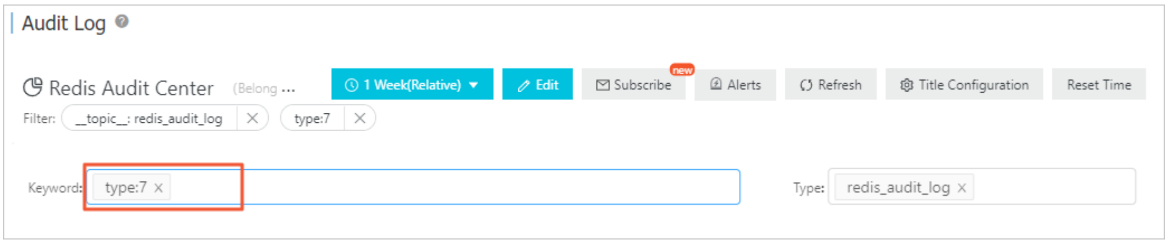
Unique access user count: 1

Unique access ip count: 1

Audit log count: 1

Aug 12, 2020, 12:24:57 ~Aug 19, 2020, 12:24:57

4. Clear the default filter conditions in the **Keyword** field. Enter *type:7* and press the Enter key. The type:7 filter condition is used to query hotkeys.

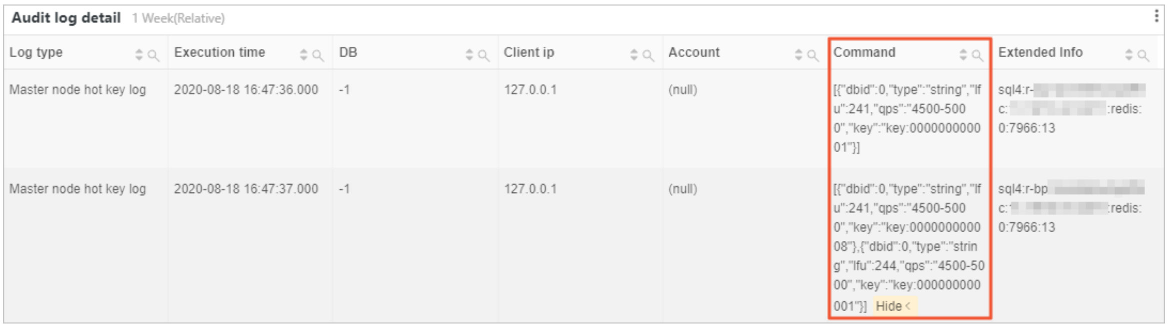


Filter: __topic__: redis_audit_log type:7

Keyword: type:7

Type: redis_audit_log

5. In the **Audit log detail**, view the details of the historical hotkeys.



Audit log detail 1 Week(Relative)

Log type	Execution time	DB	Client ip	Account	Command	Extended Info
Master node hot key log	2020-08-18 16:47:36.000	-1	127.0.0.1	(null)	[{"dbid":0,"type":"string","lfu":241,"qps":"4500-5000","key":"key:000000000001"}]	sql4:r-[redacted] c:[redacted]:redis:0:7966:13
Master node hot key log	2020-08-18 16:47:37.000	-1	127.0.0.1	(null)	[{"dbid":0,"type":"string","lfu":241,"qps":"4500-5000","key":"key:000000000008"},{"dbid":0,"type":"string","lfu":244,"qps":"4500-5000","key":"key:000000000001"}] Hide <	sql4:r-bp[redacted] c:[redacted]:redis:0:7966:13

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

In the **Command** column, view the details of the hotkeys. The following table describes the fields in the hotkey details.

Field	Example	Description
`dbid`	`"dbid":0`	The database in which the hotkey resides.
`type`	`"type":"string"`	The type of data structure that the hotkey uses.
`lfu`	`"lfu":241`	The LFU value of the hotkey.
`qps`	`"qps":"4500-5000"`	The QPS of the hotkey. The value is displayed as a range. **Note** If the QPS of a key reaches 6,000, the system stops calculating the accurate QPS of the key. In this case, **>=6000** is displayed for such a hotkey.
`key`	`"key":"key:000000000008"`	The hotkey.

10.1.4. Download audit logs

This topic describes how to download the audit logs of an ApsaraDB for Redis instance to your computer. You can archive, filter, and analyze the downloaded audit logs.

Prerequisites

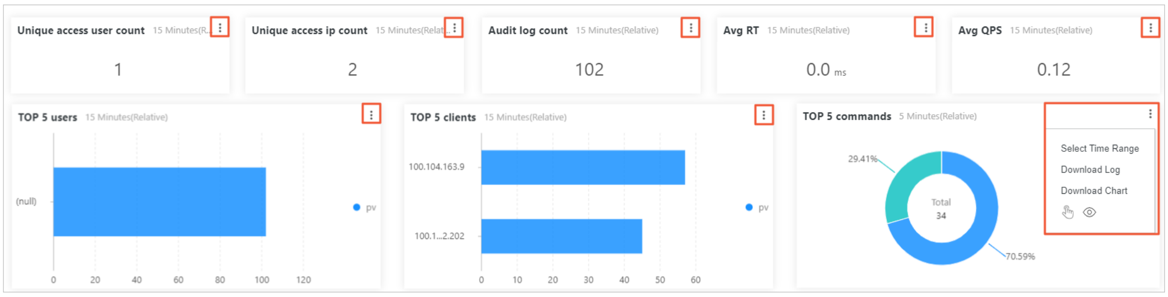
The audit log feature is enabled for the ApsaraDB for Redis instance. For more information about audit logs, see Enable the new audit log feature.

Procedure

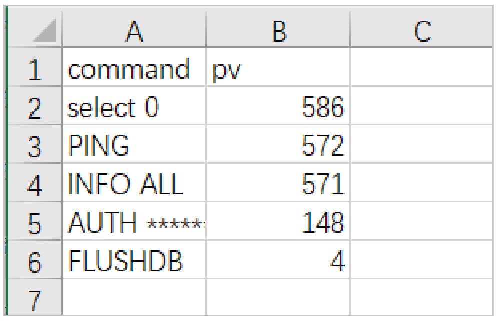
1.
2. In the left-side navigation pane, choose **Logs > Audit Log**.

3. In the log chart section, choose ⋮ > **Download Log** in the upper-right corner of the specified 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. For more information, see Filter the audit logs of an instance.
   > - Filter the log entries of a chart based on the time when the log entries were generated. Click **Select Time Range** above the **Download Log** button to select a time range.


Unique access user count 15 Minutes(R...
1
Unique access ip count 15 Minutes(Relat...
2
Audit log count 15 Minutes(Relative)
102
Avg RT 15 Minutes(Relative)
0.0 ms
Avg QPS 15 Minutes(Relative)
0.12
TOP 5 users 15 Minutes(Relative)
(null)
pv
0
20
40
60
80
100
120
TOP 5 clients 15 Minutes(Relative)
100.104.163.9
100.1...2.202
pv
0
10
20
30
40
50
60
TOP 5 commands 5 Minutes(Relative)
29.41%
Total
34
70.59%
Select Time Range
Download Log
Download Chart

After you click **Download Log**, the log entries that meet the specified criteria are downloaded in the web browser and saved as a *.csv* file to your computer. You can view the file 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			

10.1.5. Subscribe to audit log reports

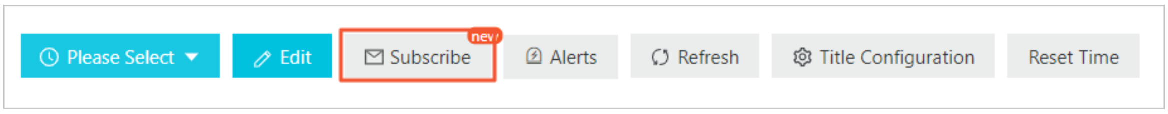
This topic describes how to subscribe to audit log reports of ApsaraDB for Redis by using emails or DingTalk ChatBots. This allows you to periodically check the status of an ApsaraDB for Redis instance.

Prerequisites

The audit log feature is enabled for the ApsaraDB for Redis instance. For more information about audit logs, see Enable the new audit log feature.

Procedure

1.
2. In the left-side navigation pane, choose **Logs** > **Audit Log**.
3. On the **Audit Log** page, click **Subscribe** in the upper-right corner.

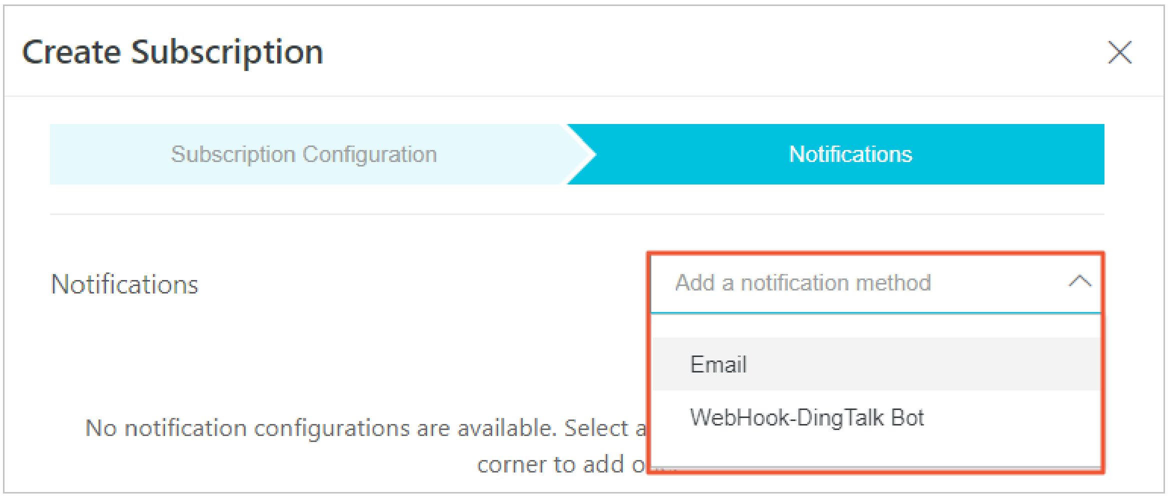


4. On the **Create Subscription** panel, complete the settings and click **Next** in the lower part of the page.

   The following table describes the parameters listed on the page.

Parameter	Description
Subscription Name	The description of the subscription. You can customize the description.
Frequency	The frequency at which ApsaraDB for Redis delivers the reports.
Add Watermark	Specifies whether to enable the watermark feature. After you enable the watermark feature, ApsaraDB for Redis adds watermarks to the images in the audit log reports that are pushed to you. The watermarks can be email addresses or webhook URLs.

5. In the **Notifications** step, click the drop-down list on the right and select a notification method.


Create Subscription
Subscription Configuration
Notifications
Notifications
Add a notification method
Email
WebHook-DingTalk Bot
No notification configurations are available. Select a
corner to add o...

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

> **Note** For more information about how to obtain the WebHook request URL, see DingTalk chatbot webhooks.

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

10.2. View slow logs

This topic describes how to view slow logs collected during a specified period of time in the ApsaraDB for Redis console. You can view slow logs to resolve performance issues and optimize requests.

Prerequisites

An ApsaraDB for Redis instance of one of the following versions is created:

- Performance-enhanced instances
- : 4.0, 5.0, or 6.0

Context

Slow logs record requests that take longer to execute than a specified threshold. Slow logs are classified into slow logs from data nodes and slow logs from proxy nodes.

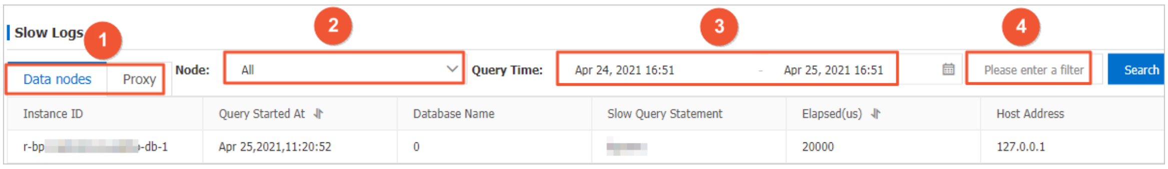
> **Note** Only the slow logs from data nodes are collected for standard instances.

Slow log type	Description	Parameter
Slow logs from data nodes	• The command execution time collected in slow logs that were generated on a data node includes only the amount of time required to actually run a command on the data node. The amount of time required for the data node to communicate with a proxy node or client and the execution latency of the command in the single-threaded queue are not included. • In most cases, the number of slow logs from data nodes is small due to the high-performance capabilities of ApsaraDB for Redis.	• slowlog-log-slower-than: specifies the threshold of command execution time for slow logs from data nodes. If a command runs for a period of time that exceeds this threshold, the command is recorded in a slow log. Default value: 20000. Unit: μs. 20000 μs is equal to 20 ms. **Note** In most cases, the actual latency is higher than the specified value of this parameter because this value does not include the amount of time required to transmit and process data among clients, proxies, and data nodes. • slowlog-max-len: specifies the maximum number of slow log entries that can be stored. Default value: 1024. For more information, see Modify parameters of an instance.

Slow log type	Description	Parameter
Slow logs from proxy nodes	• The command execution time collected in slow logs from proxy nodes starts from the time when a proxy node sends a request to a data node and ends at the time when the proxy node receives the response from the data node. This includes the command execution time on the data node, the data transmission time over the network, and the queuing latency of the command. • Slow logs from proxy nodes are retained for 72 hours. The number of slow logs from proxy nodes allowed to be stored is unlimited. • In most cases, the latency value recorded in a slow log from proxy nodes is closer to the actual latency of the application. As such, we recommend that you check the accuracy of this value when you troubleshoot timeout issues.	rt_threshold_ms: specifies the threshold of command execution time for slow logs from proxy nodes. Default value: 500. Unit: ms. We recommend that you set the threshold to a value close to the client timeout value, which is from 200 ms to 500 ms. For more information, see Modify parameters of an instance.

Procedure

1.
2. In the left-side navigation pane, choose **Logs > Slow Logs**.
3. Specify filter conditions to filter results based on your business requirements.



Node: All

Query Time: Apr 24, 2021 16:51 - Apr 25, 2021 16:51

Please enter a filter

Instance ID	Query Started At	Database Name	Slow Query Statement	Elapsed(us)	Host Address
r-bp[redacted]-db-1	Apr 25,2021,11:20:52	0	[redacted]	20000	127.0.0.1

No.	Description
①	The node type and node ID.
②	Note For standard instances, only the slow logs from **Data Node** are collected. If you use standard instances, skip this step.
③	The time range to query. By default, slow logs collected during the last hour are displayed.
④	A keyword that is specified to filter slow logs. Example: *bgsave*.

> Note
>
> By default, the **Host Address** parameter for master-replica cluster instances and read/write splitting instances displays the IP address of proxy nodes. To obtain the IP address of a specific client, set ptod_enabled to `1` in **System Parameters**. For more information, see Modify parameters of an instance.

Related API operations

API	Description
DescribeSlowLogRecords	Queries slow logs of an ApsaraDB for Redis instance in a specified time period.

References

Use slow logs to troubleshoot timeout issues

10.3. View active logs

This topic describes how to view 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 were generated in the last 72 hours to troubleshoot O&M issues.

Prerequisites

Your instance is of one of the following editions:

- Performance-enhanced instances
- instances that run Redis 4.0 or later.

Procedure

1. Log on to the ApsaraDB for Redis console and go to the Instances page. In the upper-left part, select the region in which the instance is deployed. Then, find the instance and click the instance ID.
2. In the left-side navigation pane, choose **Logs > Active Logs**.
3. On the **Active Logs** page, click the 📅 icon to specify a time range.

   > 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**.

4. Click **Search**.

Related API operations

API	Description
DescribeRunningLogRecords	Queries active logs of an ApsaraDB for Redis instance.

10.4. Use slow logs to troubleshoot timeout issues

A common issue that affects the service performance is connection timeouts caused by slow requests. 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

An ApsaraDB for Redis instance of one of the following versions is created:

- Performance-enhanced instances
- : 4.0, 5.0, or 6.0

Context

Slow logs record requests that take longer to execute than a specified threshold. Slow logs are classified into slow logs from data nodes and slow logs from proxy nodes.

ⓘ **Note** Only the slow logs from data nodes are collected for standard instances.

Slow log type	Description	Parameter
Slow logs from data nodes	• The command execution time collected in slow logs that were generated on a data node includes only the amount of time required to actually run a command on the data node. The amount of time required for the data node to communicate with a proxy node or client and the execution latency of the command in the single-threaded queue are not included. • In most cases, the number of slow logs from data nodes is small due to the high-performance capabilities of ApsaraDB for Redis.	• slowlog-log-slower-than: specifies the threshold of command execution time for slow logs from data nodes. If a command runs for a period of time that exceeds this threshold, the command is recorded in a slow log. Default value: 20000. Unit: μs. 20000 μs is equal to 20 ms. ⓘ **Note** In most cases, the actual latency is higher than the specified value of this parameter because this value does not include the amount of time required to transmit and process data among clients, proxies, and data nodes. • slowlog-max-len: specifies the maximum number of slow log entries that can be stored. Default value: 1024. For more information, see Modify parameters of an instance.

Slow log type	Description	Parameter
Slow logs from proxy nodes	- The command execution time collected in slow logs from proxy nodes starts from the time when a proxy node sends a request to a data node and ends at the time when the proxy node receives the response from the data node. This includes the command execution time on the data node, the data transmission time over the network, and the queuing latency of the command. - Slow logs from proxy nodes are retained for 72 hours. The number of slow logs from proxy nodes allowed to be stored is unlimited. - In most cases, the latency value recorded in a slow log from proxy nodes is closer to the actual latency of the application. As such, we recommend that you check the accuracy of this value when you troubleshoot timeout issues.	rt_threshold_ms: specifies the threshold of command execution time for slow logs from proxy nodes. Default value: 500. Unit: ms. We recommend that you set the threshold to a value close to the client timeout value, which is from 200 ms to 500 ms. For more information, see Modify parameters of an instance.

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 SLOWLOG subcommand [argument]. - Log on to the ApsaraDB for Redis console or call an API operation: ○ View slow logs ○ DescribeSlowLogRecords
Slow logs from proxy servers	Log on to the ApsaraDB for Redis console or call an API operation: - View 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:

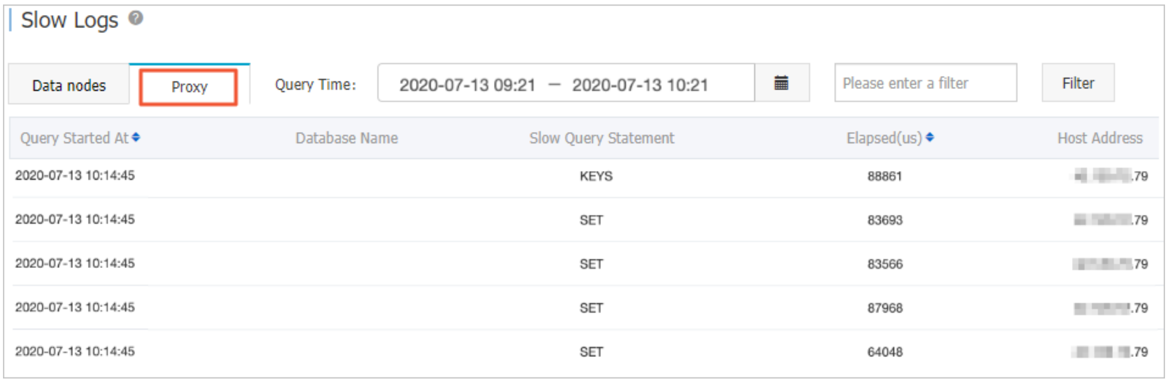
1. If a service timeout issue occurs, first check the slow logs generated on proxy servers. For more information, see View slow logs.

> Note
>
> - For standard instances, go to Step 3 and analyze slow logs from data nodes.
> - If no slow logs from proxy servers exist, you can check the network between the client and the ApsaraDB for Redis instance.

2. Find the command recorded by the earliest slow log from proxy servers.

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

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.

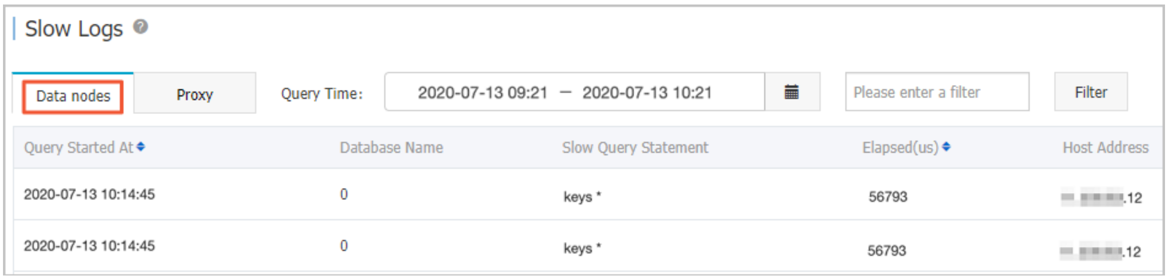


Query Started At	Database Name	Slow Query Statement	Elapsed(us)	Host Address
2020-07-13 10:14:45		KEYS	88861	[redacted].79
2020-07-13 10:14:45		SET	83693	[redacted].79
2020-07-13 10:14:45		SET	83566	[redacted].79
2020-07-13 10:14:45		SET	87968	[redacted].79
2020-07-13 10:14:45		SET	64048	[redacted].79

3. Check the slow logs from data nodes to find the slow logs from proxy servers that cause the timeout issue.

> Note Typically, the command that first generates slow logs in slow logs from proxy servers can also generate slow logs on data nodes. The number of slow logs from a data node is usually less than that of a 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 slow logs from proxy servers, you can see that the slow log caused by the **KEYS** command also exists in slow logs from data nodes. 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.



Query Started At	Database Name	Slow Query Statement	Elapsed(us)	Host Address
2020-07-13 10:14:45	0	keys *	56793	[redacted].12
2020-07-13 10:14:45	0	keys *	56793	[redacted].12

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

11.Backup and recovery

11.1. Backup and restoration solutions

ApsaraDB for Redis offers high performance, high security, high availability, and diverse architectures. An increasing number of applications run ApsaraDB for Redis as their database engine for persistent storage. ApsaraDB for Redis provides a variety of solutions that allow you to back up or restore data in different scenarios.

Persistence solutions

ApsaraDB for Redis supports the following persistence solutions:

- **RDB persistence**

  ApsaraDB for Redis creates snapshots on a regular basis for the data stored in the engine storage, generates Redis database (RDB) files, and then saves the files to disks. This process is called RDB persistence. RDB files are small in size and easy to migrate. 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 seven days.

- **AOF persistence**

  ApsaraDB for Redis records all commands that write data, such as **SET** in logs. This process is called append-only file (AOF) persistence. When you restart an ApsaraDB for Redis instance, the system reruns the commands in the AOFs to restore data. If AOFs are larger than required, open source Redis runs an AOF rewrite task to recreate the AOFs at a reduced file size.

  You can specify the AOF_FSYNC_EVERYSEC policy to enable AOF persistence for ApsaraDB for Redis instances. After you specify this policy, the system records all write commands in an AOF every second and saves the AOF to disks. The policy has a negligible impact on the performance and can minimize data loss caused by accidental operations.

- **AOF persistence for performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair)**

  ApsaraDB for Redis Enhanced Edition (Tair) supports data backup and restoration based on RDB snapshots and optimizes AOF persistence. After optimization, AOFs can be archived incrementally to prevent performance degradation caused by AOF rewrite. Incremental archive also allows data in an instance or a key to be restored to a point in time accurate to the second as this method saves each write operation and its timestamp. For more information, see Use data flashback to restore data by point in time.

Backup and restoration solutions

Category	Solution implementation	Description

Category	Solution implementation	Description
Data backup	Automatic or manual backup	You can use ApsaraDB for Redis to persist data. Backups are automatically created based on the default backup policy. You can modify the automatic backup policy or manually create a temporary backup. RDB
	Download a backup file	Backup files of ApsaraDB for Redis are retained free of charge for seven days. If you want to retain backup files for more than seven days, you can download the backup files to your computer. For example, you may want to retain data for more than seven days due to regulatory or security requirements.
	Use the redis-shake tool to back up data	You can use redis-shake in dump mode to back up the data of an ApsaraDB for Redis instance in an RDB file and store the RDB file in your computer.
Data restoration	Restore data from a backup set to a new instance	ApsaraDB for Redis allows you to create an instance from a specified backup set. The data in the new instance is the same as that in the backup set. This feature is suitable for scenarios such as data restoration, quick workload deployment, and data verification.
	Use data flashback to restore data by point in time	After you enable the data flashback feature, you can restore data of an ApsaraDB for Redis instance to a specified point in time accurate to the second. This feature minimizes data loss caused by accidental operations and is suitable for scenarios in which data is frequently restored. **Note** This feature is supported only by performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances. AOF
	Use redis-shake to restore data	You can use redis-shake in restore mode to restore data from an RDB file to an ApsaraDB for Redis instance.

11.2. Automatic or manual backup

ApsaraDB for Redis supports data persistence. Backups are created based on the default backup policy. You can modify the policy as your needs change. You can also manually create temporary backups.

Context

By default, ApsaraDB for Redis uses Redis database (RDB) snapshots to persist data. Data stored in memory is persisted to disks at points in time. Data retrieval is not affected when ApsaraDB for Redis creates backups.

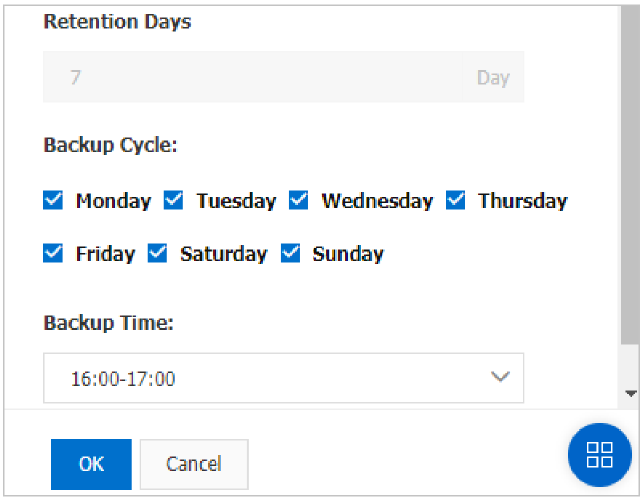
> Note ApsaraDB for Redis optimizes data persistence based on Append Only Files (AOFs). ApsaraDB for Redis archives incremental data so that data can be restored within seconds and O&M efficiency can be improved. For more information, see Use data flashback to restore data by point in time. AOF

Precautions

When an instance backup is being created, another backup cannot be created. We recommend that you create another backup after the first backup is generated.

Procedure

1.
2. In the left-side navigation pane, click **Backup and Recovery**.
3. Perform the operations that are described in the following table based on your business requirements.

Operation	Procedure
Modify the automatic backup policy	i. In the upper-right corner, click **Backup Settings**. ii. In the panel that appears, specify Backup Cycle and Backup Time. Configure an automatic backup policy  **Retention Days** 7 Day **Backup Cycle:** ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday ☑ Sunday **Backup Time:** 16:00-17:00 - **Retention Days**: The number of days for which to retain backups. This parameter is automatically set to 7 and cannot be changed. - **Backup Cycle**: The backup cycle. You can select one or more days of a week on which to back up data. By default, one backup is created per day. - **Backup Time**: The period of time to back up data. You can specify one or more hours of a day during which to back up data. We recommend that you back up data during off-peak hours. iii. Click **OK**.

Operation	Procedure
Manually create a temporary backup	i. Click **Create Backup** in the upper-right corner. ii. In the message that appears, click **OK**.

Related API operations

Operation	Description
CreateBackup	Manually creates a database backup for an ApsaraDB for Redis instance.
DescribeBackupPolicy	Queries the backup policy of an ApsaraDB for Redis instance. The backup policy includes the backup cycle and backup time.
ModifyBackupPolicy	Modifies the backup policy of an ApsaraDB for Redis instance.

Related information

- Restore data from a backup set to a new instance
- Use data flashback to restore data by point in time
- Download a backup file

11.3. Download a backup file

Backup files of ApsaraDB for Redis are retained for seven days free of charge. If you want to retain backup files for more than seven days due to regulatory or security requirements, you can download the backup files to your computer. You can also restore downloaded backup files to a self-managed database. This way, you can perform data analytics or run tests by using the restored backup files in the self-managed database.

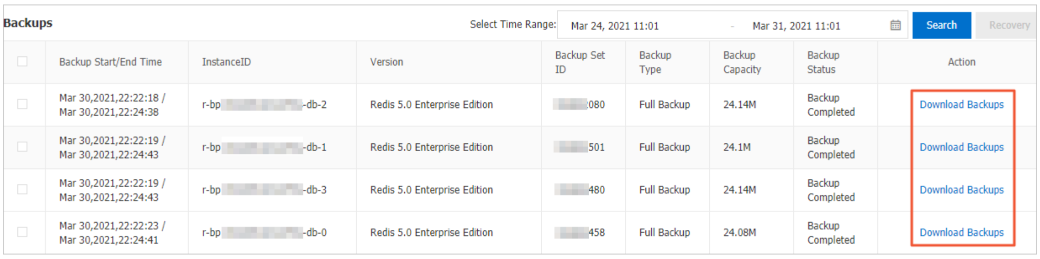
Prerequisites

The instance is an instance of the ApsaraDB for Redis Community Edition or a performance-enhanced or persistent memory-optimized instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances and persistent memory-optimized instances, see Performance-enhanced instances and Persistent memory-optimized instances.

Procedure

1.
2. In the left-side navigation pane, click **Backup and Recovery**.
3. Find the backup file that you want to download and click **Download Backups** in the **Action** column.

   Download a backup file



Backups Select Time Range: Mar 24, 2021 11:01 - Mar 31, 2021 11:01

□	Backup Start/End Time	InstanceID	Version	Backup Set ID	Backup Type	Backup Capacity	Backup Status	Action
□	Mar 30,2021,22:22:18 / Mar 30,2021,22:24:38	r-bp[redacted]-db-2	Redis 5.0 Enterprise Edition	[redacted]080	Full Backup	24.14M	Backup Completed	Download Backups
□	Mar 30,2021,22:22:19 / Mar 30,2021,22:24:43	r-bp[redacted]-db-1	Redis 5.0 Enterprise Edition	[redacted]501	Full Backup	24.1M	Backup Completed	Download Backups
□	Mar 30,2021,22:22:19 / Mar 30,2021,22:24:43	r-bp[redacted]-db-3	Redis 5.0 Enterprise Edition	[redacted]480	Full Backup	24.14M	Backup Completed	Download Backups
□	Mar 30,2021,22:22:23 / Mar 30,2021,22:24:41	r-bp[redacted]-db-0	Redis 5.0 Enterprise Edition	[redacted]458	Full Backup	24.08M	Backup Completed	Download Backups

> **Note** If the instance is a cluster instance or read/write splitting instance, you must download a backup file of each data node in the instance to ensure data consistency. For more information about cluster instances and read/write splitting instances, see Cluster master-replica instances and Read/write splitting instances.

4. In the dialog box that appears, perform one of the operations that are described in the following table.

Operation	Procedure
Download the backup file over the Internet	i. Click **Get URL for Internet**. ii. Paste the URL into the address bar of a browser and press the Enter key. The browser downloads the backup file.

Operation	Procedure

Download the backup file over an internal network. For example, download the backup file on an Elastic Compute Service (ECS) instance.	i. Click **Get URL for Intranet**.
	ii. Select a download method based on the operating system of your computer.
	■ Windows: Paste the URL into the address bar of a browser and press the Enter key. The browser downloads the backup file.
	■ Linux: Run the command in the following format:
	`wget -c '<URL that is used to download the backup file over the internal network>' -O <Name of the downloaded backup file>.<Suffix of the downloaded backup file>`
	Example:
	`wget -c 'http://rds****.oss-cn-hangzhou-internal.aliyuncs.com/custins416****/hins1****.rdb?...' -O backupfile.rdb`
	Note If you download the backup file on an ECS instance, the ECS instance and the ApsaraDB for Redis instance can be deployed in networks of different types.

What to do next

After the backup file is downloaded, you can restore the downloaded backup file to a self-managed database. This way, you can perform data analytics or run tests by using the restored backup files in the self-managed database. For more information, see Use redis-shake to restore data.

Related API operations

Operation	Description
DescribeBackups	Queries the information and URL of a backup file of an ApsaraDB for Redis instance.

11.4. Restore data from a backup set to a new instance

If you restore data from a backup set to a source ApsaraDB for Redis instance, the data in the source instance is overwritten and cannot be restored. As a result, data loss may occur. ApsaraDB for Redis allows you to create an instance from a specified backup set. The data in the new instance is the same as that in the backup set. This feature can be applied in scenarios such as data recovery, quick workload deployment, and data verification.

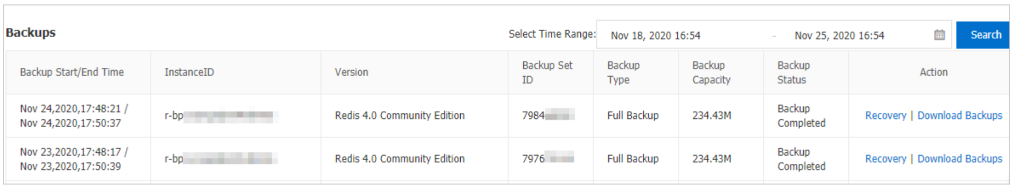
Billing

When you perform the operations described in this topic, an ApsaraDB for Redis instance is created, and you are charged for the ApsaraDB for Redis instance. For more information, see Billable items and prices.

Procedure

1.
2. In the left-side navigation pane, click **Backup and Recovery**.
3. Restore backup sets.
    - Standard master-replica instances: Find the backup set that you want to manage and click **Recovery** in the Action column.

      Restore a standard instance

      Backups

      Select Time Range: Nov 18, 2020 16:54 - Nov 25, 2020 16:54 Search

Backup Start/End Time	InstanceID	Version	Backup Set ID	Backup Type	Backup Capacity	Backup Status	Action
Nov 24,2020,17:48:21 / Nov 24,2020,17:50:37	r-bp[redacted]	Redis 4.0 Community Edition	7984[redacted]	Full Backup	234.43M	Backup Completed	Recovery | Download Backups
Nov 23,2020,17:48:17 / Nov 23,2020,17:50:39	r-bp[redacted]	Redis 4.0 Community Edition	7976[redacted]	Full Backup	234.43M	Backup Completed	Recovery | Download Backups

    - Cluster master-replica instances and read/write splitting instances: Select all the backup sets of a specific point in time and click **Recovery**.

    > Note You can also filter backup sets by time range.

4. In the message that appears, read the content and click **OK**.
5. On the **Clone Instance** page, configure the parameters described in the following table.

Parameter	Description
Source Type	Set the value to **Backup Set**.
Clone Backup Set	Select the backup set that you want to manage from the drop-down list.
Region	The region where the source ApsaraDB for Redis instance is deployed is automatically selected and cannot be changed.
Zone	The zone in which you want to create the instance. Each region has multiple isolated locations known as zones. Each zone has its own independent power supply and network. To minimize the network latency between an Elastic Compute Service (ECS) instance and an ApsaraDB for Redis instance that are deployed in the same zone, connect them over an internal network. Note To implement zone-disaster recovery, you can deploy the ApsaraDB for Redis instance across multiple zones in the same region.

Parameter	Description
Network Type	○ **VPC** (recommended): A virtual private cloud (VPC) is an isolated network with higher security and better performance than the classic network. We recommend that you select the VPC network type. ○ **Classic Network**: Cloud services in the classic network are not isolated. Unauthorized access to a cloud service is blocked only by using security groups or whitelists. Notice ○ The ApsaraDB for Redis instance must be of the same network type as the ECS instance that you want to connect. Otherwise, these instances cannot communicate over an internal network. ○ If the network type for both the ECS instance and the ApsaraDB for Redis instance is VPC, these instances must be deployed in the same VPC. Otherwise, they cannot communicate with each other over the internal network. ○ You can switch the network type of an ApsaraDB for Redis instance from the classic network to VPC. However, you cannot switch the network type of an ApsaraDB for Redis instance from VPC to classic network. For more information, see Change the network type from classic network to VPC.
VPC	The VPC in which you want to create the instance. If you do not have a VPC, create one first. For more information, see Create and manage a VPC.
VSwitch	The vSwitch that you want to connect to the instance in the VPC. If no vSwitches are available in the VPC in the current zone, create a vSwitch. For more information, see Work with vSwitches.
Edition	○ **Community Edition**: This edition is compatible with the Redis protocol and provides database solutions that use both memory and disks. ○ **Enhanced Edition (Tair)**: This edition is developed based on the ApsaraDB for Redis Community Edition and is optimized in performance, storage, and data structures. For more information, see Overview.
Series	**Performance-enhanced**: uses a multi-threading model. This parameter is available only if **Edition** is set to **Enhanced Edition (Tair)**. The performance of this series type is three times that of a Community Edition instance of the same specifications. This series type also provides multiple data structure modules to simplify development. For more information, see Performance-enhanced instances.
Version	The major version of the ApsaraDB for Redis database engine. Note The instances that run Redis 2.8 will soon be phased out. We recommend that you create an instance that uses the latest engine version for more features and more stable performance.

Parameter	Description
Architecture Type	○ **Cluster**: eliminates the performance bottleneck that is caused by the single-threading model. You can use the high-performance cluster instance to process large-capacity workloads. ○ **Standard**: runs in a master-replica architecture, provides high-performance caching services, and ensures high data reliability. ○ **Read/Write Splitting**: ensures high availability (HA) and high performance and supports multiple specifications. The read/write splitting architecture allows a large number of concurrent reads of hot data from read replicas. This reduces the loads on the master node and minimizes O&M costs. For more information, see Overview.
Shards	The number of data shards for the cluster instance. Data is distributed across the data shards in the cluster instance. **Note** This parameter is available only if the **Architecture Type** parameter is set to **Cluster**.
Node Type	○ If you set the **Architecture Type** parameter to **Cluster** or **Standard**, you must set the Node Type parameter to **Master-Replica**. This creates a dual-node hot-standby architecture that provides HA. ○ If you set the **Architecture Type** parameter to **Read-Write Splitting**, you can select the node type based on the number of read replicas.
Instance Class	Each instance type contains a group of specifications. An instance type includes the memory capacity, maximum number of concurrent connections, and maximum bandwidth items. For more information, see Overview. **Note** Database metadata is generated when an ApsaraDB for Redis instance is created. The size of the metadata on each shard of a cluster instance ranges from 30 MB to 50 MB. The total size of the metadata for a cluster instance equals the total size of metadata on all shards of the cluster instance.
Set Password	○ **Later**: Set a password after the instance is created. For more information, see Change or reset the password. ○ **Now**: Specify a password for the instance. ■ The password must be 8 to 32 characters in length. ■ The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. ■ Special characters include `! @ # $ % ^ & * ( ) _ + - =`
Instance Name	The name of the instance, which is used to identify and manage the instance.

Parameter	Description
Duration	The subscription duration for the instance. You can select one to nine months for a monthly subscription or select one to three years for an annual subscription. ? **Note** This parameter is available only to subscription instances.

6. Read and select ApsaraDB for Redis (Subscription) Agreement of Service.
7. Click **Buy Now**.
   After the payment is complete, wait for 1 to 5 minutes. Then, you can find the new ApsaraDB for Redis instance in the ApsaraDB for Redis console.

What to do next

After the new instance is created, you can connect to the new instance to verify data. If the instance passes the verification, you can use the new instance to restore your workloads. If you no longer need the source instance, you can release the instance to save resources. For more information, see Release 或退订instances.

Related API operations

Operation	Description
CreateInstance	Creates an instance and restores data from a specified backup set to the instance.

11.5. Use data flashback to restore data by point in time

After the data flashback feature is enabled, you can restore the data of an instance or a specified key to a point in time that is accurate to seconds from an unexpired backup file. You can restore data to a new instance or the original instance. Such a refined data restoration capability can protect business data and prevent data loss that is caused by accidental operations to the greatest extent.

Prerequisites

- The instance is a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances.
- The instance uses the standard architecture or the cluster architecture. For more information about standard instances and cluster instances, see Standard master-replica instances and Cluster master-replica instances.
- The minor version of the instance is the latest. For more information, see Update the minor version.

> ? **Note** If the **Upgrade Minor Version** button is dimmed, the minor version is the latest.

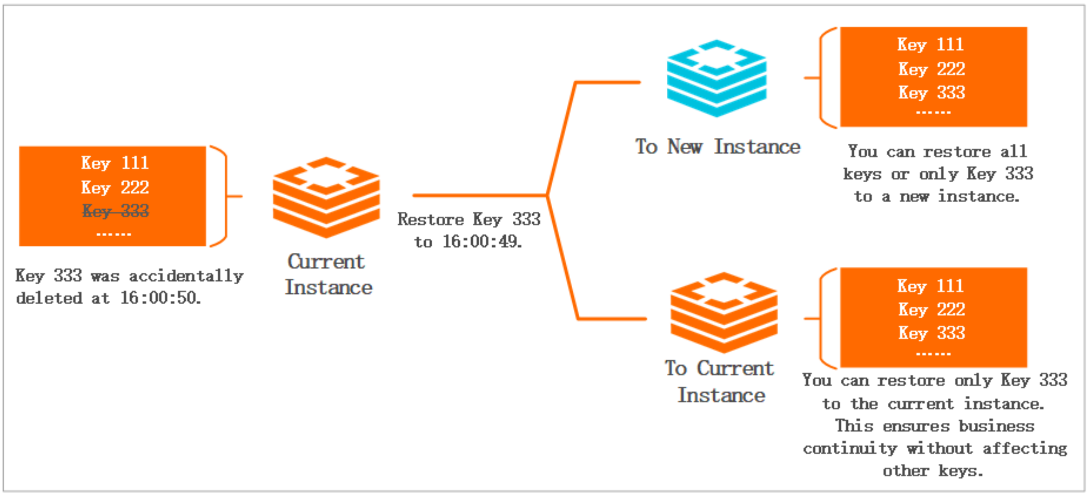
Data flashback introduction

To protect your business data in the cloud, ApsaraDB for Redis Enhanced Edition (Tair) provides the data flashback feature in addition to the data backup and restoration features based on Redis Database (RDB) snapshots. ApsaraDB for Redis Enhanced Edition (Tair) optimizes the persistence mechanism based on append-only-files (AOFs) and incrementally archives AOFs so that data can be restored to a point in time accurate to seconds. This facilitates O&M and allows you to use ApsaraDB for Redis Enhanced Edition (Tair) for persistent storage.

After the data flashback feature is enabled, you can restore the data of an instance or a specified key to a point in time that is accurate to seconds from an unexpired backup file. The maximum retention period of backup files is seven days. You can restore data to a new instance or the original instance. Such a refined data restoration capability can protect business data and prevent data loss that is caused by accidental operations to the greatest extent.

> **Note** If you restore data to the original instance, the key that you want to restore is written back to the original instance, which may increase the queries per second (QPS) or latency of the original instance. We recommend that you restore data during off-peak hours.

Two modes of data flashback


Key 111
Key 222
~~Key 333~~
......
Key 333 was accidentally deleted at 16:00:50.
Current Instance
Restore Key 333 to 16:00:49.
To New Instance
Key 111
Key 222
Key 333
......
You can restore all keys or only Key 333 to a new instance.
To Current Instance
Key 111
Key 222
Key 333
......
You can restore only Key 333 to the current instance. This ensures business continuity without affecting other keys.

Limits

- You can restore data only to a specific point in time that is in the time range from when data flashback is enabled to the current time. This period can be up to seven days.
- After the data flashback feature is enabled, the point in time for data backup may be changed due to configuration changes, cross-zone migrations, or minor version upgrades. For example, if you change the configurations of an instance, the point in time that you can restore data to starts from the time when the configuration change is complete.

  > **Note** If you change the architecture of an instance (such as from standard to cluster), date flashback is disabled for the instance. To use data flashback, you must re-enable the feature.

- After you enable the data flashback feature, it requires about 10 minutes for the system to upload data and logs.

Billing

During the trial period of the data flashback feature, you can restore data to a point in time within the last seven days free of charge. After the official release, this feature is charged based on points in time of restoration. For more information, see this topic or the announcement on the Alibaba Cloud website.

> **Note** If you use the data flashback feature to restore data to a new instance, the system creates an instance and restores data to the instance. You must pay for the new instance. You can set the billing method of the new instance to pay-as-you-go and release the instance after it is no longer needed. For more information, see Billable items and prices.

Enable the data flashback feature

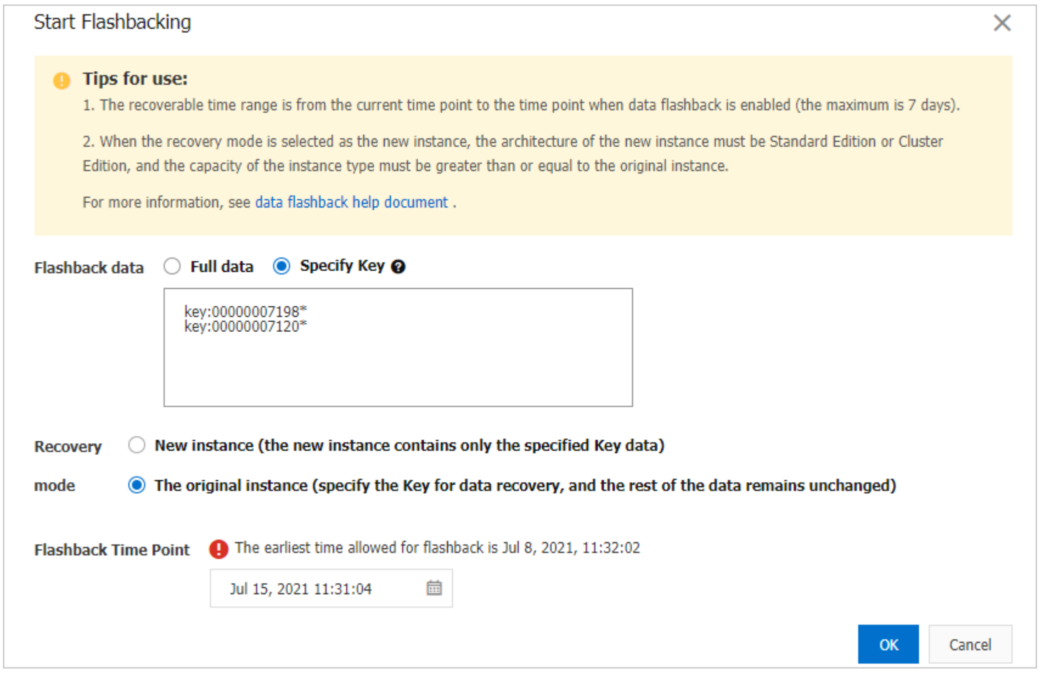
1.
2. In the left-side navigation pane, click **Backup and Recovery**.
3. On the **Backup and Recovery** page, click the **Data Flashback** tab.
4. Click **Enable Now**.

   > **Note** If the appendonly parameter is set to *no*, AOF persistence is disabled. In the dialog box that appears, you are prompted to enable the AOF feature. For more information about how to enable the AOF feature, see Modify parameters of an instance.

Perform data flashback

1.
2. In the left-side navigation pane, click **Backup and Recovery**.
3. On the **Backup and Recovery** page, click the **Data Flashback** tab.
4. Click **Start Flashbacking**. In the dialog box that appears, set the flashback parameters.

   Perform data flashback



Start Flashbacking

Tips for use:

1. The recoverable time range is from the current time point to the time point when data flashback is enabled (the maximum is 7 days).

2. When the recovery mode is selected as the new instance, the architecture of the new instance must be Standard Edition or Cluster Edition, and the capacity of the instance type must be greater than or equal to the original instance.

For more information, see data flashback help document .

Flashback data ○ Full data ● Specify Key

```
key:00000007198*
key:00000007120*
```

Recovery mode ○ New instance (the new instance contains only the specified Key data)
● The original instance (specify the Key for data recovery, and the rest of the data remains unchanged)

Flashback Time Point The earliest time allowed for flashback is Jul 8, 2021, 11:32:02

Jul 15, 2021 11:31:04

Parameter	Description
Flashback data	○ **Full data**: All data on the instance is restored. ○ **Specify Key**: Specify one or more keys whose data you want to restore. Each key name occupies a line. You can specify regular expressions based on the following rules: ■ Period (.): matches a single character except `'\r\n'`. ■ Asterisk (*): matches zero or more occurrences of a preceding subexpression. For example, `h.*llo` matches `hllo` or `heeeello`. ■ Question mark (?): matches zero or one occurrence of a preceding subexpression. For example, `h.?llo` matches `hllo` or `hello`. ■ Character set [Characters]: matches a character included in the brackets ([ ]). For example, `h[ae]llo` matches `hallo` or `hello`. ■ Negative character set [^Characters]: does not match a character in the brackets ([ ]). For example, `h[^ae]llo` matches `hcllo` or `hdllo`, but not `hallo` or `hello`. ■ Character range [Character1-Character2]: matches a character in the range of `Character1 to Character2`. For example, `h[a-b]llo` matches `hallo` and `hbllo`. **Note** To ensure the efficiency of data restoration, we recommend that you specify no more than 10 keys or specify no more than three regular expressions.
Restoration mode	○ **New instance**: restores data to a new instance. ○ **The original instance**: restores data to the current instance. If you set the **Flashback data** parameter to **Specify Key**, only the data of the specified keys can be restored to the specified point in time and other keys are not affected.
Flashback Time Point	The point in time to which you want to restore data. **Note** If you set the **Recovery mode** parameter to **The original instance**, you must specify this parameter. If you set the **Recovery mode** parameter to **New instance**, the **Clone Instance** page appears. You must specify the point in time to which you want to restore data on this page.

5. Click **OK**.
    - If you set the **Recovery mode** parameter to **The original instance**, the current instance enters the **Restoring** state. Wait until the instance state changes to **Running**.
    - If you set the **Recovery mode** parameter to **New instance**, the **Clone Instance** page appears. You must specify the point in time to which you want to restore data and the configurations of the new instance on this page.

Note The architecture of the new instance must be standard or cluster, and the capacity of the new instance must be greater than or equal to that of the original instance. For more information about instance parameters, see Create Redis本地盘实例an ApsaraDB for Redis Community Edition instance or a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair).

Related API operations

Operation	Description
RestoreInstance	Restores the data of an ApsaraDB for Redis instance from a backup file to the instance. If you use this operation together with the data flashback feature, you can restore data of a specified key to a specified point in time that is accurate to seconds.

11.6. Use redis-port to restore data

This topic describes how to use redis-port to restore data. You can restore the data of an ApsaraDB for Redis instance in the ApsaraDB for Redis console. You can run the script that is provided in this topic to download a backup of the instance. The script uses redis-port to restore data from the backup.

Prerequisites

- The engine version of the source ApsaraDB for Redis instance is Redis 2.8 or 4.0.
- The engine version of the destination ApsaraDB for Redis instance is 2.8, 4.0, or 5.0.

Context

redis-port is an open source tool that can be used to migrate data between Redis databases. The tool can run multiple data migration tasks at the same time. In this example, redis-port is used in restore mode to restore the data of an ApsaraDB for Redis instance to a specified ApsaraDB for Redis instance.

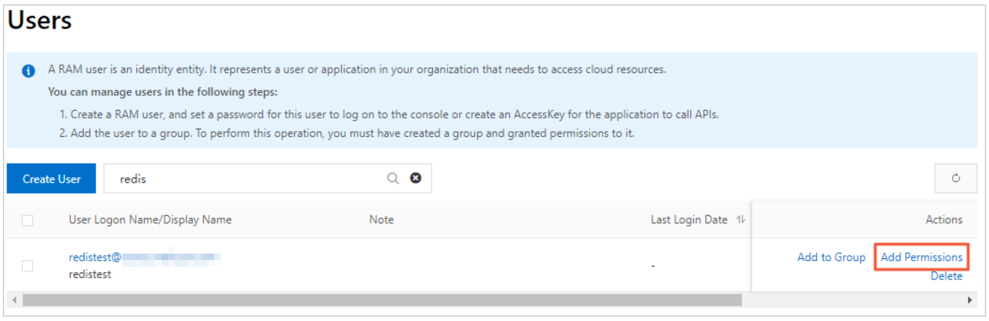
Prerequisites

Notice The script provided in this example requires you to provide an AccessKey pair to complete the authentication before you can obtain the backup file of the instance. To avoid disclosing the AccessKey pair of your Alibaba Cloud account, we recommend that you create a Resource Access Management (RAM) user, authorize the RAM user, and create an AccessKey pair for the RAM user.

1. Create a RAM user and grant the **AliyunKvstoreFullAccess** permission to the RAM user. This permission allows the RAM user to manage ApsaraDB for Redis instances.
   i. Log on to the RAM console.
   ii. Create a RAM user.
   iii. In the left-side navigation pane, choose **Identities > Users**.

iv. On the Users page, find the RAM user to which you want to grant permissions and click **Add Permissions** in the **Actions** column.

Add permissions



Users

A RAM user is an identity entity. It represents a user or application in your organization that needs to access cloud resources.

You can manage users in the following steps:

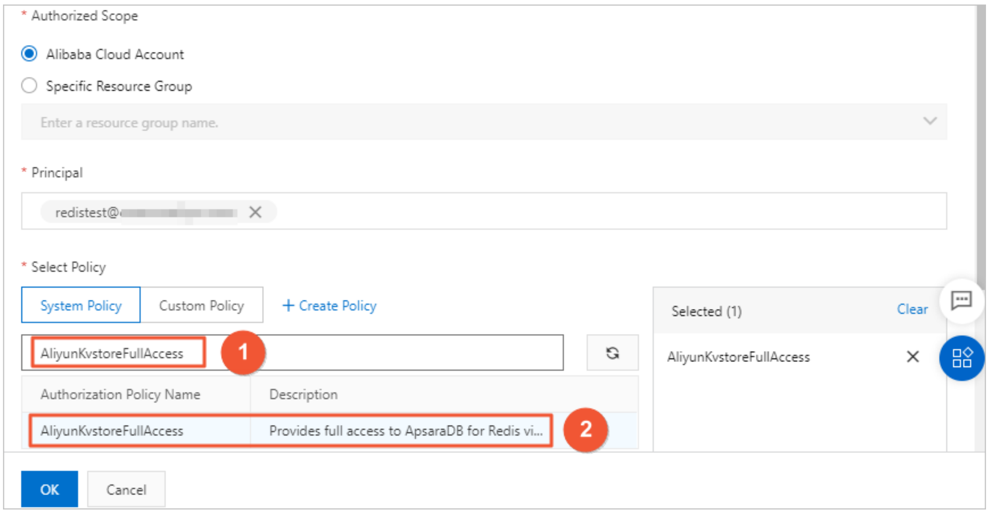
1. Create a RAM user, and set a password for this user to log on to the console or create an AccessKey for the application to call APIs.
2. Add the user to a group. To perform this operation, you must have created a group and granted permissions to it.

Create User | redis

	User Logon Name/Display Name	Note	Last Login Date	Actions
□	redistest@[redacted] redistest		-	Add to Group Add Permissions Delete

v. In the dialog box that appears, enter *AliyunKvstoreFullAccess* in the search box and click the policy name to add the policy to the Selected section.

Grant the AliyunKvstoreFullAccess permission



* Authorized Scope

● Alibaba Cloud Account

○ Specific Resource Group

Enter a resource group name.

* Principal

redistest@[redacted] ✕

* Select Policy

System Policy | Custom Policy | + Create Policy

AliyunKvstoreFullAccess

Authorization Policy Name	Description
AliyunKvstoreFullAccess	Provides full access to ApsaraDB for Redis vi...

Selected (1) Clear

AliyunKvstoreFullAccess ✕

vi. Click **OK**.

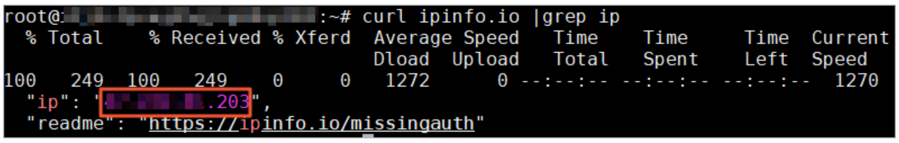
vii. Click **Complete**.

2. Create an AccessKey pair for the RAM user. For more information, see Create an AccessKey pair.

Procedure

1. Perform the following operations based on the location where redis-port is installed:

> **Notice**
>
> - The ApsaraDB for Redis instances that are mentioned in the following table refer to the source and destination ApsaraDB for Redis instances. Perform the following operations on the source and destination ApsaraDB for Redis instances.
> - We recommend that you install redis-port on Elastic Compute Service (ECS) instances. You can connect to the source and destination ApsaraDB for Redis instances through a virtual private cloud (VPC) to achieve lower network latency and higher security.

<table>
<tr><th>Location where redis-port is installed</th><th>Operation</th></tr>
<tr><td>ECS instances</td><td>
i. Make sure that the ECS instance and the ApsaraDB for Redis instance are deployed in the same VPC. In this case, the same VPC ID is displayed in the Basic Information section of the instances.
Note
- If the instances are deployed in different VPCs, you can change the VPC to which the ECS instance belongs. For more information, see Change the VPC of an ECS instance.
- The network types of the ECS instance and the ApsaraDB for Redis instance may be different. For example, the ECS instance belongs to the classic network and the ApsaraDB for Redis instance belongs to a VPC. For information about how to connect to an ApsaraDB for Redis instance from an ECS instance when the instances are deployed in different types of networks, see Connect an ECS instance to an ApsaraDB for Redis instance in different types of networks.
ii. Obtain the internal IP address of the ECS instance. For more information, see Network FAQ.
iii. Add the internal IP address of the ECS instance to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.
</td></tr>
<tr><td>On-premises machine</td><td>
i. By default, only internal endpoints are available for ApsaraDB for Redis instances. If you want to connect to an ApsaraDB for Redis instance over the Internet, you must apply for a public endpoint. For more information, see Apply for a public endpoint for an ApsaraDB for Redis instance.
ii. Run the <b>curl ipinfo.io |grep ip</b> command on the on-premises device to obtain its public IP address. The following figure shows a sample result.

<pre>
root@[redacted]:~# curl ipinfo.io |grep ip
  % Total   % Received % Xferd Average Speed   Time   Time   Time Current
                                 Dload Upload   Total   Spent   Left Speed
100   249 100   249   0   0   1272   0 --:--:-- --:--:-- --:--:-- 1270
  "ip": "[redacted].203",
  "readme": "https://ipinfo.io/missingauth"
</pre>
iii. Add the public IP address of the on-premises device to a whitelist of the ApsaraDB for Redis instance. For more information, see Configure whitelists.
</td></tr>
</table>

2. Log on to the device where redis-port is installed. The device may be an ECS instance or an on-premises machine.

   Note In this example, an ECS instance that runs the Ubuntu 16.04.6 operating system is used.

3. Run the following command to install the core library of OpenAPI Explorer:

```
apt-get update
apt install python-pip -y
pip install --upgrade "pip < 21.0"
pip install aliyun-python-sdk-core
```

4. Run the following command to download the script package:

```
wget 'https://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/attach/120287/cn_zh/1618924922413/redis-rdb-auto-restore.tar'
```

> **Note** The script package contains the script *rdb_restore.py*, redis-port, and other files. After you can run the script, the script automatically downloads the specified instance backup file and runs redis-port to restore the backup to the specified ApsaraDB for Redis instance.

5. Run the following command to decompress the script package:

```
tar -xvf redis-rdb-auto-restore.tar
```

6. Run the following command to switch to the directory of the decompressed script package and modify the configuration file:

```
cd redis-rdb-auto-restore
vim config.json
```

> **Note** After you run the preceding command, the system opens an editor. Enter *a* to use the editing mode.

Parameters

Parameter	Description	Example
acesskeyID	Specify the AccessKey ID and AccessKey secret of the RAM user. For more information, see Prerequisites.	LTAI5tBHRFKq************
acesskeySecret		SoA0hZtZJ7Lszgy7aL************
regionid	The ID of the region where the source ApsaraDB for Redis instance is deployed. For more information about the mappings between region IDs and region names, see Regions and zones.	cn-hangzhou

7. To save the configuration file and exit the editor, press Esc, enter *:wq*, and press Enter.
8. Run the following command to launch redis-port to migrate data:

```
python rdb_restore.py -c r-bp1q4xidqiyqpq**** -d 2021-04-19 -t r-bp1q4xidqiyqpq****.redis.rds.aliyuncs.com -p 6379 -a redistest:Pawd123456 -b 9382****2,9382****3 -t r-bp1ky7j6qc7umk****.redis.rds.aliyuncs.com -p 6379 -a Pawd123456
```

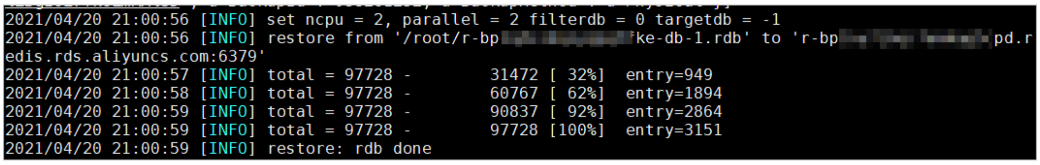
Parameters in rdb_restore.py

Parameter	Required	Description	Example
-c	Yes	The ID of the source ApsaraDB for Redis instance.	SoA0hZtZJ7Lszgy7aL************
-d	Yes	The timestamp of the backup that you want to download. The timestamp is in the *yyyy-MM-dd* format.	2021-04-19
-t	Yes	The endpoint that is used to connect to the destination ApsaraDB for Redis instance. For more information, see View endpoints. ○ If the ECS instance is connected to the ApsaraDB for Redis instance through a VPC, you must obtain the internal endpoint of the ApsaraDB for Redis instance. ○ If you connect an on-premises machine to the ApsaraDB for Redis instance over the Internet, you must obtain the public endpoint of the ApsaraDB for Redis instance.	r-bp1q4xidqiyqpq****.redis.rds.aliyuncs.com
-p	Yes	The port of the destination ApsaraDB for Redis instance. The default port is 6379.	6379
-a	Yes	The account that has the **read and write** permissions and the password of the destination ApsaraDB for Redis instance. Separate the account and password with a colon (:). For more information about how to create an account, see Create and manage database accounts.	redistest:Pawd123456
-o	No	The directory where the downloaded backup of the source instance is saved. The default directory is */root/*.	/root/

Parameter	Required	Description	Example
-b	No	The ID of the backup of the source instance. If multiple backups are created, you can use this parameter to specify the backup that you want to download. Note ◦ You can check the **Backup and Recovery** page in the console or call the DescribeBackups operation to query the backup ID. ◦ If the source instance uses a cluster architecture or read /write splitting architecture, you must download the backup of each shard by specifying this parameter. Separate multiple backup IDs with commas (,).	938261111,938262222
-h	No	Queries help information about a command.	None

After you run the program, the operational log is printed. If the data is restored, `restore: rdb done` appears.

The output that is returned by redis-port



```
2021/04/20 21:00:56 [INFO] set ncpu = 2, parallel = 2 filterdb = 0 targetdb = -1
2021/04/20 21:00:56 [INFO] restore from '/root/r-bp[redacted]ke-db-1.rdb' to 'r-bp[redacted]pd.r
edis.rds.aliyuncs.com:6379'
2021/04/20 21:00:57 [INFO] total = 97728 -        31472 [ 32%]  entry=949
2021/04/20 21:00:58 [INFO] total = 97728 -        60767 [ 62%]  entry=1894
2021/04/20 21:00:59 [INFO] total = 97728 -        90837 [ 92%]  entry=2864
2021/04/20 21:00:59 [INFO] total = 97728 -        97728 [100%]  entry=3151
2021/04/20 21:00:59 [INFO] restore: rdb done
```

11.7. 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 Create and 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 redis-shake to migrate the data of a self-managed Redis database from a backup file to an ApsaraDB for Redis instance.
> - 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

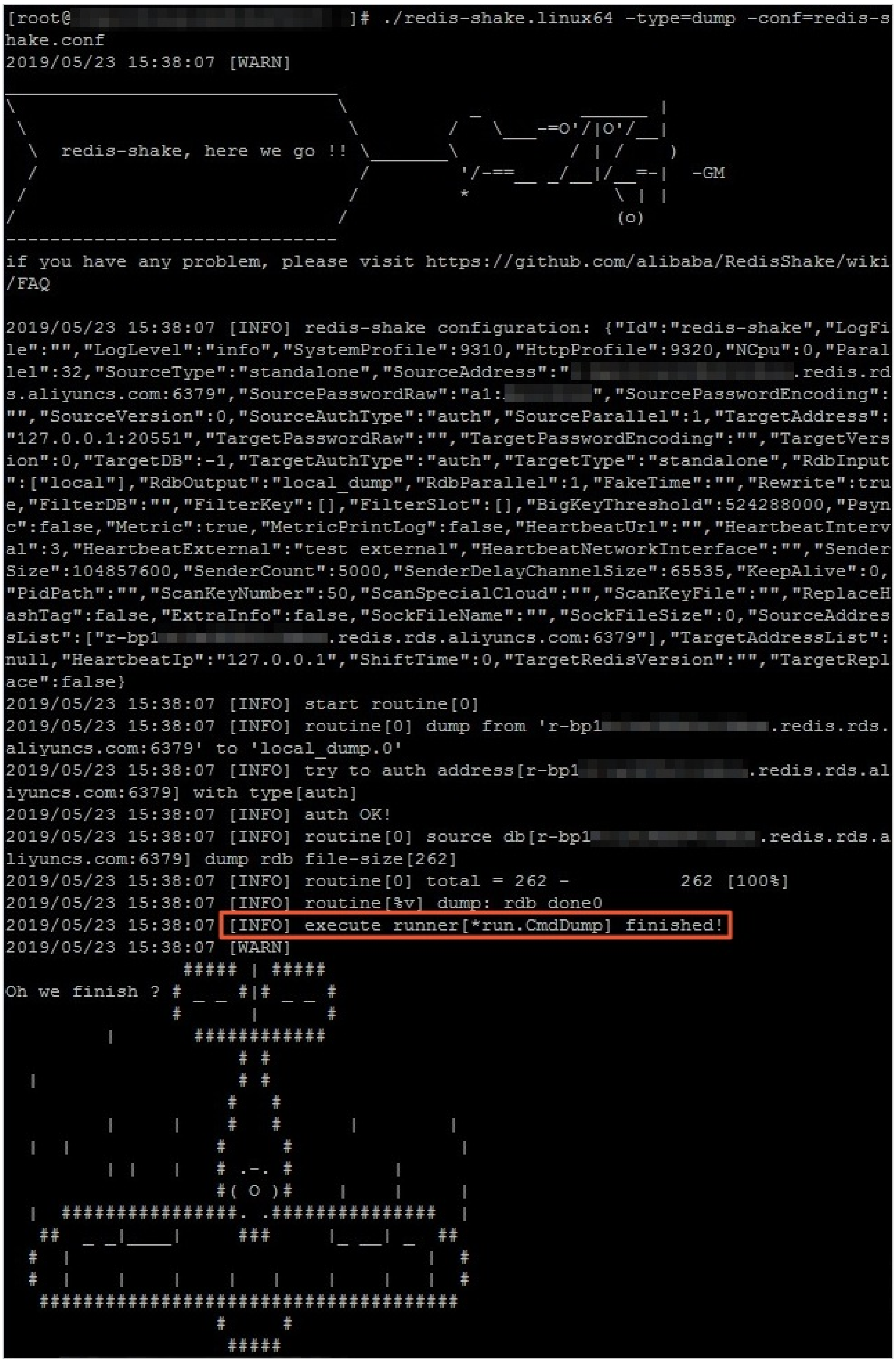
of the redis-shake tool. Parameters for the dump mode of the redis-shake tool

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@[redacted]]# ./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":"[redacted].redis.rd
s.aliyuncs.com:6379","SourcePasswordRaw":"a1:[redacted]","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[redacted].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[redacted].redis.rds.
aliyuncs.com:6379' to 'local_dump.0'
2019/05/23 15:38:07 [INFO] try to auth address[r-bp1[redacted].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[redacted].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 redis-shake to migrate the data of a self-managed Redis database from a backup file to an ApsaraDB for Redis instance.

11.8. Use redis-shake to restore data

You can use the restore mode of redis-shake to restore data from an RDB file to an ApsaraDB for Redis instance.

You can use redis-shake to restore data from an RDB file to an ApsaraDB for Redis instance. For more information, see Use redis-shake to migrate the data of a self-managed Redis database from a backup file to an ApsaraDB for Redis instance.

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

12.FAQ

12.1. Product features

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

ApsaraDB for Redis is compatible with multiple Redis versions. This provides you with a wide range of choices.

- Performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair): The instances are compatible with Redis 5.0, 4.0, 3.2, and 2.8.
- Persistent memory-optimized instances of the ApsaraDB for Redis Enhanced Edition (Tair): The instances are compatible with Redis 6.0, 5.0, 4.0, 3.2, and 2.8.
- Storage-optimized instances of the ApsaraDB for Redis Enhanced Edition (Tair): The instances are compatible with Redis 4.0, 3.2, and 2.8.
- Hybrid-storage instances (phased out) of the ApsaraDB for Redis Enhanced Edition (Tair): The instances are compatible with Redis 4.0, 3.2, and 2.8.
- ApsaraDB for Redis Community Edition instances: The instances are compatible with Redis 7.0, 6.0, 5.0, 4.0, 3.2, and 2.8.

> **Note** For more information about the Redis commands that are supported by different ApsaraDB for Redis editions and series, see Overview.

12.1.2. What is the relationship between ApsaraDB for Redis and open source Redis?

ApsaraDB for Redis is a cloud-native, high-performance, and in-memory database service that is fully compatible with open source Redis.

You can use Redis-compatible clients to perform operations such as establishing connections to and storing data in ApsaraDB for Redis instances.

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

ApsaraDB for Redis is compatible with most open source Redis commands and operations, and disables only a few commands.

For more information, see Supported Redis commands.

12.1.4. Does ApsaraDB for Redis support distributed cluster instances?

Yes, 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.

12.1.5. Can multiple replica nodes be configured for an ApsaraDB for Redis instance?

Yes, multiple replica nodes can be configured for an ApsaraDB for Redis instance. ApsaraDB for Redis read/write splitting instances support different specifications that include one, three, or five read-only nodes. For more information, see Read/write splitting instances.

ApsaraDB for Redis standard mater-replica instances and cluster mater-replica instances are configured with one master node and one replica node. For these instances, the number of read replicas cannot be changed.

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

Yes, all ApsaraDB for Redis instances have limits on CPU processing capability, data transmission bandwidth, and the number of connections.

The performance parameters vary based on different types of instances. For more information, see Specifications. You can also view the information on the buy page.

12.1.7. How many databases does a single ApsaraDB for Redis instance support?

By default, a single ApsaraDB for Redis instance can support up to 256 databases.

> ? **Note** A single ApsaraDB for Redis cluster instance can also support up to 256 databases by default. For more information about cluster instances, see Cluster master-replica instances.

12.1.8. Do all ApsaraDB for Redis instances have master and replica nodes?

All Redis instances including cluster instances configured with a master-replica architecture have master and replica nodes.

For more information about the master-replica instances, see Standard master-replica instances and Cluster master-replica instances.

12.1.9. Does ApsaraDB for Redis support read-only replica nodes?

An ApsaraDB for Redis instance runs in a master-replica architecture. However, no replica node works as a read-only node.

If you require a read-only node, use a read /write splitting instance. For more information, see Read/write splitting instances.

12.1.10. Does ApsaraDB for Redis support failovers between master and replica nodes?

Yes, ApsaraDB for Redis supports automatic failovers between master and replica nodes.

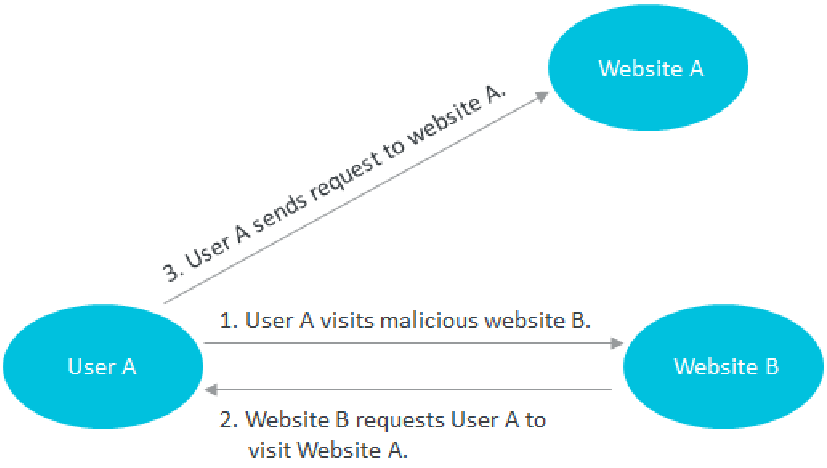
ApsaraDB for Redis automatically manages the synchronization and failover operations between the master and replica nodes.

12.1.11. Redis CSRF vulnerability analysis and security measures available for ApsaraDB for Redis

This topic describes how the Redis CSRF vulnerability occurs and the security measures available for ApsaraDB for Redis.

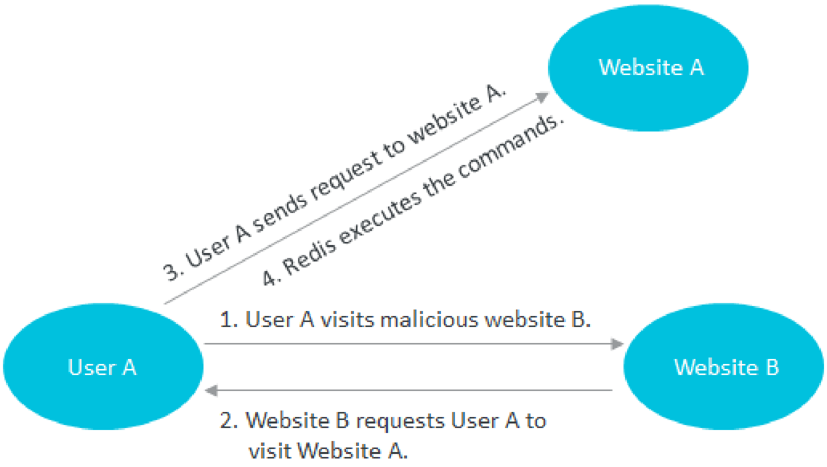
CSRF introduction

Cross-site request forgery (CSRF), also known as One Click Attack or Session Riding, is abbreviated as CSRF or XSRF and is a malicious use of websites.


Website A
3. User A sends request to website A.
1. User A visits malicious website B.
User A
Website B
2. Website B requests User A to visit Website A.

The preceding figure shows a simplified model of a CSRF attack: You visit a malicious website B. The HTTP information returned by malicious website B requires you to visit website A. However, due to a possible trust relationship between you and website A, this request is executed like what you actually sent.

Redis CSRF attack model


Website A
3. User A sends request to website A.
4. Redis executes the commands.
1. User A visits malicious website B.
User A
Website B
2. Website B requests User A to visit Website A.

Based on the preceding model of CSRF, a malicious website can require you to send an HTTP request to a Redis instance. Redis supports the text-based protocol. If Redis parses an illegal protocol, it will not disconnect from the connection. At this time, the attacker is able to send Redis commands carried by a valid HTTP request and execute them on Redis. If no password verification is required between you and the Redis instance, the Redis commands can be executed and the data is encrypted for ransom. This is similar to the MongoDB ransomware attack event.

Kernel repair

The Redis author fixed this issue in version 3.2.7. The solution is to log statement execution with keywords `POST` and `Host:` and disconnect from the connection to avoid executing subsequent Redis legal requests.

Redis security risk

In earlier versions, Redis exposed a security vulnerability. Hackers can obtain the root privileges of the Redis service under specific conditions. The main reason for this security vulnerability is that users do not know much about the security mechanism of Redis and are inexperienced in Redis operations and maintenance. At the same time, Redis does not provide robust security protection mechanisms. ApsaraDB for Redis provides more secure Redis services. We recommend that you use ApsaraDB for Redis as your Redis services in the cloud.

Security policies of ApsaraDB for Redis

Support internal access instead of access from the Internet

ApsaraDB for Redis provides trusted internal access. You cannot directly access ApsaraDB for Redis from the Internet if you do not apply for a public endpoint.

Physical network isolation

The network where ApsaraDB for Redis instances are deployed are physically isolated from the network from which you want to access ApsaraDB for Redis instances. You cannot directly access the network of backend servers.

Support VPC for network isolation

If you are an Alibaba Cloud user and deploy a virtual private cloud (VPC) to connect your services, only the services in the same VPC can access each other.

Support whitelists

ApsaraDB for Redis supports whitelist settings. You can configure a whitelist of IP addresses that can be used to access ApsaraDB for Redis instances in the ApsaraDB for Redis console.

Password authentication

ApsaraDB for Redis forcibly enables password authentication for instances deployed in a classic network. You can set a complex password to avoid password cracking.

Permission isolation

ApsaraDB for Redis isolates each instance at the backend based on permission settings and file directories. Each instance can access only its own path. This avoids mutual effects among instances.

Disable Risky commands

ApsaraDB for Redis disables some risky system management commands such as config and save. To modify parameters, you must perform two-factor authentication in the ApsaraDB for Redis console. This also avoids direct operations on the backend configuration files and management commands.

Security monitoring

ApsaraDB for Redis features comprehensive security monitoring for physical machines, regular scanning and updated security monitoring policies, and can detect security risks in early stages.

Password authentication for cluster instances of ApsaraDB for Redis

The cluster version of Redis 3.0 does not support password authentication. Cluster instances of ApsaraDB for Redis support password authentication. This improves security performance.

12.2. Database connections

12.2.1. How can I connect to an ApsaraDB for Redis instance by using redis-cli?

You can use redis-cli to connect to an ApsaraDB for Redis instance.

For more information, see Use redis-cli to connect to an ApsaraDB for Redis instance.

12.2.2. What do I do if the "WRONGPASS invalid username-password pair" error message is returned when I attempt to connect to an ApsaraDB for Redis instance?

If you attempt to use an incorrect or invalid password to connect to an ApsaraDB for Redis Community Edition instance that uses a major version of 6.0 or a major version of 5.0 and a minor version of 5.0.8 or later, the "WRONGPASS invalid username-password pair" error message appears.

You must enter the correct password in one of the following formats:

- If you use the default account whose username is the same as the instance ID, you can enter only the password.
- If you use a custom account, the format of the password must be `<user>:<password>`. For example, if the username of the custom account is `testaccount` and the password is `Rp829dlwa`, you must enter `testaccount:Rp829dlwa` as the database password.

> Note
>
> - If you use a management tool such as Redis Desktop Manager (RDM) to connect to the instance, enter a password in the format of `<user>:<password>`.
> - If you forget your password, you can reset it. For more information, see Change or reset the password.
> - If you attempt to use an incorrect password to connect to an ApsaraDB for Redis Community Edition instance that runs the 6.0 major version after password-free access has been enabled for the instance, the "WRONGPASS invalid username-password pair" error is still returned. You must enter the correct password or left the password field empty.

12.2.3. Do I need a password to connect to an ApsaraDB for Redis instance?

You must pass the password authentication when you connect to an ApsaraDB for Redis instance.

You can customize a password when you create an ApsaraDB for Redis instance. You can also set or modify the password of an instance in the ApsaraDB for Redis console after you create the instance. For more information, see Change or reset the password.

If an ApsaraDB for Redis instance with local disks that resides within a virtual private cloud (VPC) is an instance of Community Edition or a performance-enhanced instance of Enhanced Edition (Tair), or if an instance resides within a VPC is a persistent memory-optimized instance of Enhanced Edition (Tair), you can enable password-free access for the instance to allow the instance to be connected without using passwords. For more information, see Enable password-free access.

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

No. You can use a Redis client to connect to ApsaraDB for Redis instances from an ECS instance.

For more information, see Connect to an ApsaraDB for Redis instance.

12.2.5. Does ApsaraDB for Redis support Redis clients such as Jedis?

Yes. All clients that are compatible with the Redis protocol support connections to ApsaraDB for Redis. You can use any of these clients that are suitable for your applications.

For more information about how to connect to an ApsaraDB for Redis instance by using a Redis client, see Use a client to connect to an ApsaraDB for Redis instance.

12.2.6. 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 the configurations of an instance.

12.2.7. Why are clients still able to access an instance even when an instance whitelist does not include the client IP addresses?

The clients are able to access the instance because the #no_loose_check-whitelist-always parameter of the instance is set to no and the instance has password-free access over a virtual private cloud (VPC) enabled.

By default, the **#no_loose_check-whitelist-always** parameter of an instance is set to no. This way, after password-free access over a VPC is enabled for the instance, clients within the same VPC can directly connect to the instance without adding their IP addresses to an instance whitelist. For more information about password-free access, see Enable password-free access.

If you do not want to allow clients to access an instance when their IP addresses are not included in an instance whitelist, set the **#no_loose_check-whitelist-always** parameter to yes on the **Parameter Settings** page. For more information, see Modify parameters of an instance.

12.3. Using the database

12.3.1. Does ApsaraDB for Redis support Bloom filters?

Yes, ApsaraDB for Redis supports Bloom filters. A Bloom filter is used to determine whether an element exists in a data set. Bloom filters are suitable for scenarios such as cache penetration prevention and the use of web interceptors.

Bloom filters are supported by performance-enhanced instances of the ApsaraDB for Redis Enhanced Edition (Tair). For more information about performance-enhanced instances, see Performance-enhanced instances. For more information about Bloom filters supported by performance-enhanced instances, see TairBloom commands. Performance-enhanced instances also integrate multiple Redis modules developed by Alibaba Cloud. These modules include TairString (including CAS and CAD commands), TairHash, TairGIS, TairBloom, TairDoc, TairTS, TairCpc, TairZset, TairRoaring, and TairSearch. These modules facilitate development in sophisticated scenarios and allow you to focus on business innovation.

Warning Bloom filter-related API operations such as **BF.ADD** supported by existing instances of ApsaraDB for Redis 4.0 are only for internal use. These operations are no longer supported by instances of major versions later than 4.0 and newly purchased instances of ApsaraDB for Redis 4.0. As such, if you call the operations on these instances, unknown errors such as failed cache analytics may take place. We recommend that you change your instance into a performance-enhanced instance of the ApsaraDB for Redis Enhanced Edition (Tair) that supports optimized Bloom filters. For more information about performance-enhanced instances, see Performance-enhanced instances.

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

This topic describes the database features of ApsaraDB for Redis, such as the supported number of databases, memory usage limit, and switching methods.

Each ApsaraDB for Redis instance has 256 databases named from DB0 to DB255. 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.

You can run the **SELECT** command to switch among different databases. For example, to switch to DB10, you can run the following command:

```
SELECT 10
```

12.3.3. How can I import sample data to ApsaraDB for Redis?

You can use multiple methods to import data from on-premises Redis instances or Redis instances deployed in other cloud services to ApsaraDB for Redis instances. For more information, see Overview.

12.3.4. 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-bp1xxxxxxxxxxxxx (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.

12.3.5. Can I restore the deleted data of an ApsaraDB for Redis instance?

The ApsaraDB for Redis instance is deleted due to misoperation, and you have no backup files on your on-premises machine. In this case, you can attempt to restore data in the ApsaraDB for Redis console by using the automatically saved backup files. You cannot restore data without backup files.

By default, ApsaraDB for Redis automatically backs up data of an ApsaraDB for Redis instance once a day and retains backup files for seven days. You can restore data from the backup files to an ApsaraDB for Redis instance in the console. For more information, see Backup and restoration solutions.

Warning Use caution with commands, features, and API operations that can clear data.

12.3.6. How can I monitor an ApsaraDB for Redis instance? Does the system automatically generate alerts when the capacity is reached?

ApsaraDB for Redis provides multiple sets of monitoring metrics. You can set alert rules based on your business requirements.

For more information, see Metrics.

You can set alert rules for the specific monitoring metrics based on your business requirements. For example, you can set an alert when the memory utilization of an instance reaches a specified threshold. For more information, see Alert settings.

12.3.7. Why is the memory usage of a new ApsaraDB for Redis instance not equal to 0?

ApsaraDB for Redis shows the same service behavior as Redis. After you create an instance, the instance generates database metadata that occupies a small amount of storage on the instance. You can view the occupied space in the ApsaraDB for Redis console.

Memory occupied by metadata information:

- The metadata size for standard instances is 30 MB to 50 MB.
- The metadata size on each shard of cluster instances is 30 MB to 50 MB. The total size of metadata for a cluster equals the sum of the metadata size on each shard of the cluster.

12.3.8. Can data be automatically and evenly distributed to data shards after I upgrade an ApsaraDB for Redis standard instance to an ApsaraDB for Redis cluster instance or change the number of data shards of an ApsaraDB for Redis cluster instance?

If you upgrade an ApsaraDB for Redis standard instance to an ApsaraDB for Redis cluster instance or change the number of data shards of an ApsaraDB for Redis cluster instance, proxy nodes evenly distribute data to data shards. You do not need to perform additional operations.

> **Note** Proxy nodes can enable architecture changes, manage the traffic to read replicas, cache hotkey data, and support multiple databases for cluster instances. For more information, see Features of proxy nodes.

12.3.9. Why am I unable to switch between the proxy mode and the direct connection mode when I use cloud disk-based cluster instances?

Connection method and code vary with instance mode. You can connect to cloud disk-based cluster instances in proxy mode in the same manner as you connect to standard instances and connect to cloud disk-based cluster instances in direct connection mode in the same manner as you connect to native Redis clusters. If you use a cloud disk-based cluster instance, you cannot enable both the proxy and direct connection modes for the instance at the same time or switch between the two modes.

To switch between the two modes, you can use the instance restoration feature on the **Backup and Recovery** page to restore backup data in an existing cloud disk-based cluster instance to a new instance. Then, on the **Clone Instance** page that appears, set **Connection Mode** to **Proxy** or **Direct Connection**. For more information, see Restore data from a backup set to a new instance.

Warning After the new cloud disk-based cluster instance is created with a new mode, you must modify the connection code accordingly to connect to the instance. Proceed with caution.

12.3.10. Do I need to modify the code after I upgrade an ApsaraDB for Redis standard instance to an ApsaraDB for Redis cluster instance?

If you upgrade an ApsaraDB for Redis standard instance to an ApsaraDB for Redis cluster instance that uses local disks, you do not need to modify the code. This is because proxy nodes enable architecture changes and allow you to use a cluster instance in the same way as you use a standard instance.

If your business requirements outgrow the capabilities of a standard instance, you can migrate the data of the standard instance to a cluster instance that has proxy nodes without having to modify the code. This significantly reduces your costs.

Note

- Proxy nodes can balance loads, route commands, manage the traffic to read replicas, cache hotkey data, and support multiple databases for cluster instances. For more information, see Features of proxy nodes.
- If you upgrade an ApsaraDB for Redis standard instance to an ApsaraDB for Redis cluster instance that uses cloud disks, you must modify the code based on the SDK that you use.

12.4. Data persistence

12.4.1. Does ApsaraDB for Redis support data persistence?

Yes, ApsaraDB for Redis supports data persistence. This topic describes the data persistence mechanism of ApsaraDB for Redis.

ApsaraDB for Redis Community Edition provides a hybrid of memory and hard disks for storage and backs up data in RDB and append-only file (AOF) files to meet data persistence requirements. You can back up and restore data in the ApsaraDB for Redis console.

12.5. Set parameters

12.5.1. Does ApsaraDB for Redis support changing configuration parameters?

To ensure instance security and stability, ApsaraDB for Redis allows you to modify only some parameter configurations.

For more information, see Modify parameters of an instance.

12.5.2. 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 Modify parameters of an instance.

12.5.3. How do I configure semi-synchronous replication for persistent memory-optimized instances of the ApsaraDB for Redis Enhanced Edition (Tair)?

Semi-synchronous replication cannot be manually enabled for persistent memory-optimized instances. By default, semi-synchronous replication is disabled. To enable semi-synchronous replication, submit a ticket.

If semi-synchronous replication is enabled, logs are transmitted from a master node to a replica node after the data update that the client initiates is complete on the master node. After the replica node receives all logs, the master node returns the log transmission information to the client. This minimizes data loss while maintaining high availability.

> ⓘ **Note** If a replica node is unavailable or the communication between a master node and a replica node is abnormal, semi-synchronous replication degrades into asynchronous replication.

12.6. Expiration policy

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

This topic describes the eviction policies of ApsaraDB for Redis.

By default, an ApsaraDB for Redis instance evicts data by using the volatile-lru policy. To modify the eviction policy for an instance, log on to the ApsaraDB for Redis console, click the instance ID on the Instances page to go to the Instance Information page, and then click **System Parameters** in the left-side navigation pane. For more information, see Modify parameters of an instance.

- volatile-lru

  The system evicts only data that has time to live (TTL) configured based on the least recently used (LRU) algorithm.

- volatile-ttl

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

- allkeys-lru

  The system evicts data based on the LRU algorithm.

- volatile-random

  The system only randomly evicts data that has TTL configured.

- allkeys-random

  The system randomly evicts data.

- noeviction

  No data is evicted, and an error message is returned if new data is written when the memory is full (except DEL and some other commands).

- volatile-lfu

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

- allkeys-lfu

  The system evicts the least frequently used keys based on the LFU algorithm.