GeminiDB Redis

Issue	01
Date	2024-09-30

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

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

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

Contents

1 Service Overview

1.1 Highlights

Cloud-native GeminiDB is a key-value (KV) database service featuring high stability, cost-effectiveness, elasticity, and easy O&M. It is fully compatible with the Redis protocol, supports advanced functions such as PITR recoveries for game rollback and FastLoad for feature data import, and it allows you to set the field expiration time for hash keys and blacklist for high-risk keys.

GeminiDB is widely used in scenarios such as game friends list and player rankings, ad placement, personalized recommendations, e-commerce inventory, IoV data storage, and ERP systems. For details, see **Application Scenarios**.

GeminiDB has the following advantages over open-source on-premises KV databases (such as Redis and Pika databases):

Table 1-1 Comparison between GeminiDB and open-source on-premises KV databases

Dimension	Item	Open-Source On-Premises KV Database	GeminiDB
Stability	Performance jitter caused by forks	**Service stability is severely affected by fork issues.** When RDB backups are generated, the Append Only File (AOF) is rewritten, or full data is synchronized, a fork is called. This increases latency and causes out of memory (OOM) issues.	**Service stability is improved as fork issues are addressed.** There is no performance jitter during backup and synchronization.

Dimension	Item	Open-Source On-Premises KV Database	GeminiDB
	Long latency in big key scenarios	**The single-thread architecture slows down subsequent requests.** In a single-thread architecture, big key requests slow down all subsequent requests and may trigger flow control or OOM issues on shards.	**The multi-thread architecture reduces the impact on subsequent keys.** GeminiDB uses a multi-thread architecture, which improves concurrency and reduces the impact of big keys on subsequent read and write operations of other keys.
	Bandwidth limiting during peak hours	**Flow control is easily triggered, affecting services.** Open-source on-premises databases typically use a hybrid deployment that strictly limits the bandwidth. Flow control is easily triggered for smaller instances.	**Up to 10 Gbit/s is supported, allowing GeminiDB to handle service surges.** By using an independent container deployment, GeminiDB can enable a load balancer to support a bandwidth of 10 Gbit/s.
	Impact of scale-out on services	**Scale-out can take several minutes or sometimes even hours, greatly affecting services.** Adding nodes involves data migration. Services may be affected for a few minutes or up to several hours.	**Smooth scale-out is supported and has minimal impact on services.** Scale-out can be completed in seconds and without interrupting services. Adding nodes does not requires any data migration. There is just a few seconds of jitter.
	HA scenarios such as node breakdowns and primary/secondary switchovers	**Long switchover time: RTO > 30s**	**Second-level jitters, RTO < 10s**

Dimension	Item	Open-Source On-Premises KV Database	GeminiDB
Performance	QPS	**QPS per shard: 80,000 to 100,000** In a single-thread architecture, the QPS of a single shard does not increase after CPUs are added.	**QPS per shard: 10,000 to 300,000** In a multi-thread architecture, the QPS can increase linearly as CPUs are added.
	Latency	**Low latency**	**Low latency** In most service scenarios, the average latency is 1 ms, and the p99 latency is about 2 ms.
O&M capabilities	Audit logs of risky operations	Not supported	**High-risk commands can be traced.**
	Circuit breakers triggered by abnormal requests to keys	Not supported	**Key blacklists and one-click circuit breakers for high-risk operations are supported, so the entire instance is not affected.**
	Slow query logs	Supported	**Supported. More details can be found in the logs.**
	Big key diagnosis	Not supported	**Online diagnosis of big keys by category is supported.**
	Hot key diagnosis	Supported	Online diagnosis of hot keys is supported.

Dimension	Item	Open-Source On-Premises KV Database	GeminiDB
Cost	Utilization cost	**The in-memory storage is expensive.**	**The cost of databases with the same specifications is 30% lower than open-source on-premises databases.** Users can purchase additional compute resources and storage resources independently to eliminate the resource waste associated with coupled storage and compute.
	Data compression	Not supported	**The compression ratio (4:1) enables databases with the same specifications to store more data.**
	Scale-out	**Coupled storage and compute increases costs exponentially.**	**Decoupled storage and compute supports independent scaling of compute and storage resources.**
Availability	/	**If any pair of primary and standby nodes is faulty, the entire cluster becomes unavailable.**	**GeminiDB provides superlative fault tolerance (N-1 reliability).**
Data reliability	/	**Weak** Thousands or tens of thousands of records will be lost if nodes are restarted and the network fluctuates. Weak data consistency may cause dirty reads.	**High reliability** GeminiDB provides three-copy storage, so it can serve as the primary database to replace the traditional DB+Cache solution, and it also ensures strong data consistency and avoids dirty reads.

Dimension	Item	Open-Source On-Premises KV Database	GeminiDB
Advanced features	Autoscaling	Not supported	Supported
	Setting the expiration time for fields in hashes	Not supported	**Supported. Service design is less complex and concurrency is increased.**
	Fast data loading	Not supported	**FastLoad allows feature data to be imported faster, reducing the impact on online services.**
	Point-In-Time Recovery (PITR)	Not supported	Supported PITR rollbacks and quick data restoration to the original instance are supported, making GeminiDB a great fit for gaming applications.
	DR instances	Not supported	**Intra-region and cross-region DR instances can be created.**

1.2 Application Scenarios

As a key-value database compatible with Redis APIs, GeminiDB Redis API extends application scenarios of Redis so that it can better meet diversified service requirements such as persistent and hybrid storage.

E-Commerce

- For e-commerce applications, some commodity data is more frequently queried than others. GeminiDB Redis API stores frequently queried commodity information in memory as hot data, and cold data in the shared storage pool. This not only meets the quick access requirements of popular commodities, but also avoid excessive in-memory storage costs
- GeminiDB Redis API can permanently store massive amounts of historical order data of e-commerce applications. It allows you to access data through the Redis API and provides TB-level storage.
- There may be a large number of concurrent access requests within a short period of time during an e-commerce promotion. GeminiDB Redis API works

as a front-end cache (large memory required) to help back-end databases handle service peaks. You can easily add compute nodes in seconds to handle the expected peak traffic.

Gaming

- The schema of gaming services is simple. You can select GeminiDB Redis API as a persistent database and use simple Redis APIs to quickly develop and launch services. For example, the sorted set structure of Redis can be used to display game rankings in real time.
- In delay-sensitive gaming scenarios, GeminiDB Redis API can be used as the front-end cache (large memory required) to accelerate access to applications.

Live Streaming

The most-viewed live streaming content usually dominates most traffic of the live streaming applications. GeminiDB Redis API can efficiently use memory resources by retaining popular live streaming data in the memory and other data in the shared storage, reducing your business costs.

Online Education

Online education applications store a large amount of data such as courses and Qs&As. However, only hot data (including most-viewed courses, latest question libraries, and lectures by famous teachers) is frequently accessed. GeminiDB Redis API can save data in memory or shared storage based on data access frequency, achieving a balance between performance and costs.

Persistent Storage for Other Applications

With the rapid development of the Internet, various large-scale applications have increasing requirements for persistent storage. Specifically, a massive amount of data needs to be stored, including historical orders, feature engineering, log records, location coordinates, machine learning, and user profiles. A common feature of these scenarios is large data volume and long validity period. Therefore, a large-capacity and low-cost key-value storage service is required to collect and transfer data. Redis is the most widely used key-value service. Its various data structures and operation APIs have innate advantages in storing such data. However, the native Redis can only be used as a cache and cannot guarantee persistence.

In addition to compatibility with Redis APIs, GeminiDB Redis API provides large-capacity, low-cost, and high-reliability data storage capabilities, making it well-suited to persistent storage scenarios.

1.3 Compatible APIs and Versions

This section describes the compatible APIs and versions supported by GeminiDB RedisAPI.

Table 1-2 Compatible APIs and versions

Compatible API	Instance Type	Version
Redis	● Cluster With a sharded cluster architecture, this type of instance supports connections through proxies and is compatible with Redis clusters. High horizontal scalability, millions of QPS, and tens of TB of data are supported.	6.2 (including 6.2.*X*), 5.0, and earlier versions.

1.4 Instance Specifications

This section describes available GeminiDB Redis instance specifications. The instance specifications depend on the selected CPU model.

GeminiDB Redis API allows for cold and hot data exchange and has the ability to handle a capacity that surpasses the limitations of memory. Hot data is stored in the memory, and full data is stored in the high-performance storage pool. The total instance space refers to the total storage capacity, which determines the upper limit of data storage. For details about the node memory, see table 4.

Table 1-3 GeminiDB Redis cluster instance specifications (fast configuration)

Instance Type	Storage (GB)	Nodes	Node Flavor	vCPUs	QPS	Max. Connections	Databases	Accounts
Cluster	4	2	geminidb.redis.medium.2	1	20,000	20,000	256	200
	8	2	geminidb.redis.medium.4	1	20,000	20,000	256	200
	16	2	geminidb.redis.large.4	2	40,000	20,000	256	200
	24	3	geminidb.redis.large.4	2	60,000	30,000	256	200

Instance Type	Storage (GB)	Nodes	Node Flavor	vCPUs	QPS	Max. Connections	Databases	Accounts
	32	4	geminidb.redis.large.4	2	80,000	40,000	256	200
	48	3	geminidb.redis.xlarge.4	4	120,000	30,000	1,000	200
	64	4	geminidb.redis.xlarge.4	4	160,000	40,000	1,000	200
	96	3	geminidb.redis.2xlarge.4	8	240,000	30,000	1,000	200
	128	4	geminidb.redis.2xlarge.4	8	320,000	40,000	1,000	200
	192	6	geminidb.redis.2xlarge.4	8	480,000	60,000	1,000	200
	256	8	geminidb.redis.2xlarge.4	8	640,000	80,000	1,000	200
	384	10	geminidb.redis.2xlarge.4	8	800,000	100,000	1,000	200
	512	6	geminidb.redis.4xlarge.4	16	960,000	60,000	1,000	200
	768	9	geminidb.redis.4xlarge.4	16	1,440,000	90,000	1,000	200
	1024	12	geminidb.redis.4xlarge.4	16	1,920,000	120,000	1,000	200
	2048	22	geminidb.redis.4xlarge.4	16	3,520,000	220,000	1,000	200
	4096	24	geminidb.redis.8xlarge.4	32	7,680,000	240,000	1,000	200

Instance Type	Storage (GB)	Nodes	Node Flavor	vCPUs	QPS	Max. Connections	Databases	Accounts
	8192	36	geminidb.redis.8xlarge.4	32	11,520,000	360,000	1,000	200

Table 1-4 GeminiDB Redis instance specifications

Flavor	vCPUs	Min. Persistent Storage Space (GB) per Single-node Instance	Max. Persistent Storage Space (GB) per Single-node Instance	Maximum Connections per Single-node Instance
geminidb.redis.medium.4	1	4	32	10,000
geminidb.redis.large.4	2	8	64	10,000
geminidb.redis.xlarge.4	4	16	128	10,000
geminidb.redis.2xlarge.4	8	32	256	10,000
geminidb.redis.4xlarge.4	16	64	512	10,000
geminidb.redis.8xlarge.4	32	128	1024	10,000
geminidb.redis.medium.8	1	8	64	10,000
geminidb.redis.large.8	2	16	128	10,000
geminidb.redis.xlarge.8	4	32	256	10,000
geminidb.redis.2xlarge.8	8	64	512	10,000
geminidb.redis.4xlarge.8	16	128	1024	10,000
geminidb.redis.8xlarge.8	32	256	2048	10,000

2 Getting Started with GeminiDB Redis API

2.1 Getting to Know GeminiDB Redis API

This section describes GeminiDB Redis instance type, helping you quickly create and connect to a GeminiDB Redis instance.

Table 2-1 Instance types

Instance Type	Scenario	Reference
Cluster	With a sharded cluster architecture, this type of instance supports connections through proxies and is compatible with Redis clusters. It offers strong horizontal expansion capability, supporting millions of QPS and dozens of TB-level services.	**Buying and Connecting to a Cluster Instance**

Connection Methods

Data Admin Service (DAS) enables you to manage instances on a web-based console, simplifying database management and improving working efficiency. You can connect and manage instances through DAS. By default, you have the permission of remote login. DAS is secure and convenient for connecting to GeminiDB Redis instances.

Table 2-2 Connection on DAS

Method	Scenario	Remarks
DAS	You can log in to an instance on the console without using an IP address.	● Easy to use, secure, advanced, and intelligent ● By default, you have the permission of remote login. DAS is secure and convenient for connecting to DB instances.

More Connection Operations

- See **Connection Modes**.

2.2 Buying and Connecting to a Cluster Instance

This section describes how to buy a GeminiDB Redis cluster instance on the GeminiDB console and connect to the instance.

With a sharded cluster architecture, this type of instance supports connections through proxies and is compatible with Redis clusters. It offers strong horizontal expansion capability, supporting millions of QPS and dozens of TB-level services.

Each tenant has up to 50 GeminiDB Redis instances by default. To request a higher quota, contact customer service.

- **Step 1: Buy an instance**.
- **Step 2: Connect to the instance through DAS**.

  For details about other connection methods, see **Connecting to an Instance**.

Step 1: Buying an Instance

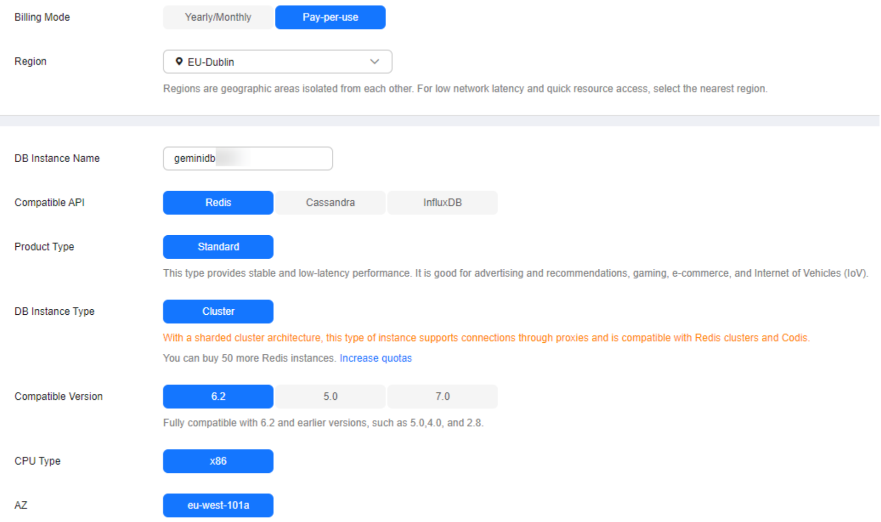
1. Log in to the GeminiDB console.
2. In the service list, choose **Databases** > **GeminiDB**.
3. On the **Instances** page, click **Buy DB Instance**.
4. On the displayed page, select a billing mode, configure instance specifications, and click **Next**.

The following parameters are for reference only. Select proper specifications as needed. **Table 3-1** lists details about the parameters.

- **Billing Mode**: Select **Pay-per-use**.
- **Region**: Select EU-Dublin.
- **DB Instance Name**: Enter a custom name.
- **Compatible API**: Select **Redis**.
- **DB Instance Type**: Select **Cluster**.
- **Compatible Version**: Select **6.2**.
- **CPU Type**: Select **x86** (default).

- **AZ**: Select **cn-north-4a,cn-north-4b,cn-north-4c**.
- **Instance Creation Mode**: Select **Fast configure**.
- **Instance Specifications**: Select **8 GB** and **Standard 1 vCPU**.
- Network information: Configure **VPC**, **subnet**, and **Database Port** (default)
- **Database Password**: Enter a value based on the password policy.
- **Enterprise Project**: Select **default**.
- Retain the default values for other parameters.

Figure 2-1 Billing mode and basic information



Billing Mode: Yearly/Monthly | Pay-per-use

Region: EU-Dublin

Regions are geographic areas isolated from each other. For low network latency and quick resource access, select the nearest region.

DB Instance Name: geminidb

Compatible API: Redis | Cassandra | InfluxDB

Product Type: Standard

This type provides stable and low-latency performance. It is good for advertising and recommendations, gaming, e-commerce, and Internet of Vehicles (IoV).

DB Instance Type: Cluster

With a sharded cluster architecture, this type of instance supports connections through proxies and is compatible with Redis clusters and Codis.

You can buy 50 more Redis instances. Increase quotas

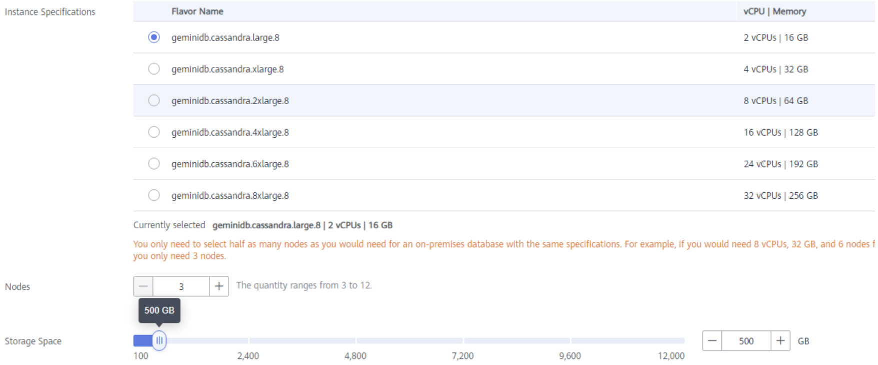
Compatible Version: 6.2 | 5.0 | 7.0

Fully compatible with 6.2 and earlier versions, such as 5.0,4.0, and 2.8.

CPU Type: x86

AZ: eu-west-101a

Figure 2-2 Specifications and storage



Instance Specifications

	Flavor Name	vCPU | Memory
●	geminidb.cassandra.large.8	2 vCPUs | 16 GB
○	geminidb.cassandra.xlarge.8	4 vCPUs | 32 GB
○	geminidb.cassandra.2xlarge.8	8 vCPUs | 64 GB
○	geminidb.cassandra.4xlarge.8	16 vCPUs | 128 GB
○	geminidb.cassandra.6xlarge.8	24 vCPUs | 192 GB
○	geminidb.cassandra.8xlarge.8	32 vCPUs | 256 GB

Currently selected **geminidb.cassandra.large.8 | 2 vCPUs | 16 GB**

You only need to select half as many nodes as you would need for an on-premises database with the same specifications. For example, if you would need 8 vCPUs, 32 GB, and 6 nodes f you only need 3 nodes.

Nodes: 3 The quantity ranges from 3 to 12.

Storage Space: 500 GB

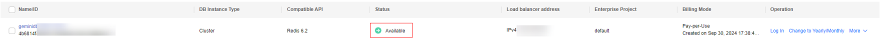
5. On the order confirmation page, check the instance information. If you need to modify the information, click **Previous**. If no modification is required, read and agree to the service agreement and click **Submit**.

6. Click **Back to Instance Management** to go to the instance list.

7. On the **Instances** page, view and manage the created instance.

- Creating an instance takes about 5 to 9 minutes. During the process, the instance status becomes **Creating**.
- After the creation is complete, the status changes to **Available**.

Figure 2-3 Successful purchase



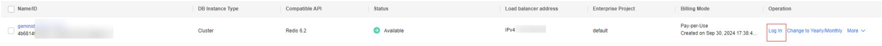
Name/ID	DB Instance Type	Compatible API	Status	Load balancer address	Enterprise Project	Billing Mode	Operation
geminidt 4b6814f	Cluster	Redis 6.2	Available	IPv4	default	Pay-per-Use Created on Sep 30, 2024 17:38:4...	Log In Change to Yearly/Monthly More

Step 2: Connecting to an Instance Through DAS

DAS enables you to manage DB instances from a web-based console, simplifying database management and improving efficiency. You can connect and manage instances through DAS. By default, you have permission of remote login. DAS is secure and convenient for connecting to DB instances.

1. Log in to the GeminiDB console.
2. In the service list, choose **Databases** > **GeminiDB**.
3. In the instance list, locate the target instance and click **Log In** in the **Operation** column.

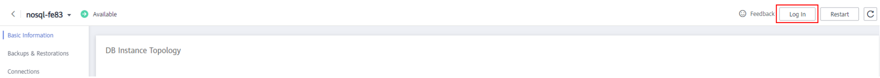
Figure 2-4 Connecting to a GeminiDB Redis instance



Name/ID	DB Instance Type	Compatible API	Status	Load balancer address	Enterprise Project	Billing Mode	Operation
geminidt 4b6814f	Cluster	Redis 6.2	Available	IPv4	default	Pay-per-Use Created on Sep 30, 2024 17:38:4...	Log In Change to Yearly/Monthly More

Alternatively, click the instance name to go to the **Basic Information** page. Click **Log In** in the upper right corner of the page.

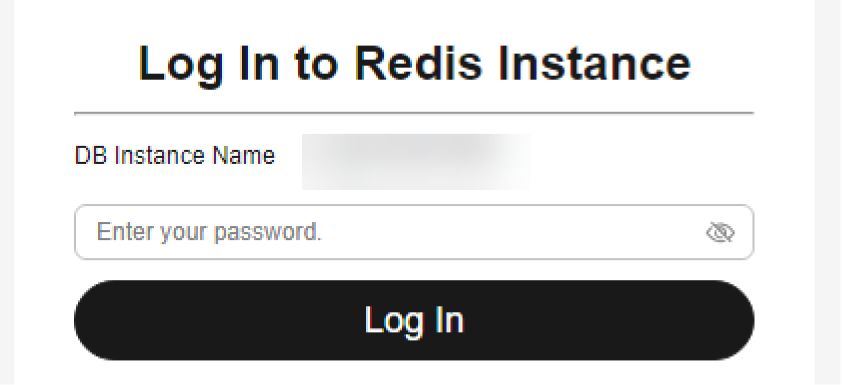
Figure 2-5 Connecting to a GeminiDB Redis instance



4. Enter the password for logging in to the instance.

You need to enter the password only when you log in to a GeminiDB Redis instance first time or after you reset the password.

Figure 2-6 Logging in to the GeminiDB Redis instance



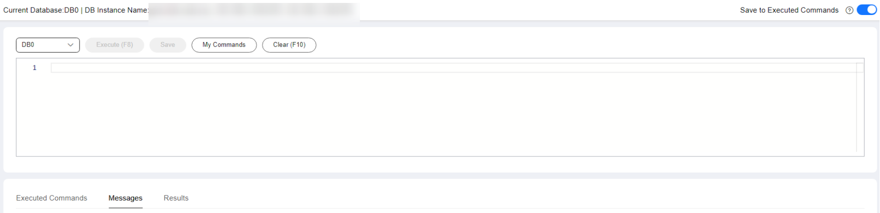
Log In to Redis Instance

DB Instance Name

Enter your password.

5. Manage relevant databases.

Figure 2-7 Instance homepage



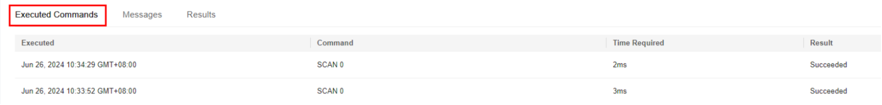
Current Database:DB0 | DB Instance Name:

- Save commands to the execution record.

  This function is enabled by default to save the recently executed commands for your later query.

  Then you can click the **Executed Commands** tab on the lower page to view historical commands.

Figure 2-8 Viewing executed commands



Executed	Command	Time Required	Result
Jun 26, 2024 10:34:29 GMT+08:00	SCAN 0	2ms	Succeeded
Jun 26, 2024 10:33:52 GMT+08:00	SCAN 0	3ms	Succeeded

  If this function is disabled, the commands executed subsequently are not displayed. You can click ● next to **Save Executed SQL Statements** in the upper right corner to disable this function.

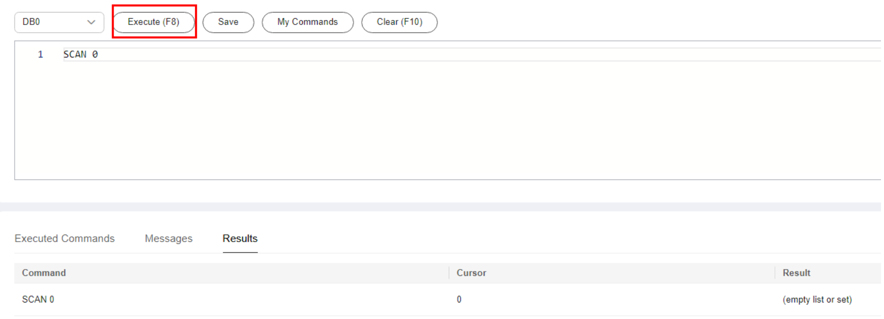
- Execute a command.

  Enter a command in the command window and click **Execute** or **F8**.

  NOTE

  - Do not use transactions, Lua scripts, Pub/Sub commands, or other commands that have blocking semantics.
  - For an instance that supports multiple databases, you can change the current database on the console but cannot change it using a SELECT statement.

Figure 2-9 Executing a command



```
SCAN 0
```

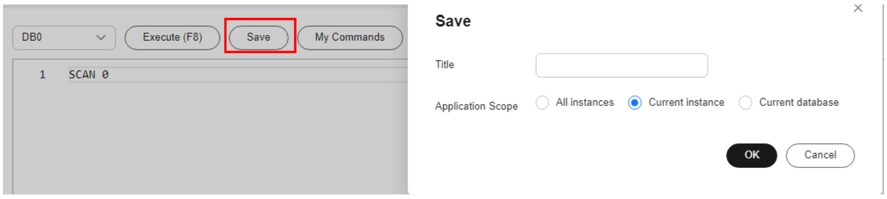
Command	Cursor	Result
SCAN 0	0	(empty list or set)

  After a command is executed, you can view the execution result on the **Results** page.

- Save a command.

  You can save a command to all instances, the current instance, or the current database. Then you can view details in **My Commands**.

Figure 2-10 Saving a command



```
SCAN 0
```

Save

Title

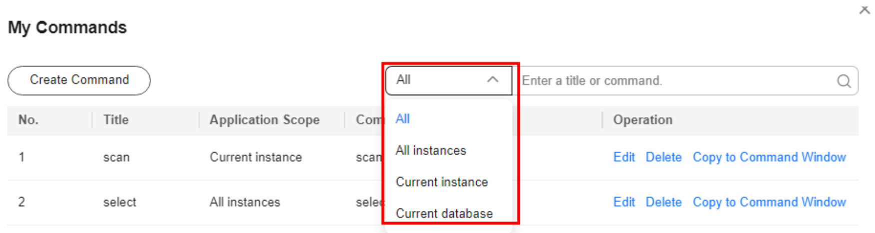
Application Scope ○ All instances ● Current instance ○ Current database

– View my commands.

Common commands are displayed the **My Commands** page.

You can set a filter to narrow the scope of commands. If you select **All**, all commands saved in the current account are displayed.

Figure 2-11 Filtering commands

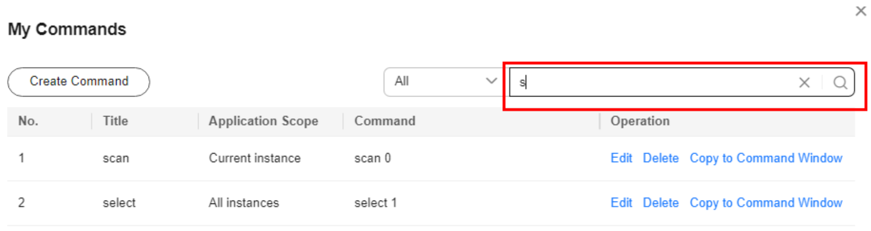


My Commands

No.	Title	Application Scope	Com	Operation
1	scan	Current instance	scan	Edit Delete Copy to Command Window
2	select	All instances	selec	Edit Delete Copy to Command Window

Alternatively, you can enter a command title or statement in the search box to search for the corresponding command.

Figure 2-12 Searching for a command

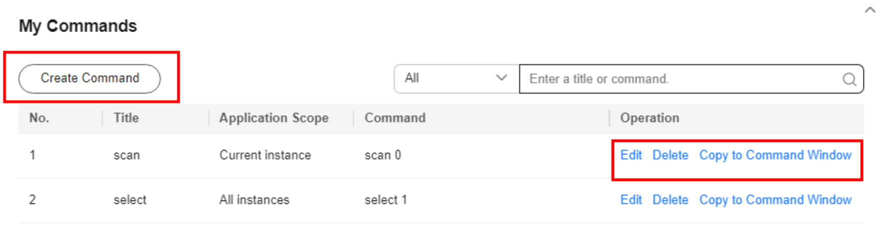


My Commands

No.	Title	Application Scope	Command	Operation
1	scan	Current instance	scan 0	Edit Delete Copy to Command Window
2	select	All instances	select 1	Edit Delete Copy to Command Window

On the **My Commands** page, you can also create, edit, and delete a command or copy it to the **Execute** window.

Figure 2-13 Managing a command



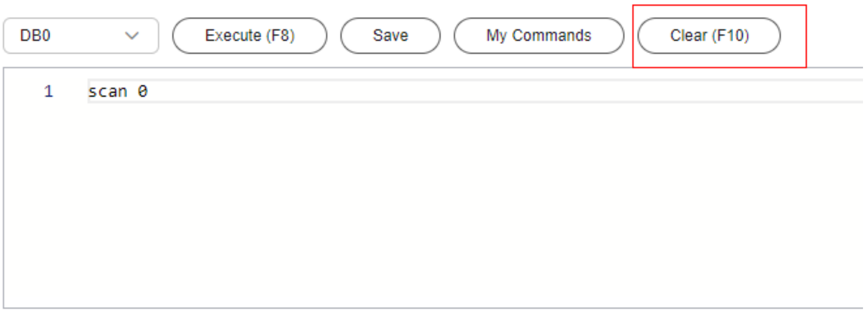
My Commands

No.	Title	Application Scope	Command	Operation
1	scan	Current instance	scan 0	Edit Delete Copy to Command Window
2	select	All instances	select 1	Edit Delete Copy to Command Window

– Clear a command.

You can also press **F10** to clear the command in the execution window.

Figure 2-14 Clearing a command



```
scan 0
```

3 Working with GeminiDB Redis API

3.1 IAM Permissions Management

3.1.1 Creating a User and Assigning Permissions

This section describes how to use **IAM** to control fine-grained permissions for your GeminiDB resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing GeminiDB resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your GeminiDB resources.

If your Huawei Cloud account does not require individual IAM users, skip this section.

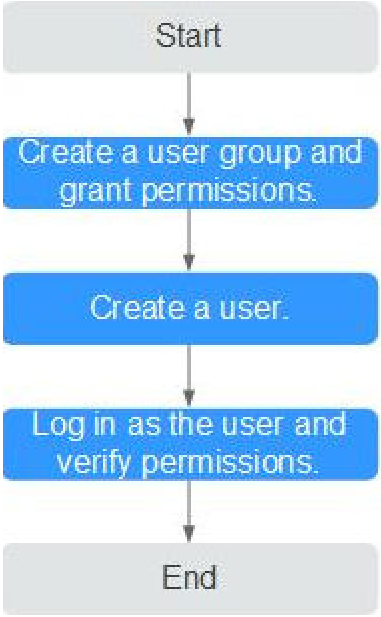
The following describes the procedure for granting permissions (see **Figure 3-1**).

Prerequisites

Learn about the permissions supported by GeminiDB and choose policies or roles based on your requirements. For details about the permissions, see . For system policies of other services, see **Permissions Policies**.

Process Flow

Figure 3-1 Process of granting GeminiDB permissions


Start
Create a user group and grant permissions.
Create a user.
Log in as the user and verify permissions.
End

1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console and attach the **GeminiDB FullAccess** policy to the group.

   > NOTE
   >
   > To use some interconnected services, you also need to configure permissions of such services.
   >
   > For example, when using DAS to connect to a DB instance, you need to configure the **GaussDB FullAccess** and **DAS FullAccess** permissions.

2. **Create an IAM user** and add it to a user group.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the management console using the created user, and verify the user's permissions:

   Choose **Service List** > **GeminiDB** and click **Buy DB Instance**. If you can buy an instance, the required permission policy has taken effect.

3.1.2 Creating a Custom Policy

Custom policies can be created to supplement the system-defined policies of GeminiDB. For the actions supported for custom policies, see **GeminiDB Actions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following describes examples of common GeminiDB custom policies.

Example Custom Policy

- Example 1: Allowing users to create GeminiDB instances

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "nosql:instance:create"
              ]
          }
      ]
  }
  ```

- Example 2: Refusing users to delete GeminiDB instances

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the policies assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

  The following method can be used if you need to assign permissions of the **GeminiDB FullAccess** policy to a user but you want to prevent the user from deleting GeminiDB instances. Create a custom policy for denying instance deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on GeminiDB instances except deleting GeminiDB instances. The following is an example of the deny policy:

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
            "Effect": "Deny"
            "Action": [
                  "nosql:instance:delete"
              ],
            }
      ]
  }
  ```

- Example 3: Defining permissions for multiple services in a policy

  A custom policy can contain the actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

  ```
  {
          "Version": "1.1",
          "Statement": [
                  {
                          "Action": [
                                  "nosql:instance:create",
                                  "nosql:instance:rename",
                                  "nosql:instance:delete",
                                  "vpc:publicIps:list",
                                  "vpc:publicIps:update"
                          ],
                          "Effect": "Allow"
                  }
          ]
  }
  ```

3.2 Billing Management

3.2.1 Renewing Instances

This section describes how to renew your yearly/monthly GeminiDB Redis instances.

Precautions

- Pay-per-use instances cannot be renewed.

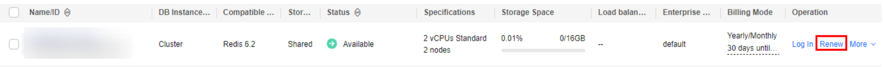
Renewing a Yearly/Monthly Instance

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance that you want to renew and click **Renew** in the **Operation** column.

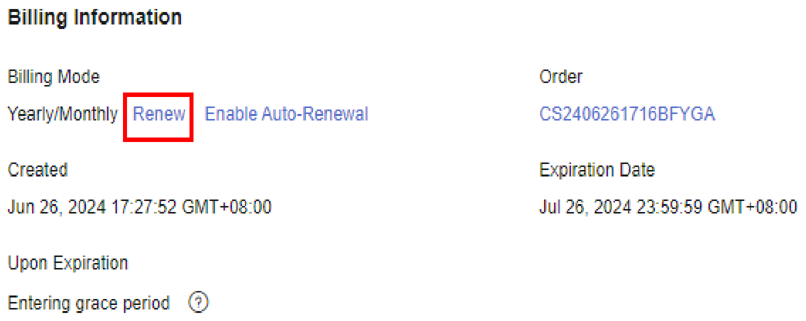
Figure 3-2 Renewing an instance



□	Name/ID	DB Instance...	Compatible ...	Stor...	Status	Specifications	Storage Space	Load balan...	Enterprise ...	Billing Mode	Operation
□	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More

Alternatively, click the instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Renew** next to the **Billing Mode** field.

Figure 3-3 Renewing an instance



Billing Information

Billing Mode
Yearly/Monthly Renew Enable Auto-Renewal

Order
CS2406261716BFYGA

Created
Jun 26, 2024 17:27:52 GMT+08:00

Expiration Date
Jul 26, 2024 23:59:59 GMT+08:00

Upon Expiration
Entering grace period

Step 4 On the displayed page, renew the instance.

----End

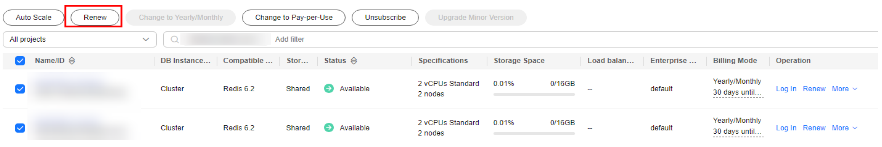
Renewing Multiple Instances in Batches

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, select the instance that you want to renew and click **Renew** above the instance list.

Figure 3-4 Batch renewing instances



☑	Name/ID	DB Instance...	Compatible ...	Stor...	Status	Specifications	Storage Space	Load balan...	Enterprise ...	Billing Mode	Operation
☑	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More
☑	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More

Step 4 In the displayed dialog box, click **Yes**.

----End

3.2.2 Changing the Billing Mode from Pay-per-Use to Yearly/ Monthly

This section describes how to change the billing mode of a GeminiDB Redis instance from pay-per-use to yearly/monthly. If you want to use a pay-per-use instance for a long time, change its billing mode to yearly/monthly to reduce costs.

Precautions

- Only when the status of a pay-per-use instance is **Available**, its billing mode can be changed to yearly/monthly.
- The function of batch changing the billing mode of pay-per-use instances to yearly/monthly is in the open beta test (OBT) phase. To use this function, contact customer service.

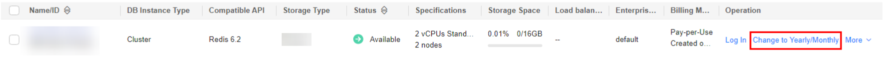
Changing the Billing Mode of a Single Instance

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose billing mode you want to change and click **Change to Yearly/Monthly** in the **Operation** column.

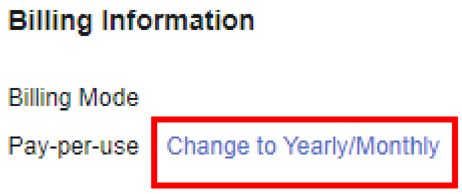
Figure 3-5 Changing from pay-per-use to yearly/monthly



□	Name/ID	DB Instance Type	Compatible API	Storage Type	Status	Specifications	Storage Space	Load balan...	Enterpris...	Billing M...	Operation
□	[redacted]	Cluster	Redis 6.2	[redacted]	Available	2 vCPUs Stand... 2 nodes	0.01% 0/16GB	--	default	Pay-per-Use Created o...	Log In Change to Yearly/Monthly More

Alternatively, click the instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Change to Yearly/Monthly** in the **Billing Mode** field.

Figure 3-6 Changing from pay-per-use to yearly/monthly



Billing Information

Billing Mode

Pay-per-use

Step 4 On the displayed page, specify a renewal duration in month. The minimum duration is one month.

Confirm the settings and click **Pay Now**.

Step 5 Select a payment method and click **Pay**.

Step 6 View the results on the **Instances** page.

In the upper right corner of the instance list, click ↻ to refresh the list. The instance status will become **Available** after the change is successful. The billing mode changes to **Yearly/Monthly**.

----End

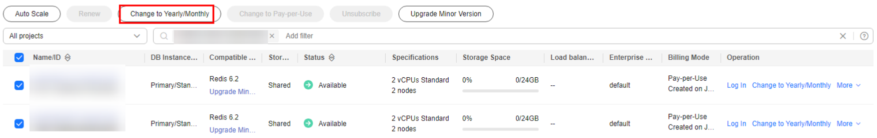
Changing the Billing Mode of Multiple Instance in Batches

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, select the instances whose billing mode you want to change and click **Change to Yearly/Monthly** above the instance list.

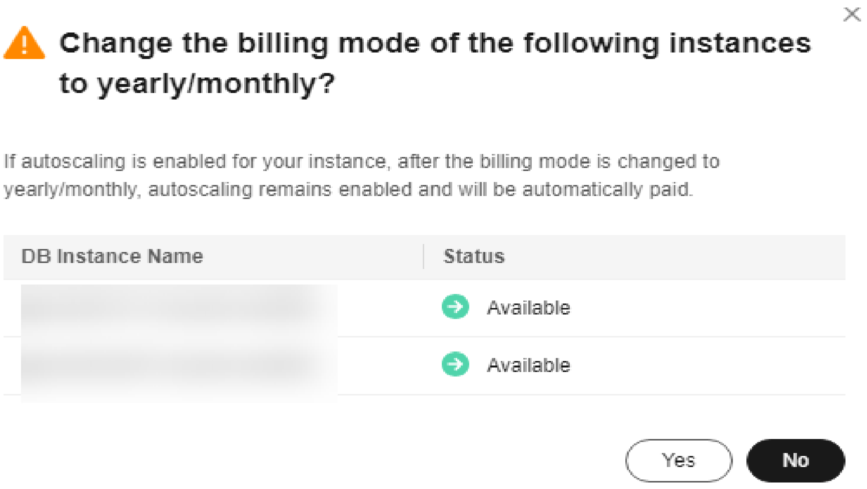
Figure 3-7 Changing the billing mode of multiple instances



☑	Name/ID	DB Instance...	Compatible ...	Stor...	Status	Specifications	Storage Space	Load balan...	Enterprise ...	Billing Mode	Operation
☑	[redacted]	Primary/Stan...	Redis 6.2 Upgrade Min...	Shared	Available	2 vCPUs Standard 2 nodes	0% 0/24GB	--	default	Pay-per-Use Created on J...	Log In Change to Yearly/Monthly More
☑	[redacted]	Primary/Stan...	Redis 6.2 Upgrade Min...	Shared	Available	2 vCPUs Standard 2 nodes	0% 0/24GB	--	default	Pay-per-Use Created on J...	Log In Change to Yearly/Monthly More

Step 4 In the displayed dialog box, click **Yes**.

Figure 3-8 Changing the billing mode to yearly/monthly



Change the billing mode of the following instances to yearly/monthly?

If autoscaling is enabled for your instance, after the billing mode is changed to yearly/monthly, autoscaling remains enabled and will be automatically paid.

DB Instance Name	Status
[redacted]	Available
[redacted]	Available

Step 5 On the displayed page, specify a renewal duration in month. The minimum duration is one month.

Confirm the settings and click **Pay Now**.

Step 6 Select a payment method and click **Pay**.

Step 7 View the results on the **Instances** page.

In the upper right corner of the instance list, click ↻ to refresh the list. The instance status will become **Available** after the change is successful. The billing mode changes to **Yearly/Monthly**.

----End

3.2.3 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

You can change the billing mode of a GeminiDB Redis instance from yearly/monthly to pay-per-use.

Precautions

- The billing mode of a yearly/monthly instance can only be changed to pay-per-use when the instance is in the **Available** status.

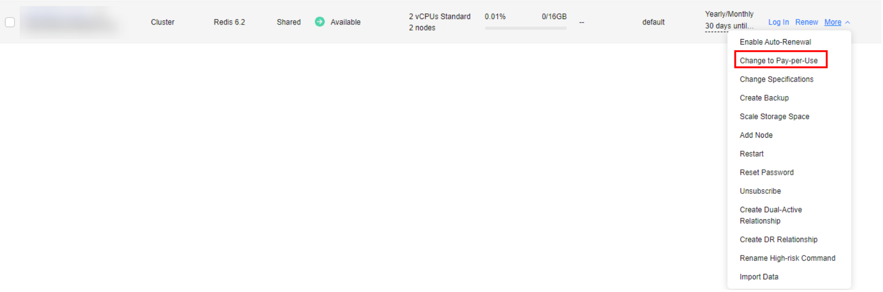
Changing the Billing Mode of a Single Instance to Pay-per-Use

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose billing mode you want to change and click **More > Change to Pay-per-Use** in the **Operation** column.

Figure 3-9 Changing from yearly/monthly to pay-per-use



□	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More

Step 4 On the displayed page, confirm the instance information and click **Change to Pay-per-Use**. The billing mode will change to pay-per-use after the instance expires.

NOTICE

Auto renewal will be disabled after the billing mode of your instances change to pay-per-use. Exercise caution when performing this operation.

Step 5 After you submit the change, a message is displayed in the **Billing Mode** column of the target DB instance, indicating that the billing mode will be changed to pay-per-use after the DB instance expires.

Step 6 To cancel the change, choose **Billing** > **Renewal** to enter the Billing Center. On the **Renewals** page, locate the target DB instance and click **More** > **Cancel Change to Pay-per-Use**.

Step 7 In the displayed dialog box, click **Yes**.

----End

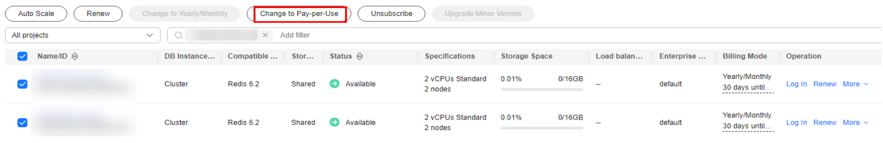
Changing the Billing Mode of Multiple Instances to Pay-per-use

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, select the instances whose billing mode you want to change and click **Change to Pay-per-Use** above the instance list.

Figure 3-10 Changing the billing mode of multiple instances to pay-per-use



☑	Name/ID	DB Instance...	Compatible ...	Stor...	Status	Specifications	Storage Space	Load balan...	Enterprise ...	Billing Mode	Operation
☑	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More
☑	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More

Step 4 In the displayed dialog box, click **Yes**.

Step 5 On the displayed page, confirm the instance information and click **Change to Pay-per-Use**. The billing mode will change to pay-per-use after the instance expires.

NOTICE

Auto renewal will be disabled after the billing mode of your instances change to pay-per-use. Exercise caution when performing this operation.

Step 6 After you submit the change, a message is displayed in the **Billing Mode** column of the target DB instance, indicating that the billing mode will be changed to pay-per-use after the DB instance expires.

Step 7 To cancel the change, choose **Billing** > **Renewal** to enter the Billing Center. On the **Renewals** page, locate the target DB instance and click **More** > **Cancel Change to Pay-per-Use**.

Step 8 In the displayed dialog box, click **Yes**.

----End

3.2.4 Unsubscribing from a Yearly/Monthly Instance

If you do not need a yearly/monthly instance any longer, unsubscribe from it.

Precautions

- Unsubscribed operations cannot be undone. Exercise caution when performing this operation. To retain data, create a manual backup before unsubscription. For details, see **Creating a Manual Backup**.
- After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved. Ensure that the manual backup is complete before submitting the unsubscription request.

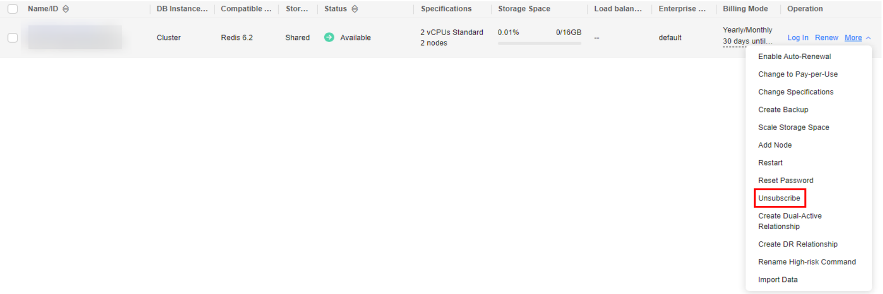
Unsubscribing from a Single Yearly/Monthly Instance

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance you want to unsubscribe and choose **More** > **Unsubscribe** in the **Operation** column.

Figure 3-11 Unsubscribing from a yearly/monthly instance



□	Name/ID	DB Instance...	Compatible ...	Stor...	Status	Specifications	Storage Space	Load balan...	Enterprise ...	Billing Mode	Operation
□	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More

Step 4 In the displayed dialog box, click **Yes**.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription result. After you unsubscribe from the instance order, the instance is no longer displayed in the instance list on the **Instances** page.

----End

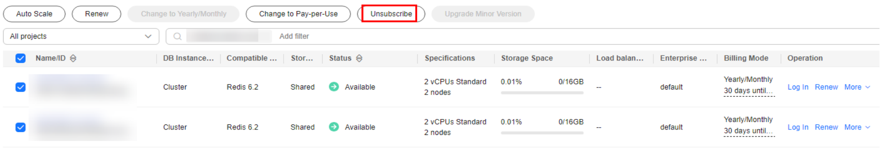
Unsubscribing from Multiple Yearly/Monthly Instances

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 Choose **Instances** in the navigation pane on the left, select the instances you want to unsubscribe from and click **Unsubscribe** above the instance list.

Figure 3-12 Unsubscribing from multiple yearly/monthly instances



☑	Name/ID	DB Instance...	Compatible ...	Stor...	Status	Specifications	Storage Space	Load balan...	Enterprise ...	Billing Mode	Operation
☑	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More
☑	[redacted]	Cluster	Redis 6.2	Shared	Available	2 vCPUs Standard 2 nodes	0.01% 0/16GB	--	default	Yearly/Monthly 30 days until...	Log In Renew More

Step 4 In the displayed dialog box, click **Yes**.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription result. After you unsubscribe from the instance order, the instance is no longer displayed in the instance list on the **Instances** page.

----End

3.3 Buying an Instance

3.3.1 Buying a Cluster Instance

This section describes how to buy a cluster Redis instance on the GeminiDB console.

Each tenant can have up to 50 GeminiDB Redis instances by default. To create more instances, contact customer service.

Prerequisites

- You have created a Huawei Cloud account.

Procedure

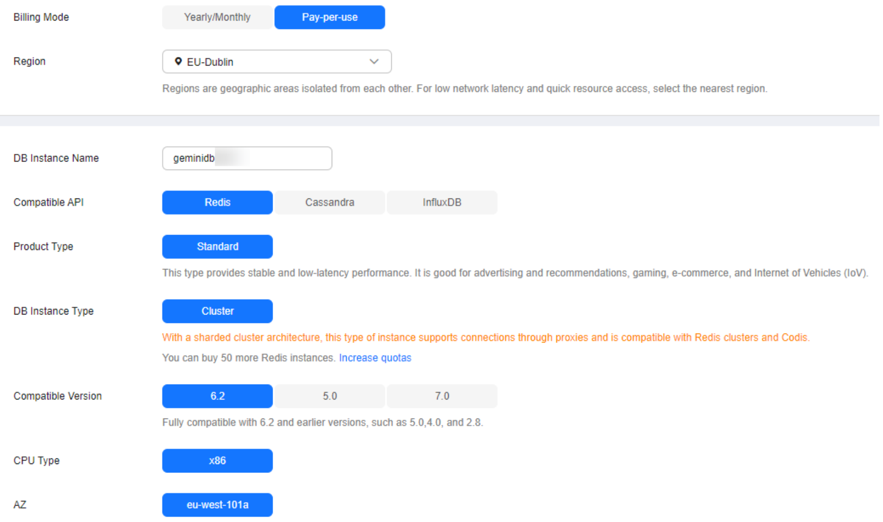
Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click **Buy DB Instance**.

Step 4 On the displayed page, specify instance specifications and click **Next**.

Figure 3-13 Billing mode and basic information



Billing Mode: Yearly/Monthly [✓ Pay-per-use]

Region: EU-Dublin

Regions are geographic areas isolated from each other. For low network latency and quick resource access, select the nearest region.

DB Instance Name: geminidb

Compatible API: [✓ Redis] Cassandra InfluxDB

Product Type: [✓ Standard]

This type provides stable and low-latency performance. It is good for advertising and recommendations, gaming, e-commerce, and Internet of Vehicles (IoV).

DB Instance Type: [✓ Cluster]

With a sharded cluster architecture, this type of instance supports connections through proxies and is compatible with Redis clusters and Codis.

You can buy 50 more Redis instances. Increase quotas

Compatible Version: [✓ 6.2] 5.0 7.0

Fully compatible with 6.2 and earlier versions, such as 5.0,4.0, and 2.8.

CPU Type: [✓ x86]

AZ: [✓ eu-west-101a]

Table 3-1 Billing mode description

Parameter	Description
Billing Mode	Select **Yearly/Monthly** or **Pay-per-use**. ● Yearly/Monthly – In this mode, specify **Required Duration** at the bottom of the page. The system bills you based on the service price. – If you do not need such an instance any longer after it expires, change the billing mode to pay-per-use to optimize costs. For details, see **Changing the Billing Mode from Yearly/Monthly to Pay-per-Use**. **NOTE** Yearly/Monthly instances cannot be deleted directly. If such an instance is no longer required, unsubscribe from it. For details, see **Unsubscribing from a Yearly/Monthly Instance**. ● Pay-per-use – If you select this billing mode, you are billed based on how much time the instance is in use. – If you expect to use an instance for a long period of time, change its billing mode to yearly/monthly to optimize costs. For details, see **Changing the Billing Mode from Pay-per-Use to Yearly/Monthly**.

Table 3-2 Basic information

Parameter	Description
Region	Region where a tenant is located **NOTICE** Select a region near to your service area to reduce network latency and experience faster access. Instances deployed in different regions cannot communicate with each other over a private network. After you buy an instance, you cannot change its region.
DB Instance Name	The instance name: ● Can be the same as an existing instance name. ● Contains 4 to 64 characters and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_). If the name contains Chinese characters, the length cannot exceed 64 bytes. You can change the name of an instance after it is created. For details, see **Modifying the Name of an Instance**.
Compatible API	Redis.

Parameter	Description
DB Instance Type	Cluster. With a sharded cluster architecture, this type of instance supports connections through proxies and is compatible with Redis clusters. It offers strong horizontal expansion capability, supporting millions of QPS and dozens of TB-level services.
Compatible Version	6.2 (including 6.2.*X*), 5.0, and earlier versions
Data Copies	The default value is **3**. Three copies of the data are created, and each copy always provides the same data as the other two to keep services available.
AZ	Availability zone where the instance is created. An AZ is a part of a region with its own independent power supplies and networks. AZs are physically isolated but can communicate with each other over a private network.

Figure 3-14 Specifications and storage

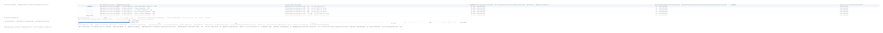


Table 3-3 Specifications and storage

Parameter	Description
Node Specifications	Instance node specifications, node quantity, and storage space
Nodes	Number of required nodes. After an instance is created, you can add nodes.
Total Storage Space	The value is an integer, and the minimum value is 32 GB. You can add a minimum of 1 GB at a time.

Table 3-4 Network

Parameter	Description
VPC	The virtual network where the instance is created. A VPC isolates networks for different services. You can select an existing VPC or create one. For details about how to create a VPC, see "Creating a VPC" in *Virtual Private Cloud User Guide*. If there are no VPCs available, the system allocates resources to you by default. **NOTE** ● After a GeminiDB Redis instance is created, its VPC cannot be changed. ● If you want to connect to a GeminiDB Redis instance using an ECS over a private network, the GeminiDB Influx instance and the ECS must be in the same VPC. If they are not, you can create a **VPC peering connection** between them.
Subnet	A subnet where your instance is created. The subnet provides dedicated and isolated networks, improving network security. **NOTE** Create an IPv4 subnet or select an existing one. IPv6 subnets are not supported.
Security Group	A security group controls access between GeminiDB Redis instances and other services. Ensure that the security group you selected allows your client to access the instance. If no security group is available, the system creates one for you.
SSL	A security protocol. Secure Sockets Layer (SSL) certificates set up encrypted connections between clients and servers, preventing data from being tampered with or stolen during transmission. You can enable SSL to improve data security. After an instance is created, you can connect to it using **SSL**. **NOTE** If SSL is not enabled when you create an instance, you can enable it after the instance is created. For details, see **Configuring an SSL Connection**.

Table 3-5 Database configuration

Parameter	Description
Database Password	Password of the administrator account. The password: ● Must be 8 to 32 characters long. ● Can include two of the following: uppercase letters, lowercase letters, digits, and special characters: ~!@#%^*-_=+? ● For security reasons, set a strong password. The system will verify the password strength. Keep your password secure. The system cannot retrieve it if it is lost.
Confirm Password	Enter the administrator password again.
Enterprise Project	This parameter is provided for enterprise users. An enterprise project groups cloud resources, so you can manage resources and members by project. The default project is **default**. Select an enterprise project from the drop-down list. For more information about enterprise projects, see ***Enterprise Management User Guide***.

Table 3-6 Tags

Parameter	Description
Tags	This setting is optional. Adding tags helps you better identify and manage your instances. Each instance supports up to 20 tags by default. A tag consists of a tag key and a tag value. ● Tag key: mandatory if the instance is going to be tagged Each tag key is unique for each instance. It can include up to 36 characters, including digits, letters, underscores (_), and hyphens (-). ● Tag value: optional if the instance is going to be tagged The value can contain up to 43 characters, including digits, letters, underscores (_), periods (.), and hyphens (-). After an instance is created, you can view its tag details on the **Tags** tab. In addition, you can add, modify, and delete tags of an existing instance. For details, see **Managing Tags**.

Table 3-7 Required duration

Parameter	Description
Required Duration	The length of your subscription if you select **Yearly/Monthly** billing. Subscription lengths range from one month to three years.
Auto-renew	● By default, this option is not selected. ● If you select this parameter, the auto-renew cycle is determined by the selected required duration.

Step 5 On the displayed page, confirm instance details.

- For yearly/monthly instances
  - To modify the configurations, click **Previous**.
  - If no modification is required, read and agree to the service agreement, click **Pay Now**, and complete the payment.
- Pay-per-use
  - To modify the configurations, click **Previous**.
  - If no modification is required, read and agree to the service agreement and click **Submit**.

Step 6 On the **Instances** page, view and manage the created instance.

The instance creation process takes about 5 to 15 minutes. After the creation is complete, the status changes to **Available**.

You can click ↻ in the upper right corner of the page to refresh the instance status.

After a DB instance is created, the default database port is **8635** and cannot be changed.

----End

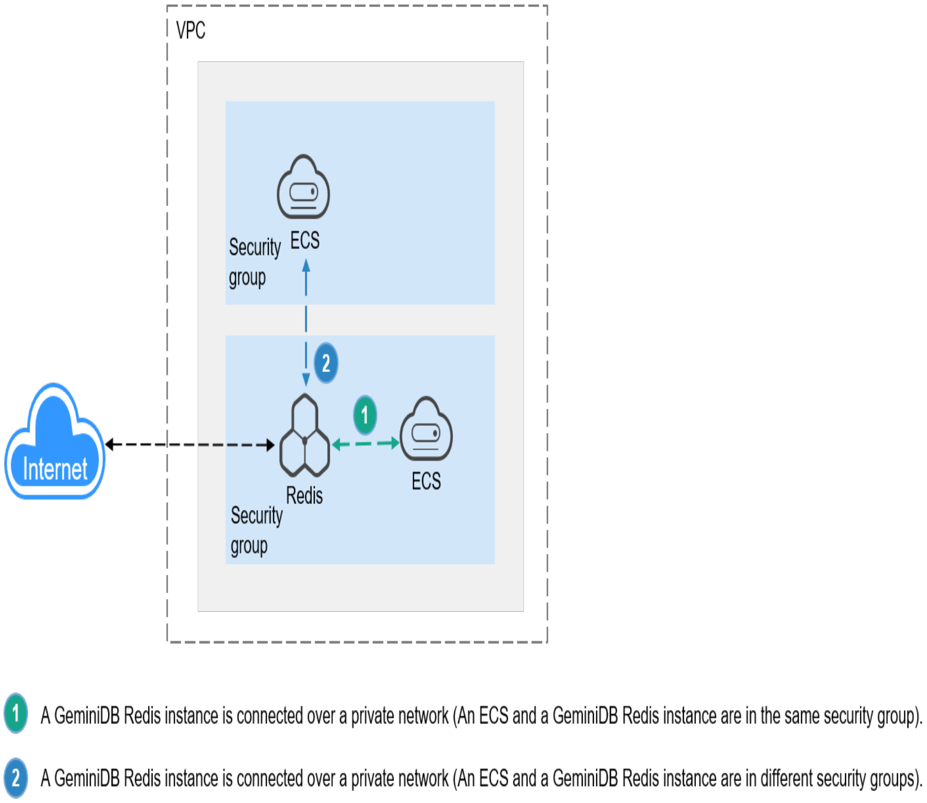
3.4 Connecting to an Instance

3.4.1 Connection Modes

GeminiDB Redis API is compatible with open-source Redis and allows traffic from applications using different types of SDKs. It can also be accessed through Data Admin Service (DAS), private networks, and public networks.

Figure 3-15 shows the process of connecting to a GeminiDB Redis instance.

Figure 3-15 Connection Methods



1 A GeminiDB Redis instance is connected over a private network (An ECS and a GeminiDB Redis instance are in the same security group).

2 A GeminiDB Redis instance is connected over a private network (An ECS and a GeminiDB Redis instance are in different security groups).

Table 3-8 Connection methods

Method	Scenario	Description
DAS	You can connect to a GeminiDB Redis instance using a web-based console client.	-

Method	Scenario	Description
Private network	You can connect to a GeminiDB Redis instance through a **private IP address**, **private domain name**, or **load balancer address**. This method is suitable when your application is deployed on an ECS that is in the same region and VPC as your instance.	● You are advised to use the load balancer address to connect to the instance. This ensures high reliability and eliminates the impact of SPOFs. ● High security and performance ● If the ECS and GeminiDB Redis instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured. ● If they are in different security groups, configure security group rules for them, separately. – Configure inbound rules of a security group for GeminiDB Redis instances by following **Configuring Security Group Rules for Nodes**. – The default security group rule allows all outbound data packets, so you do not need to configure a security rule for the ECS. If not all access from the ECS is allowed, you need to configure an outbound rule for the ECS.
Public network	You can connect to a GeminiDB Redis instance through a **public domain name** or an **EIP**. This method is suitable when an instance cannot be accessed over a private network. You can connect to the instance from an ECS using a public domain name or an EIP.	● For faster transmission and improved security, migrate your applications to an ECS that is in the same subnet as your instance and use a private IP address to access the instance. ● Use a public domain name to ensure high reliability and eliminate SPOFs. ● .

3.4.2 Connecting to GeminiDB Redis Instances Through DAS

DAS enables you to manage DB instances from a web-based console, simplifying database management and improving efficiency. You can connect and manage instances through DAS. By default, you have permissions required for remote login. DAS is secure and convenient for connecting to DB instances.

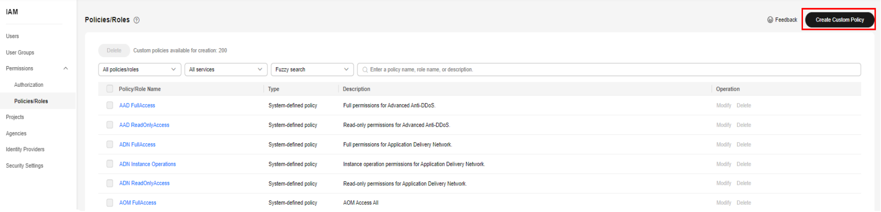
Configuring the Required Permissions

If you have an IAM account, assign DAS FullAccess permissions to all users of the account. For details, see **Create User Groups and Assign Permissions**.

You can create a custom policy to specify the type of databases that you have permissions for.

1. Log in to the IAM console and choose **Permissions** > **Policies/Roles**.

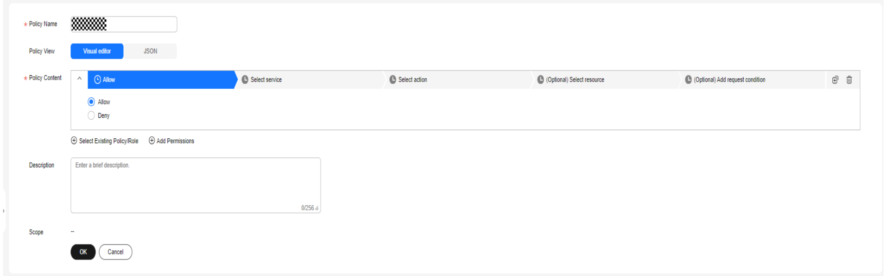
Figure 3-16 Creating a custom policy



	Policy/Role Name	Type	Description	Operation
□	AAD FullAccess	System-defined policy	Full permissions for Advanced Anti-DDoS.	Modify Delete
□	AAD ReadOnlyAccess	System-defined policy	Read-only permissions for Advanced Anti-DDoS.	Modify Delete
□	ADN FullAccess	System-defined policy	Full permissions for Application Delivery Network.	Modify Delete
□	ADN Instance Operations	System-defined policy	Instance operation permissions for Application Delivery Network.	Modify Delete
□	ADN ReadOnlyAccess	System-defined policy	Read-only permissions for Application Delivery Network.	Modify Delete
□	AOM FullAccess	System-defined policy	AOM Access All	Modify Delete

2. Specify a policy name, policy view, and content.

Figure 3-17 Configuring a custom policy


Visual editor
JSON
Allow
Select service
Select action
(Optional) Select resource
(Optional) Add request condition
Select Existing Policy/Role
Add Permissions
OK
Cancel

* Policy Name [redacted]

Policy View: Visual editor

* Policy Content: Allow

● Allow
○ Deny

Description: Enter a brief description. 0/256

Scope: --

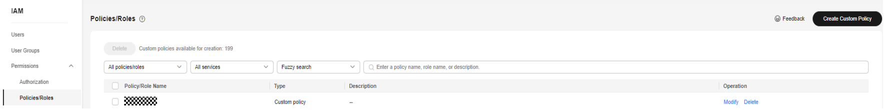
Table 3-9 Custom policy description

Parameter	Description
Policy Name	Enter a policy name.
Policy View	Select **JSON**.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Policy Content</td><td>Configure the following policy content:
<pre>
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "das:*:*",
                "nosql:instance:list"
            ],
            "Effect": "Allow"
        }
    ]
}
</pre>
Alternatively, click <b>Select Existing Policy/Role</b>, select <b>DAS FullAccess</b> as a template, and retain only the DB type information. In this example, retain only <b>nosql:instance:list</b>.</td></tr>
<tr><td>Description</td><td>Enter a policy description.</td></tr>
<tr><td>Scope</td><td>Retain the default settings (project-level service).</td></tr>
</table>

3. Click **OK**. You can then view the created custom policy on the **Permissions** page.

Figure 3-18 Viewing the created policy

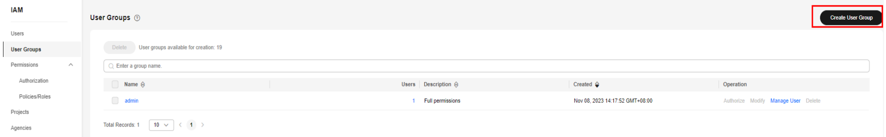


Custom policies available for creation: 199

Policy/Role Name	Type	Description	Operation
[redacted]	Custom policy	--	Modify Delete

4. Create a user group.

Figure 3-19 Creating a user group



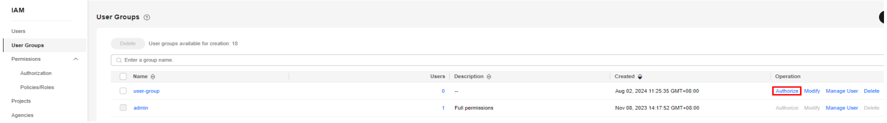
User groups available for creation: 19

Name	Users	Description	Created	Operation
admin	1	Full permissions	Nov 08, 2023 14:17:52 GMT+08:00	Authorize Modify Manage User Delete

Total Records: 1 10 1

5. Authorize the user group created in 4 using the created custom policy.

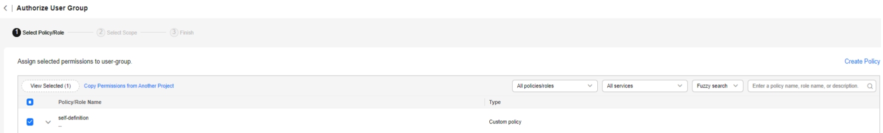
Figure 3-20 Authorizing the user group using the created custom policy



User groups available for creation: 18

Name	Users	Description	Created	Operation
user-group	0	--	Aug 02, 2024 11:25:35 GMT+08:00	Authorize Modify Manage User Delete
admin	1	Full permissions	Nov 08, 2023 14:17:52 GMT+08:00	Authorize Modify Manage User Delete

Figure 3-21 Selecting the created custom policy

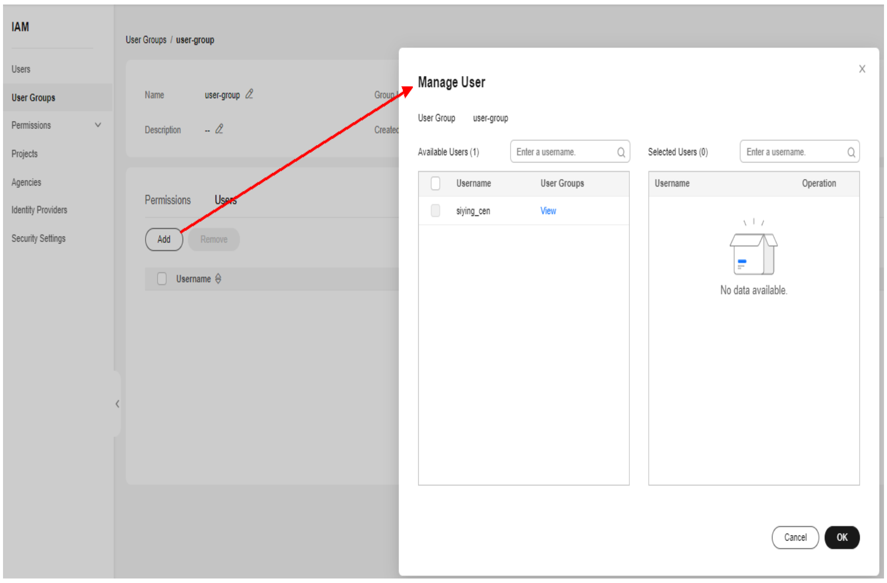


Assign selected permissions to user-group.

	Policy/Role Name	Type
☑	self-definition --	Custom policy

6. Click the name of the user group and add the required users.

Figure 3-22 Adding users



Prerequisites

There is an available GeminiDB Redis instance.

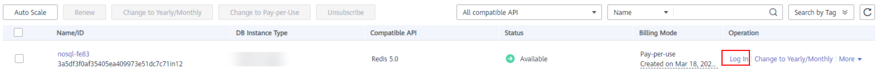
Procedure

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 In the instance list, locate the target instance and click **Log In** in the **Operation** column.

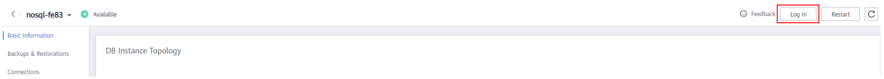
Figure 3-23 Connecting to a GeminiDB Redis instance



	Name/ID	DB Instance Type	Compatible API	Status	Billing Mode	Operation
□	nosql-fe83 3a5df3f0af35405ea409973e51dc7c71in12		Redis 5.0	Available	Pay-per-use Created on Mar 18, 202...	Log In | Change to Yearly/Monthly | More

Alternatively, click the instance name to go to the **Basic Information** page. Click **Log In** in the upper right corner of the page.

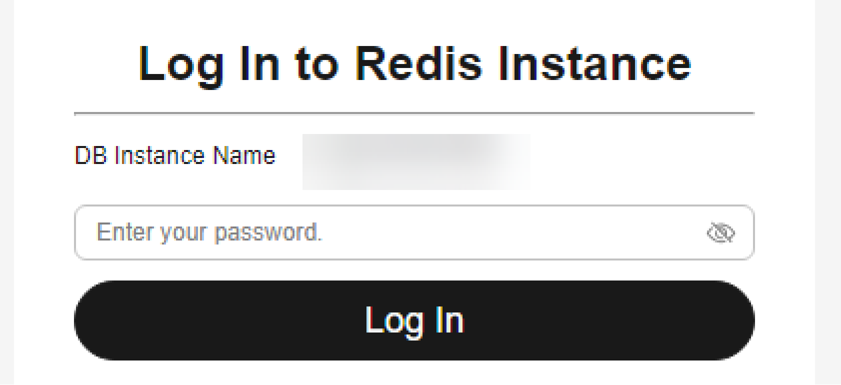
Figure 3-24 Connecting to a GeminiDB Redis instance



Step 4 Enter the password for logging in to the instance.

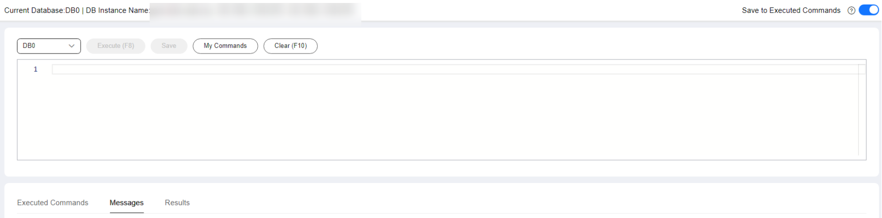
You need to enter the password only when you log in to a GeminiDB Redis instance for the first time or after you reset the password.

Figure 3-25 Login page



Step 5 Manage relevant databases.

Figure 3-26 Instance homepage

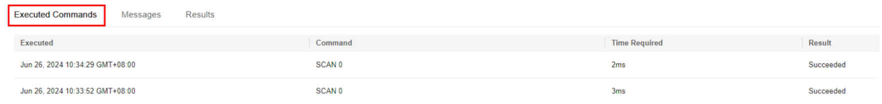


- Save to Executed Commands

  This function is enabled by default to save the recently executed commands for your later query.

  Then you can click the **Executed Commands** tab on the lower page to view historical commands.

  Figure 3-27 The Executed Commands tab

Executed	Command	Time Required	Result
Jun 26, 2024 10:34:29 GMT+08:00	SCAN 0	2ms	Succeeded
Jun 26, 2024 10:33:52 GMT+08:00	SCAN 0	3ms	Succeeded

  If this function is disabled, the commands executed subsequently are not displayed any longer. You can click ● next to **Save Executed SQL Statements** in the upper right corner to disable this function.

- Execute a command.

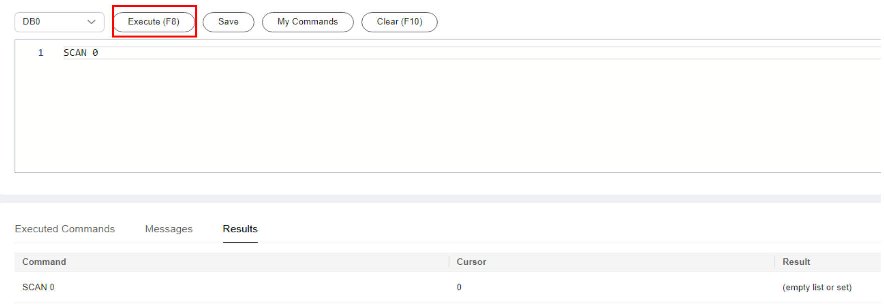
  You can enter a command in the command window and click **Execute** or **F8**.

  NOTE

  – Do not use transactions, Lua scripts, Pub/Sub commands, or other commands that have blocking semantics.

  – For an instance that supports multiple databases, you can change the current database on the console, but cannot change it using a SELECT statement.

Figure 3-28 Executing a command



```
SCAN 0
```

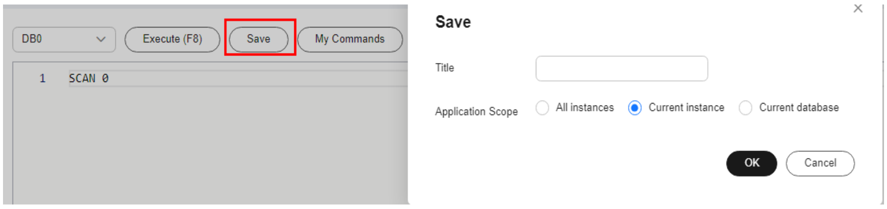
Command	Cursor	Result
SCAN 0	0	(empty list or set)

After a command is executed, you can view the execution result on the **Results** page.

- Save

  You can save a command to all instances, the current instance, or the current database. Then you can view details in **My Commands**.

Figure 3-29 Save



```
SCAN 0
```

Title

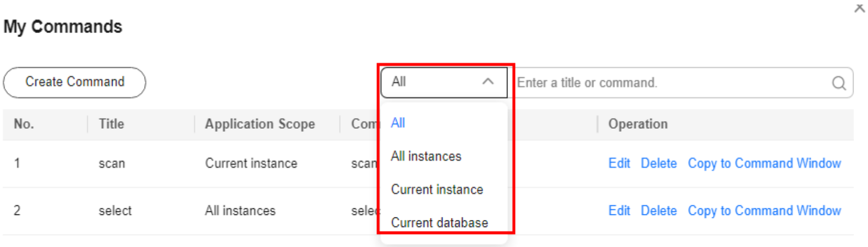
Application Scope ○ All instances ● Current instance ○ Current database

- My Commands

  Common commands are displayed the **My Commands** page.

  Set a filter to narrow the scope of commands. If you select **All**, all commands saved in the current account are displayed.

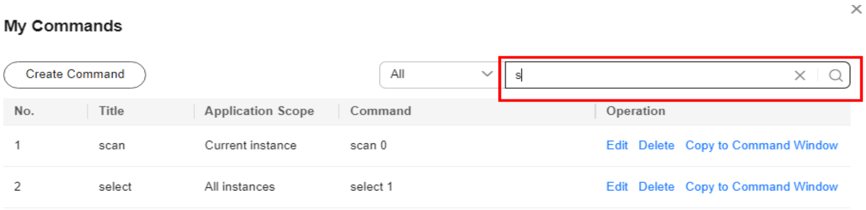
Figure 3-30 Filtering commands



No.	Title	Application Scope	Com	Operation
1	scan	Current instance	scan	Edit Delete Copy to Command Window
2	select	All instances	sele	Edit Delete Copy to Command Window

Alternatively, enter a command title or statement in the search box to search for the corresponding command.

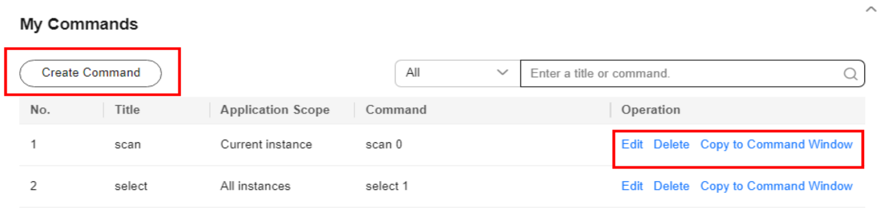
Figure 3-31 Searching for a command



No.	Title	Application Scope	Command	Operation
1	scan	Current instance	scan 0	Edit Delete Copy to Command Window
2	select	All instances	select 1	Edit Delete Copy to Command Window

On the **My Commands** page, you can also create, edit, and delete a command or copy it to the **Execute** window.

Figure 3-32 Managing a command

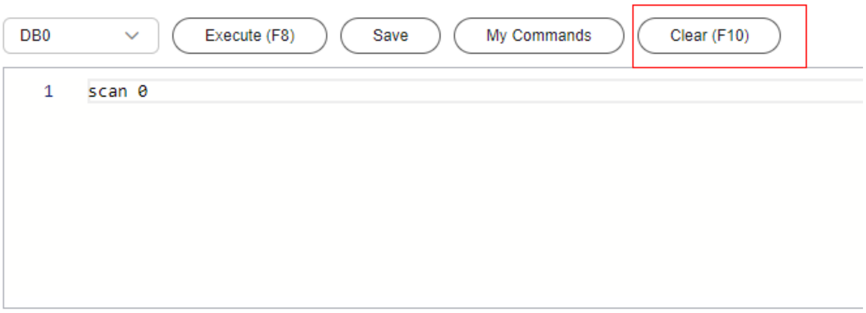


No.	Title	Application Scope	Command	Operation
1	scan	Current instance	scan 0	Edit Delete Copy to Command Window
2	select	All instances	select 1	Edit Delete Copy to Command Window

- Clear

  You can also press **F10** to clear the command in the execution window.

Figure 3-33 Clearing a command



```
scan 0
```

----End

3.4.3 Connecting to GeminiDB Redis Instances over a Private Network

3.4.3.1 Connecting to an Instance Using a Load Balancer Address (Recommended)

This section describes how to connect to a GeminiDB Redis instance using a load balancer address on a Linux ECS. Load balancing can improve data reliability and eliminate POFs.

Precautions

- The target instance must be in the same VPC and subnet as the ECS.
- The ECS must be in a security group that has access to the instances.

  Scenario 1: If the instance is associated with the default security group, you do not need to configure security group rules.

  Scenario 2: If the instance is not associated with the default security group, check whether the security group rules allow the ECS to connect to the instance.

  – If yes, the ECS can connect to the instance.

– If no, add an inbound rule to the security group.

For details about how to configure a security group, see **Configuring Security Group Rules for Nodes**.

Prerequisites

- An ECS has been created. The following uses a Linux ECS as an example. For details, see **Purchasing an ECS** in *Getting Started with Elastic Cloud Server*.
- Download the **Redis client installation package**.

Procedure

Step 1 Log in to the ECS. For details, see **Logging In to an ECS** in *Getting Started with Elastic Cloud Server*.

Step 2 Obtain the Redis client.

Method 1

Run the following command to download the Redis client.

wget http://download.redis.io/releases/redis-6.2.0.tar.gz

Method 2

Download the Redis client from the address provided in **Prerequisites** and upload the Redis client installation package to the ECS.

Step 3 Decompress the client tool package.

tar -xzf redis-6.2.0.tar.gz

Step 4 Connect to the instance in the **src** directory.

cd redis-6.2.0

make

cd src

./redis-cli -h *<DB_HOST>* **-p** *<DB_PORT>* **-a** *<DB_PWD>*

Example:

./redis-cli -h 192.xx.xx.xx -p 8635 -a *<DB_PWD>*

Table 3-10 Parameter description

Parameter	Description
<DB_HOST>	Load balancer IP address of the instance to be connected. After the load balancer IP address is created, click the instance name to go to the **Basic Information** page and obtain the load balancer IP address in the **Connection Information** area.

Parameter	Description
<DB_PORT>	Access port corresponding to the load balancer IP address of the instance. The procedure is as follows: Click the name of the instance to go to the **Basic Information** page. In the **Connection Information** area, you can find the access port in field **Database Port**.
<DB_PWD>	Administrator password set when you buy a GeminiDB Redis instance

Step 5 Check the results. If the following information is displayed, the connection is successful.

```
IP:port>
```

----End

3.4.3.2 Connecting to an Instance Using a Private Domain Name

This section describes how to connect to a GeminiDB Redis instance using a private domain name on a Linux ECS.

Precautions

- The target instance must be in the same VPC and subnet as the ECS.
- The ECS must be in a security group that has access to the instances.

  Scenario 1: If the instance is associated with the default security group, you do not need to configure security group rules.

  Scenario 2: If the instance is not associated with the default security group, check whether the security group rules allow the ECS to connect to the instance.

  – If yes, the ECS can connect to the instance.
  – If no, add an inbound rule to the security group.

    For details about how to configure a security group, see **Configuring Security Group Rules for Nodes**.

Prerequisites

- An ECS has been created. The following uses a Linux ECS as an example. For details, see **Purchasing an ECS** in *Getting Started with Elastic Cloud Server*.
- Download the **Redis client installation package**.

Procedure

Step 1 Configure the private domain name for the GeminiDB Redis instance. For details, see **Configuring a Private Domain Name**.

Step 2 Log in to the ECS. For details, see **Logging In to an ECS** in *Getting Started with Elastic Cloud Server*.

Step 3 Obtain the Redis client.

Method 1

Run the following command to download the Redis client.

wget http://download.redis.io/releases/redis-6.2.0.tar.gz

Method 2

Download the Redis client from the address provided in **Prerequisites** and upload the Redis client installation package to the ECS.

Step 4 Decompress the client tool package.

tar -xzf redis-6.2.0.tar.gz

Step 5 Connect to the DB instance in the **src** directory.

cd redis-6.2.0

make

cd src

./redis-cli -h <*DB_Domain_Name*> **-p** <*DB_PORT*> **-a** <*DB_PWD*>

Example:

./redis-cli -h redis.com -p 8635 -a *<DB_PWD>*

Table 3-11 Parameter description

Parameter	Description
<DB_Domain_Name>	The private domain name of the DB instance to be connected. The private domain name is the one created in **Step 1**.
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the instance port number, perform the following steps: Click the target instance to go to the **Basic Information** page. In the **Network Information** area, you can find the database port.
<DB_PWD>	Administrator password set when you buy a GeminiDB Redis instance

Step 6 Check the results. If the following information is displayed, the connection is successful.

```
Domain_Name:port>
```

----End

3.4.3.3 Connecting to an Instance Using a Private IP Address

You can use the private IP address to connect to the GeminiDB Redis instance.

This section uses the Linux OS as an example to describe how to connect to a GeminiDB Redis instance using the Redis-cli client. This section describes how to connect to a GeminiDB Redis instance in non-SSL mode.

Precautions

- The target instance must be in the same VPC and subnet as the ECS.
- The ECS must be in a security group that has access to the instances. For details, see **Configuring Security Group Rules for Nodes**.
- To connect to a DB instance over a non-SSL connection, SSL must be disabled. For details about how to disable SSL, see **Configuring an SSL Connection**.

Prerequisites

An ECS has been created. The following uses a Linux ECS as an example. For details, see **Purchasing an ECS** in *Getting Started with Elastic Cloud Server*.

Procedure

Step 1 Log in to the ECS. For details, see **Logging In to an ECS** in *Getting Started with Elastic Cloud Server*.

Step 2 Obtain the Redis client.

Method 1

Run the following command to download the Redis client.

```
wget http://download.redis.io/releases/redis-6.2.0.tar.gz
```

Method 2

Download the **Redis client** and upload it to the ECS.

Step 3 Decompress the client tool package.

```
tar -xzf redis-6.2.0.tar.gz
```

Step 4 Connect to the DB instance in the **src** directory.

```
cd redis-6.2.0
make
cd src
./redis-cli -h <DB_HOST> -p <DB_PORT> -a <DB_PWD>
```

Example:

```
./redis-cli -h 192.xx.xx.xx -p 8635 -a <DB_PWD>
```

Table 3-12 Parameter description

Parameter	Description
<DB_HOST>	The private IP address of the instance to be connected. To obtain this IP address, go to the **Instance Management** page and click the target DB instance name. The IP address can be found in the **Private IP Address** field under **Node Information** on the **Basic Information** page. If the instance you purchased has multiple nodes, select the private IP address of any node.
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the instance port number, perform the following steps: Click the target instance to go to the **Basic Information** page. In the **Network Information** area, you can find the database port.
<DB_PWD>	Administrator password set when you buy a GeminiDB Redis instance

Step 5 Check the results. If the following information is displayed, the connection is successful.

```
IP:port>
```

----End

3.4.4 Connecting to GeminiDB Redis Instances over a Public Network

3.4.4.1 Connecting to an Instance Using an EIP

You can connect to a GeminiDB Redis instance from an ECS or a local device over a public network.

This section uses the Linux OS as an example to describe how to connect to a GeminiDB Redis instance using the Redis-cli client. You can connect to a GeminiDB Redis instance to avoid SPOFs and achieve load balancing in the production environment.

You can connect to an instance over SSL or non-SSL connections. SSL encrypts data and is more secure. For details, see **Connecting to a instance Using SSL**. This section describes how to connect to a GeminiDB Redis instance over a non-SSL connection.

Precautions

- To connect to a DB instance over a non-SSL connection, SSL must be disabled. For details about how to disable SSL, see **Configuring an SSL Connection**.

- You need to estimate the bandwidth required by services and purchase an EIP with sufficient bandwidth resources. **Client access exceptions caused by poor public network performance will not be included in the SLA.**

Prerequisites

1. An ECS has been created. The following uses a Linux ECS as an example. For details, see **Purchasing an ECS** in *Getting Started with Elastic Cloud Server*.
2. You have bound an EIP to a node of the purchased instance and configure security group rules for the node. For details, see **Binding and Unbinding an EIP** and **Configuring Security Group Rules for Nodes**.

   NOTE

   A GeminiDB Redis instance can have multiple nodes. Select any node and bind an EIP to it.

Procedure

Step 1 Log in to the ECS. For details, see **Logging In to an ECS** in *Getting Started with Elastic Cloud Server*.

Step 2 Obtain the Redis client.

Method 1

Run the following command to download the Redis client.

```
wget http://download.redis.io/releases/redis-6.2.0.tar.gz
```

Method 2

Download the **Redis client** and upload it to the ECS.

Step 3 Decompress the client package.

```
tar -xzf redis-6.2.0.tar.gz
```

Step 4 Connect to the DB instance in the **src** directory.

```
cd redis-6.2.0
make
cd src
./redis-cli -h <DB_HOST> -p <DB_PORT> -a <DB_PWD>
```

Example:

```
./redis-cli -h 192.168.0.208 -p 8635 -a <DB_PWD>
```

Table 3-13 Parameter description

Parameter	Description
<DB_HOST>	EIP bound to the instance to be connected. To obtain the EIP, go to the **Instance Management** page and click the target instance name. The EIP can be found in the **EIP** column in the **Node Information** area on the **Basic Information** page. If the instance you bought has multiple nodes, you can bind the EIP to any node to connect to the instance. If a message is displayed indicating that no EIP has been bound to the instance, bind an EIP to the instance by following **Binding and Unbinding an EIP**.
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the instance port number, perform the following steps: Click the target instance to go to the **Basic Information** page. In the **Network Information** area, you can find the database port.
<DB_PWD>	Administrator password set when you buy a GeminiDB Redis instance

Step 5 Check the results. If the following information is displayed, the connection is successful.

```
IP:port>
```

----End

3.4.4.2 Connecting to an Instance Using a Public Domain Name

A public domain name is a domain name used to access websites or web applications on the Internet.

You can use Domain Name Service (DNS) to translate common domain names (for example, www.example.com) into IP addresses (for example, 1.2.3.4) required for network connection. In this way, you can access GeminiDB Redis instances using the resolved IP addresses.

This section uses the Linux OS as an example to describe how to use the public network domain name configured by the DNS service to connect to a GeminiDB Redis instance.

Prerequisites

- An ECS has been created. The following uses a Linux ECS as an example. For details, see **Purchasing an ECS** in *Getting Started with Elastic Cloud Server*.
- You have registered a domain name and an EIP.

- You have bound an EIP to a node of the purchased instance and configure security group rules for the node. For details, see **Binding and Unbinding an EIP** and **Configuring Security Group Rules for Nodes**.

  NOTE

  A GeminiDB Redis instance can have multiple nodes. Select any node and bind an EIP to it.

- Download the **Redis client installation package**.

Procedure

Step 1 Configure the private domain name for the GeminiDB Redis instance. For details, see **Configuring a Public Domain Name**.

Step 2 Log in to the ECS. For details, see **Logging In to an ECS** in *Getting Started with Elastic Cloud Server*.

Step 3 Obtain the Redis client.

Method 1

Run the following command to download the Redis client.

wget http://download.redis.io/releases/redis-6.2.0.tar.gz

Method 2

Download the Redis client from the address provided in **Prerequisites** and upload the Redis client installation package to the ECS.

Step 4 Decompress the client tool package.

tar -xzf redis-6.2.0.tar.gz

Step 5 Open the **src** directory and connect to the DB instance.

cd redis-6.2.0

make

cd src

./redis-cli -h <*DB_Domain_Name*> **-p** <*DB_PORT*> **-a** <*DB_PWD*>

Example:

./redis-cli -h redis.com -p 8635 -a *<DB_PWD>*

Table 3-14 Parameter description

Parameter	Description
<DB_Domain_Name>	The public domain name of the instance to be connected. The public domain name is the one created in **Step 1**.

Parameter	Description
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the instance port number, perform the following steps: Click the target instance to go to the **Basic Information** page. In the **Network Information** area, you can find the database port.
<DB_PWD>	Administrator password set when you buy a GeminiDB Redis instance

Step 6 Check the results. If the following information is displayed, the connection is successful.

```
Domain_Name:port>
```

----End

3.4.5 Configuring a Private Domain Name

This section describes how to configure and resolve private domain names.

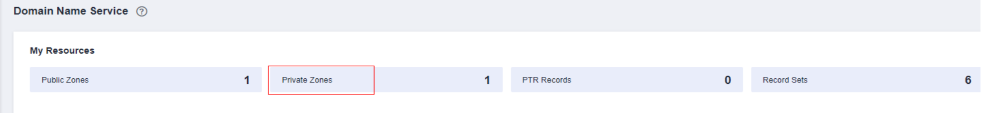
Creating a Private Domain Name

Step 1 Log in to the management console.

Step 2 Click **Service List**. Under **Network**, click **Domain Name Service**.

Step 3 On the DNS console, choose **Private Zones**.

Figure 3-34 Private zones



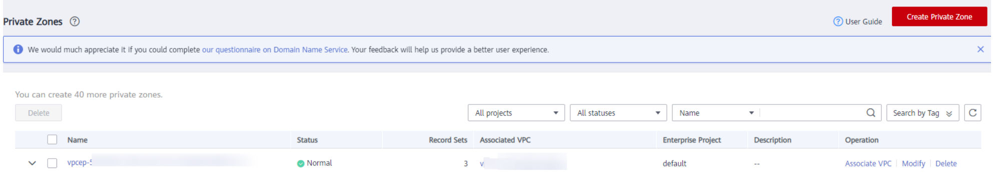
Domain Name Service

My Resources

Public Zones	1
Private Zones	1
PTR Records	0
Record Sets	6

Step 4 Click **Create Private Zone**.

Figure 3-35 Creating a private domain name



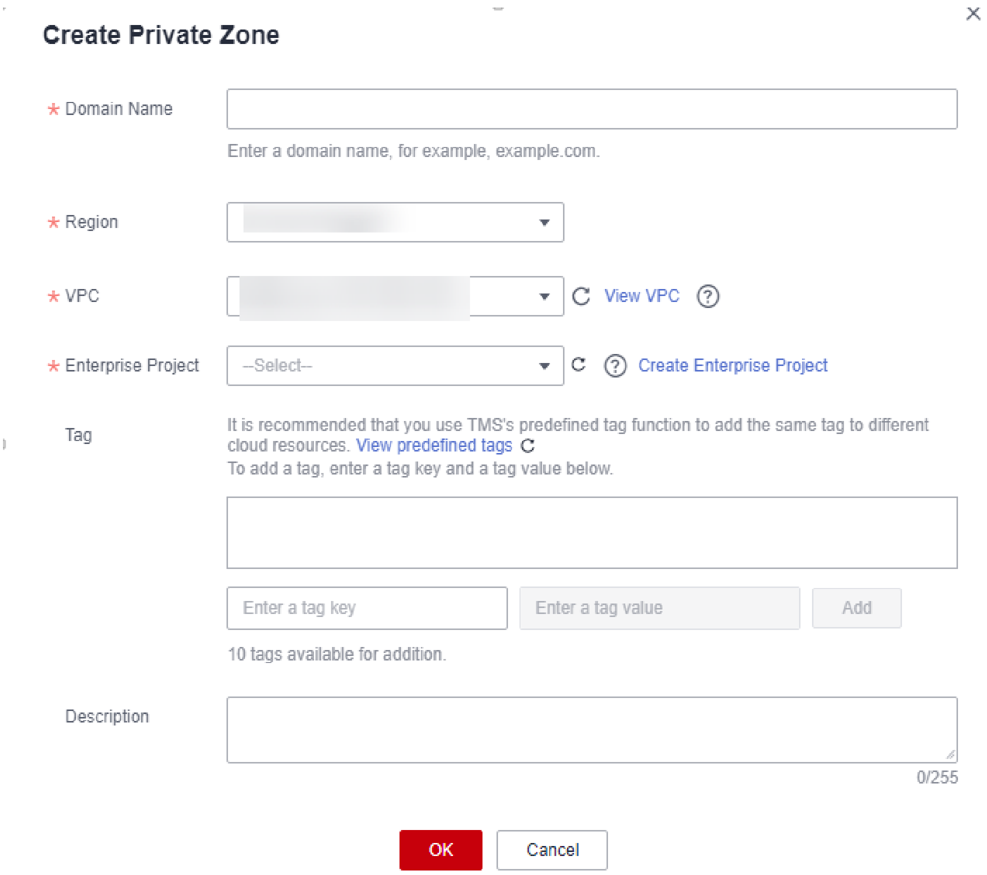
We would much appreciate it if you could complete our questionnaire on Domain Name Service. Your feedback will help us provide a better user experience.

You can create 40 more private zones.

	Name	Status	Record Sets	Associated VPC	Enterprise Project	Description	Operation
□	vpcep-[redacted]	Normal	3	v[redacted]	default	--	Associate VPC | Modify | Delete

Step 5 Configure parameters for creating a private domain name.

Figure 3-36 Creating a private zone



Enter a domain name, for example, example.com.

It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

To add a tag, enter a tag key and a tag value below.

10 tags available for addition.

Table 3-15 Parameter description

Parameter	Description	Example
Domain Name	Domain name of the private zone. You can customize any correctly formatted domain names, even top-level ones. For details about the domain name format, see **Domain Name Format and DNS Hierarchy**.	example.com
Region	The region where the tenant is located.	CN East-Shanghai1

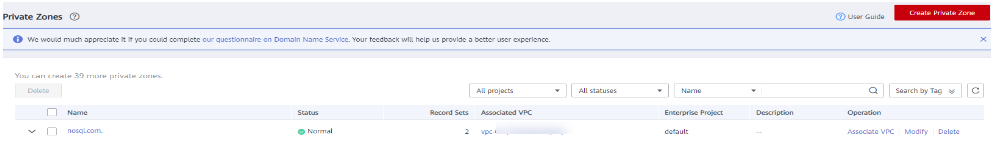
Parameter	Description	Example
VPC	The VPC associated with the private domain name must be the same as the VPC where the GeminiDB Redis instance is located. Otherwise, the private domain name cannot be resolved.	-
Enterprise Project	Enterprise project associated with the private domain name. You can manage private domain names by enterprise project. **NOTE** This parameter is available and mandatory only when **Account Type** is set to **Enterprise Account**. Configuration principles: ● If you do not manage domain names by enterprise project, select the **default** enterprise project. ● If you manage domain names by enterprise project, select an existing enterprise project.	default

Parameter	Description	Example
Tags	(Optional) Identifier of a resource. Each tag contains a key and a value. You can add a maximum of 10 tags to a domain name. The key and value naming rules are as follows: **Key**: - Cannot be left blank. - Must be unique for each resource. - Consists of a maximum of 36 characters. - Cannot start or end with a space or contain special characters =*<>\,|/ **Value**: - Cannot be left blank. - Consists of a maximum of 43 characters. - Cannot start or end with a space or contain special characters =*<>\,|/	example_key1 example_value1
Description	(Optional) Description of the zone, which cannot exceed 255 characters.	This is a zone example.

Step 6 Click **OK**. On the **Private Zones** page, view the created private domain name in the zone list.

If the status of the private domain name is **Normal**, the domain name has been successfully created.

Figure 3-37 Viewing the private domain name status



Private Zones ⓘ

ⓘ We would much appreciate it if you could complete our questionnaire on Domain Name Service. Your feedback will help us provide a better user experience.

You can create 39 more private zones.

	Name	Status	Record Sets	Associated VPC	Enterprise Project	Description	Operation
□	nosql.com.	● Normal	2	vpc-[redacted]	default	--	Associate VPC | Modify | Delete

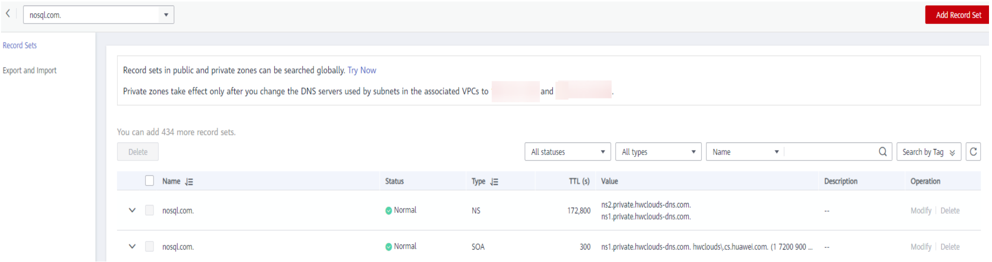
----End

Adding a Record Set for a Domain Name

After creating a private domain name, configure a record set for it so that you can access instances using the domain name.

Step 1 Click the private domain name you created. On the displayed page, click **Add Record Set** in the upper right corner.

Figure 3-38 Adding a record set



Record sets in public and private zones can be searched globally. Try Now

Private zones take effect only after you change the DNS servers used by subnets in the associated VPCs to [redacted] and [redacted].

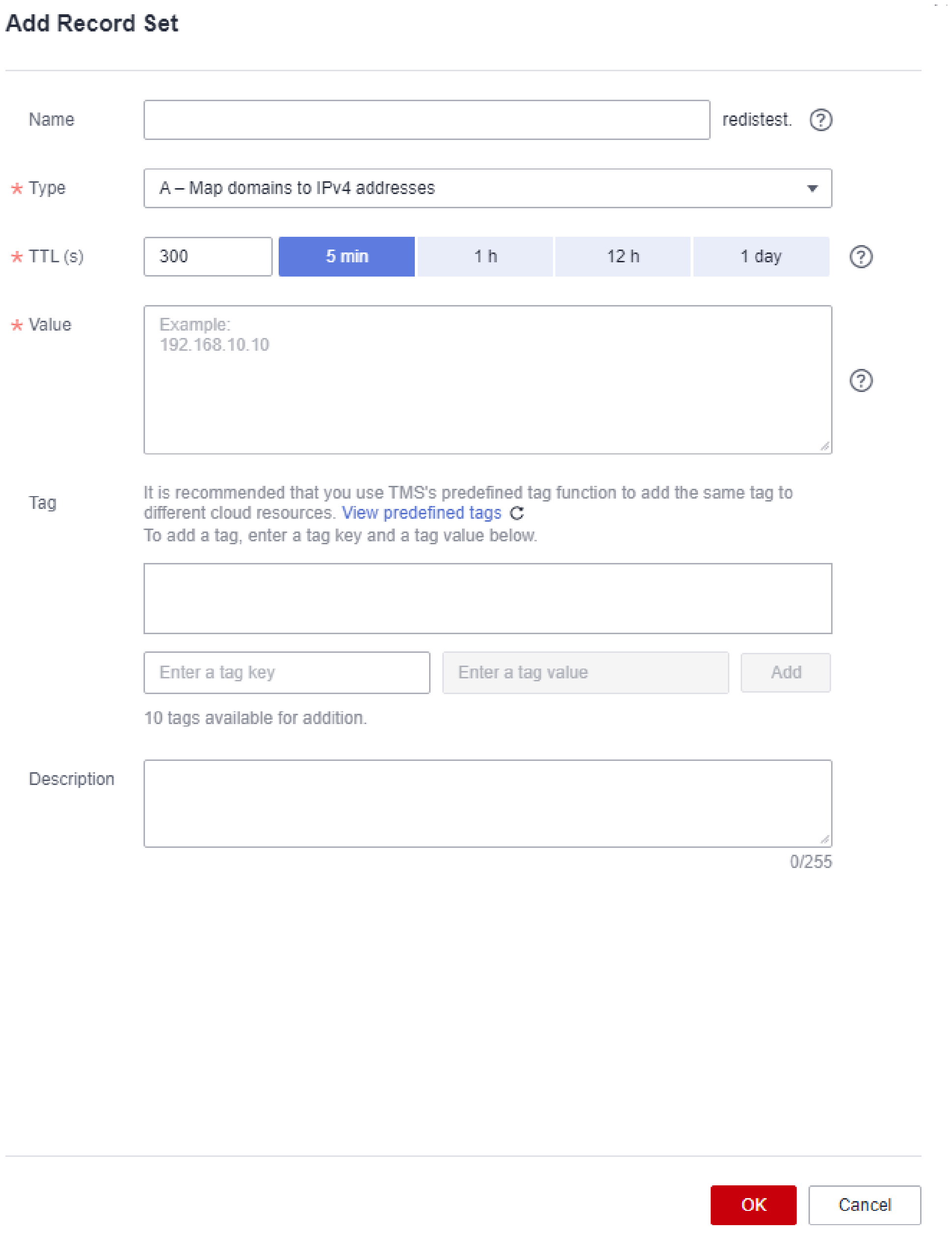
You can add 434 more record sets.

Name	Status	Type	TTL (s)	Value	Description	Operation
nosql.com.	Normal	NS	172,800	ns2.private.hwclouds-dns.com. ns1.private.hwclouds-dns.com.	--	Modify | Delete
nosql.com.	Normal	SOA	300	ns1.private.hwclouds-dns.com. hwclouds\.cs.huawei.com. (1 7200 900 ...	--	Modify | Delete

Step 2 In the displayed **Add Record Set** dialog box, configure the required parameters.

Value: Enter the load balancer IP address of the instance.

Figure 3-39 Adding a record set



Add Record Set

Name: redistest.

* Type: A – Map domains to IPv4 addresses

* TTL (s): 300 (5 min)

* Value: Example: 192.168.10.10

Tag: It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

To add a tag, enter a tag key and a tag value below.

10 tags available for addition.

Description: 0/255

For details about how to configure parameters, see **Adding an A Record Set**.

Step 3 Click **OK**.

Step 4 Switch back to the **Record Sets** page.

Step 5 View the created record set in the record set list. If the status of the record set is **Normal**, the record set is added successfully.

----End

3.4.6 Configuring a Public Domain Name

This section describes how to configure and resolve public domain names.

Procedure

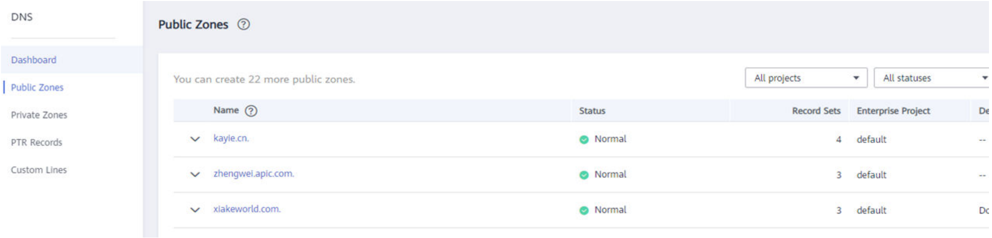
If your domain name is registered with a third-party registrar, create a public zone and add record sets to it on the DNS console.

Step 1 Log in to the GeminiDB console.

Step 2 Click **Service List** and choose **Network** > **Domain Name Service**.

Step 3 In the navigation pane on the left, choose **Public Zones**.

Figure 3-40 Public zones



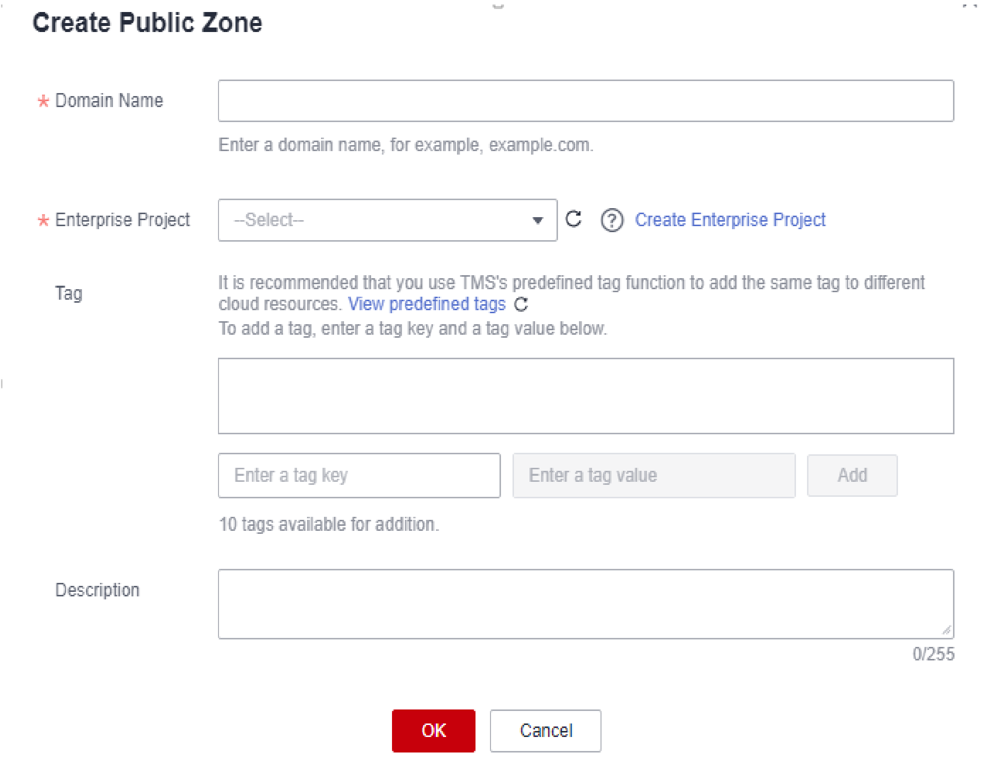
You can create 22 more public zones.

	Name	Status	Record Sets	Enterprise Project	De
∨	kayie.cn.	Normal	4	default	--
∨	zhengwei.apic.com.	Normal	3	default	--
∨	xiakeworld.com.	Normal	3	default	Dc

Step 4 In the upper right corner of the page, click **Create Public Zone**.

Step 5 Set the required parameters.

Figure 3-41 Creating a public zone



* Domain Name

Enter a domain name, for example, example.com.

* Enterprise Project --Select--

Tag

It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

To add a tag, enter a tag key and a tag value below.

Enter a tag key | Enter a tag value

10 tags available for addition.

Description

0/255

Table 3-16 Public zone parameters

Parameter	Description	Example
Domain Name	The domain name registered with the domain name registrar. The domain name can include two levels in addition to the top-level domain, for example: ● abc.example.com, the subdomain name of example.com ● abc.example.com.cn, the subdomain name of example.com.cn For details about the domain name format, see **Domain Name Formats and Structure**.	example.com
Enterprise Project	Enterprise project associated with the public domain name. You can manage public domain names by enterprise project. **NOTE** This parameter is available and mandatory only when **Account Type** is set to **Enterprise Account**. Configuration principles: ● If you do not manage domain names by enterprise project, select the **default** enterprise project. ● If you manage domain names by enterprise project, select an existing enterprise project.	default

Parameter	Description	Example
Tag	(Optional) Identifier of a resource. Each tag contains a key and a value. You can add a maximum of 10 tags to a domain name. The key and value naming rules are as follows: **Key**: ● Cannot be left blank. ● Must be unique for each resource. ● Consists of a maximum of 36 characters. ● Cannot start or end with a space or contain special characters =*<>\,|/ **Value**: ● Cannot be left blank. ● Consists of a maximum of 43 characters. ● Cannot start or end with a space or contain special characters =*<>\,|/	example_key1 example_value1
Description	(Optional) Description of the zone, which cannot exceed 255 characters.	This is a zone example.

Step 6 Click **OK**.

After the domain name is created, you can view it in the domain name list on the **Public Zones** page.

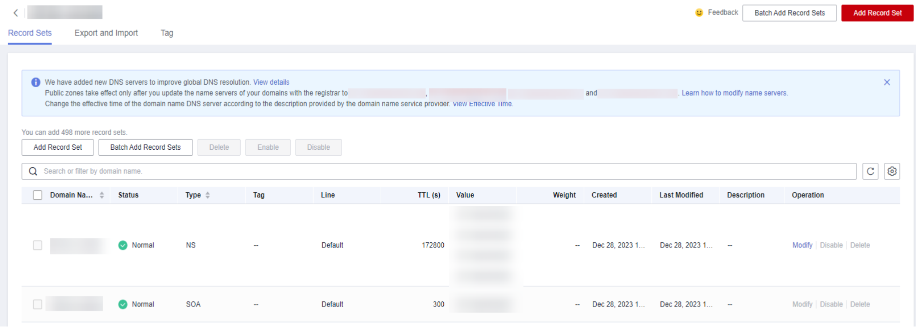
----End

Adding a Record Set for a Domain Name

After creating a public domain name, configure a record set for it so that you can access instances using the domain name.

Step 1 Click the name of the public domain name you created. On the displayed page, click **Add Record Set** in the upper right corner.

Figure 3-42 Adding a record set



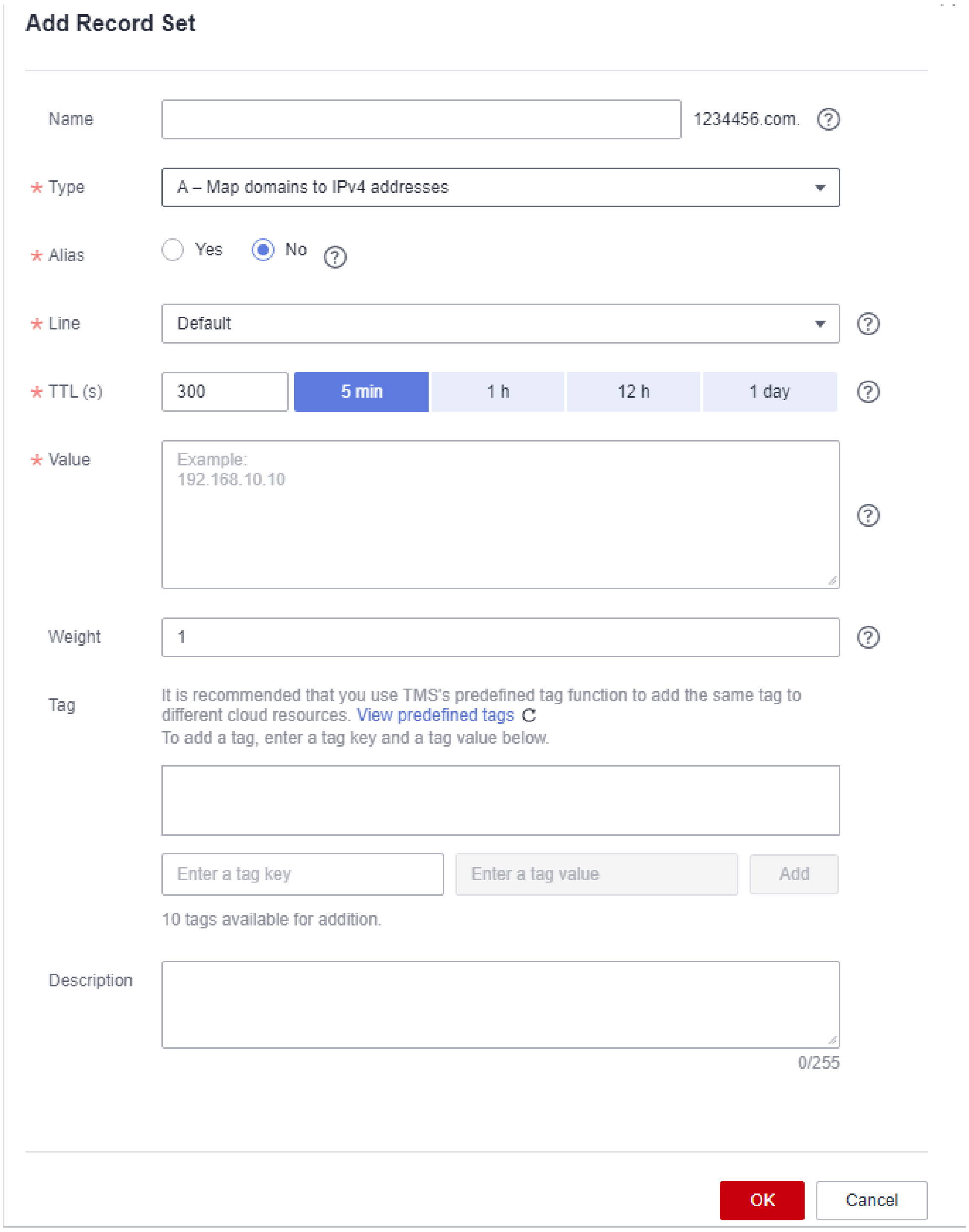
We have added new DNS servers to improve global DNS resolution. View details
Public zones take effect only after you update the name servers of your domains with the registrar to [redacted], [redacted] [redacted] and [redacted]. Learn how to modify name servers.
Change the effective time of the domain name DNS server according to the description provided by the domain name service provider. View Effective Time.

You can add 498 more record sets.

□	Domain Na...	Status	Type	Tag	Line	TTL (s)	Value	Weight	Created	Last Modified	Description	Operation
□	[redacted]	Normal	NS	--	Default	172800	[redacted]	--	Dec 28, 2023 1...	Dec 28, 2023 1...	--	Modify | Disable | Delete
□	[redacted]	Normal	SOA	--	Default	300	[redacted]	--	Dec 28, 2023 1...	Dec 28, 2023 1...	--	Modify | Disable | Delete

Step 2 In the displayed **Add Record Set** dialog box, configure the required parameters.

Figure 3-43 Adding a record set



Add Record Set

Name: 1234456.com.

* Type: A – Map domains to IPv4 addresses

* Alias: ○ Yes ● No

* Line: Default

* TTL (s): 300 | 5 min | 1 h | 12 h | 1 day

* Value: Example: 192.168.10.10

Weight: 1

Tag: It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags
To add a tag, enter a tag key and a tag value below.

Enter a tag key | Enter a tag value | Add

10 tags available for addition.

Description: 0/255

For details about how to configure parameters, see **Adding an A Record Set**.

Step 3 Click **OK**.

Step 4 Switch back to the **Record Sets** page.

Step 5 View the created record set in the record set list. If the status of the record set is **Normal**, the record set is added successfully.

----End

3.4.7 Configuring Security Group Rules for Nodes

A security group is a collection of access control rules for ECSs and GeminiDB Redis instances that have the same security protection requirements and are mutually trusted in a VPC.

To ensure database security and reliability, configure security group rules to allow specific IP addresses and ports to access the GeminiDB Redis instances.

This section describes how to configure security group rules for a GeminiDB Redis instance that is connected through a private or a public network.

Precautions

- Each account can create up to 500 security group rules by default.
- Too many security group rules will increase the first packet latency, so a maximum of 50 rules for each security group is recommended.
- One security group can be associated with only one GeminiDB Redis instance.
- For details about how to configure security group rules, see **Table 3-17**.

Table 3-17 Parameter description

Scenario	Description
Connecting to an instance over a private network	Configure security group rules as follows: ● If a GeminiDB Redis instance and the ECS used for accessing the instance are in the same security group, they can communicate with each other by default. No security group rules need to be configured. ● If the instance and the ECS are not in the same security group, configure security group rules, respectively. – Configure inbound rules for the security group associated with the GeminiDB Redis instance. For details, see **Procedure**. – There is no need to configure security rules for the ECS because the default security group rule of the ECS allows all outbound data packets. If not all outbound traffic is allowed in the security group, configure an outbound rule for the ECS.
Connecting to an instance over a public network	If you connect to a GeminiDB Redis instance through a public network, configure inbound rules for the security group associated with the GeminiDB Redis instance. For details, see **Procedure**.

Procedure

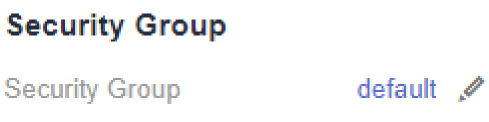
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance that you want to configure security group rules for and click its name.

Step 4 Configure security group rules.

Figure 3-44 Security group



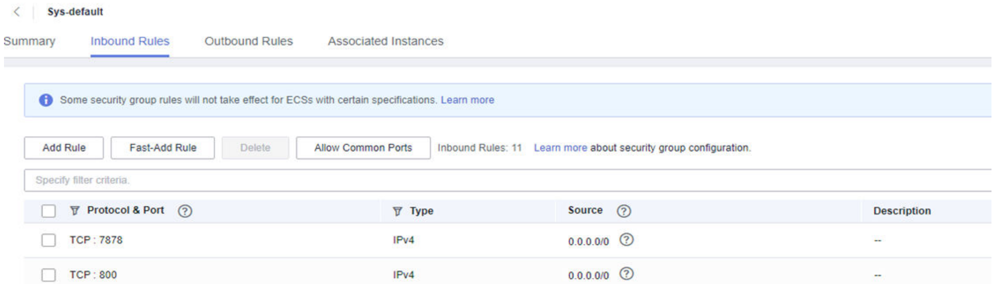
Security Group

Security Group default

Step 5 Add Inbound Rule

1. Click the **Inbound Rules** tab.

Figure 3-45 Inbound rules



Sys-default

Summary Inbound Rules Outbound Rules Associated Instances

Some security group rules will not take effect for ECSs with certain specifications. Learn more

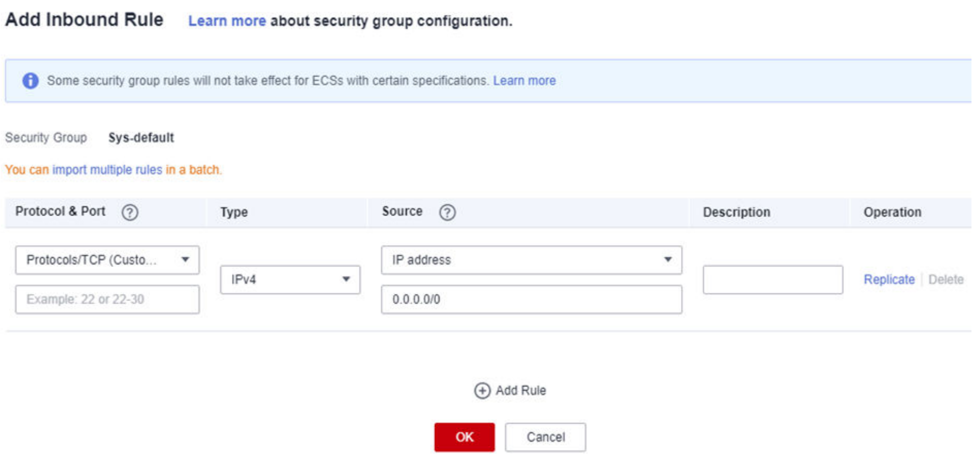
Add Rule Fast-Add Rule Delete Allow Common Ports Inbound Rules: 11 Learn more about security group configuration.

Specify filter criteria.

Protocol & Port	Type	Source	Description
TCP : 7878	IPv4	0.0.0.0/0	--
TCP : 800	IPv4	0.0.0.0/0	--

2. Click **Add Rule**. The **Add Inbound Rule** dialog box is displayed.

Figure 3-46 Adding a rule



Add Inbound Rule Learn more about security group configuration.

Some security group rules will not take effect for ECSs with certain specifications. Learn more

Security Group Sys-default

You can import multiple rules in a batch.

Protocol & Port	Type	Source	Description	Operation
Protocols/TCP (Custo... Example: 22 or 22-30	IPv4	IP address 0.0.0.0/0		Replicate | Delete

Add Rule

OK Cancel

3. Add a security group rule as prompted.

Table 3-18 Inbound rule settings

Parameter	Description	Example Value
Protocol & Port	– The network protocol required for access. Available options: **All**, **TCP**, **UDP**, **ICMP**, or **GRE** – **Port**: The port or port range that allows the access to the ECS. Range: 1 to 65535 Common ports are listed in .	TCP
Type	IP address type. This parameter is available after IPv6 is enabled. – IPv4 – IPv6	IPv4
Source	The IP address, IP address group, or security group that the rule applies to, which allows access from IP addresses or instances in another security group. Examples: – IPv4 single IP address: 192.168.10.10/32 – Subnet: 192.168.1.0/24 – All IP addresses: 0.0.0.0/0 – sg-abc (security group) For more information about IP address groups, see .	0.0.0.0/0
Description	(Optional) Provides supplementary information about the security group rule. The description can contain up to 255 characters and cannot contain angle brackets (<>).	-

Step 6 Click **OK**.

----End

3.4.8 Binding and Unbinding an EIP

Scenarios

After you create a GeminiDB Redis instance, you can bind an EIP to it to allow external access. If later you want to prohibit external access, you can also unbind the EIP from the DB instance.

Precautions

- To change the EIP that has been bound to a node, unbind it from the node first.
- You need to estimate the bandwidth required by services and purchase an EIP with sufficient bandwidth resources. **Client access exceptions caused by poor public network performance will not be included in the SLA.**

Binding an EIP

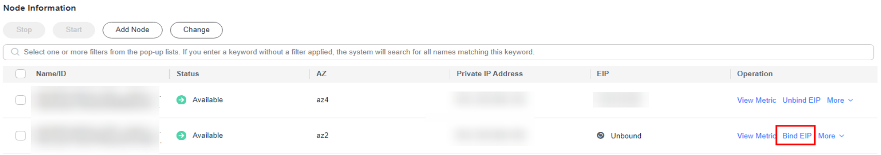
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance that you want to bind an EIP to and click its name.

Step 4 In the **Node Information** area, locate the target node and click **Bind EIP** in the **Operation** column.

Figure 3-47 Binding an EIP



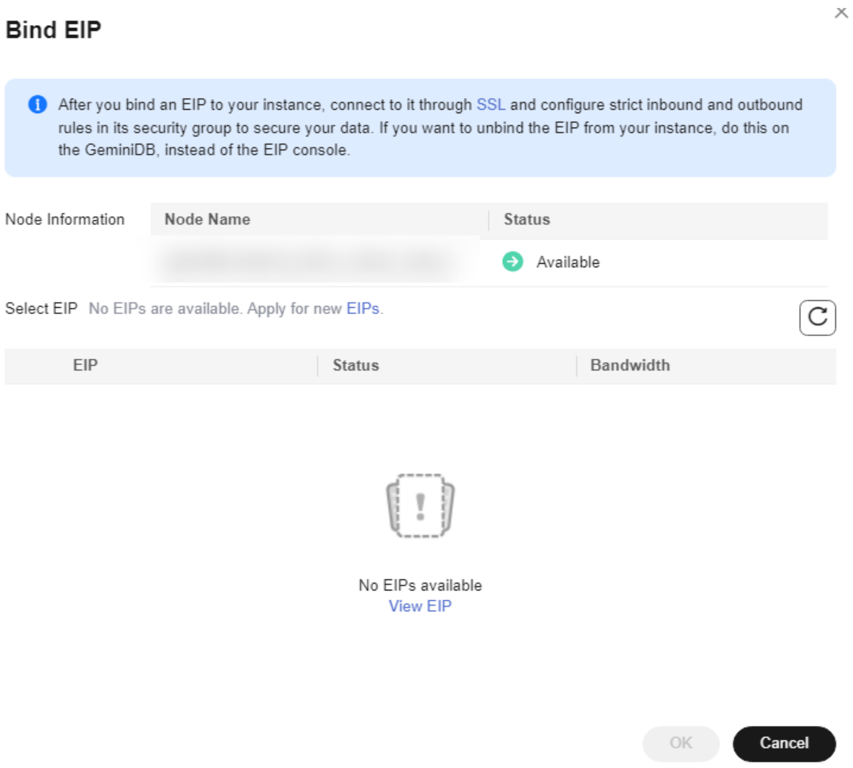
Node Information

Select one or more filters from the pop-up lists. If you enter a keyword without a filter applied, the system will search for all names matching this keyword.

□	Name/ID	Status	AZ	Private IP Address	EIP	Operation
□	[redacted]	Available	az4	[redacted]	[redacted]	View Metric Unbind EIP More
□	[redacted]	Available	az2	[redacted]	Unbound	View Metric Bind EIP More

Step 5 In the displayed dialog box, view all available EIPs, select the required EIP, and click **OK**. If no available EIPs are displayed, click **View EIP** and create an EIP.

Figure 3-48 Selecting an EIP



Bind EIP

After you bind an EIP to your instance, connect to it through SSL and configure strict inbound and outbound rules in its security group to secure your data. If you want to unbind the EIP from your instance, do this on the GeminiDB, instead of the EIP console.

Node Information

Node Name	Status
[redacted]	Available

Select EIP No EIPs are available. Apply for new EIPs.

EIP	Status	Bandwidth

No EIPs available

View EIP

Step 6 In the **EIP** column, view the EIP that is successfully bound.

To unbind the EIP from the DB instance, see **Unbinding an EIP**.

----End

Unbinding an EIP

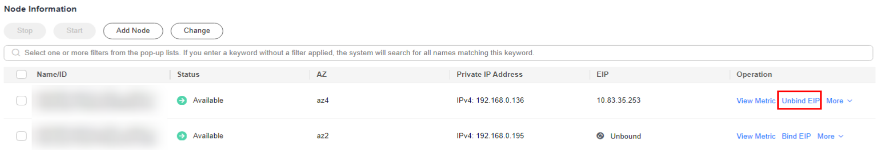
Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance that you want to unbind an EIP from.

Step 4 On the **Basic Information** page, in the **Node Information** area, locate the target node and click **Unbind EIP** in the **Operation** column.

Figure 3-49 Binding an EIP



Node Information

Stop
Start
Add Node
Change
Select one or more filters from the pop-up lists. If you enter a keyword without a filter applied, the system will search for all names matching this keyword.

□	Name/ID	Status	AZ	Private IP Address	EIP	Operation
□	[redacted]	Available	az4	IPv4: 192.168.0.136	10.83.35.253	View Metric Unbind EIP More
□	[redacted]	Available	az2	IPv4: 192.168.0.195	Unbound	View Metric Bind EIP More

Step 5 In the displayed dialog box, click **Yes**.

To bind an EIP to the DB instance again, see **Binding an EIP**.

----End

3.4.9 Viewing the IP Address and Port Number

This section describes how to query the IP address and port number of an instance on the management console.

Viewing the Load Balancer IP Address and Port

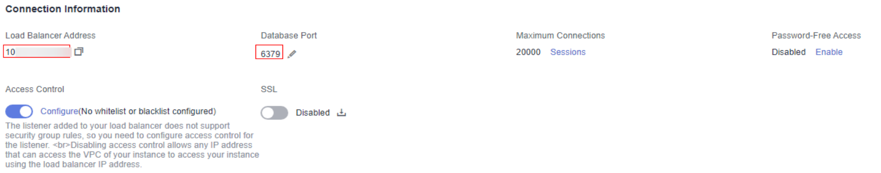
Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose IP address and port you want to view and click its name.

Step 4 In the **Connection Information** area, view the load balancer IP address and corresponding port.

Figure 3-50 Viewing the load balancer IP address and port



Connection Information

Load Balancer Address: 10 [redacted]

Database Port: 6379

Maximum Connections: 20000 Sessions

Password-Free Access: Disabled Enable

Access Control: Configure(No whitelist or blacklist configured)

The listener added to your load balancer does not support security group rules, so you need to configure access control for the listener.
Disabling access control allows any IP address that can access the VPC of your instance to access your instance using the load balancer IP address.

SSL: Disabled

----End

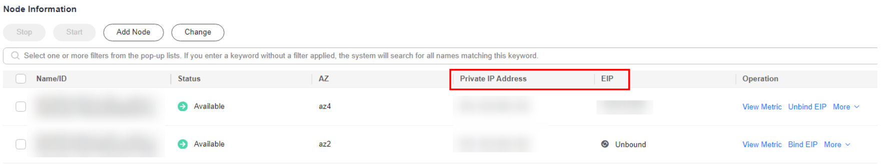
Viewing the Private IP Address or EIP

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose node IP addresses you want to view and click its name.

Figure 3-51 Obtaining IP addresses


Node Information
Stop
Start
Add Node
Change
Select one or more filters from the pop-up lists. If you enter a keyword without a filter applied, the system will search for all names matching this keyword.

Name/ID	Status	AZ	Private IP Address	EIP	Operation
[redacted]	Available	az4	[redacted]	[redacted]	View Metric Unbind EIP More
[redacted]	Available	az2	[redacted]	Unbound	View Metric Bind EIP More

----End

Viewing the Port for Accessing Each Instance Node

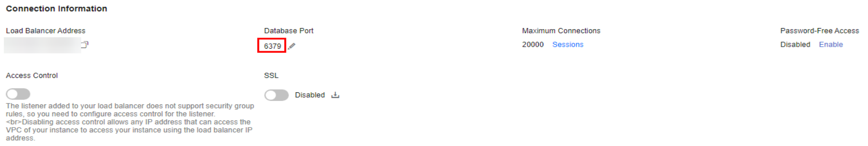
Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose node access ports you to want view and click its name.

In the **Connection Information** area on the **Basic Information** page, view the port of each instance node.

Figure 3-52 Obtaining the port number


Connection Information

Load Balancer Address: [redacted]

Database Port: 6379

Maximum Connections: 20000 Sessions

Password-Free Access: Disabled Enable

Access Control

The listener added to your load balancer does not support security group rules, so you need to configure access control for the listener.
Disabling access control allows any IP address that can access the VPC of your instance to access your instance using the load balancer IP address.

SSL: Disabled

----End

3.4.10 Configuring an SSL Connection

Secure Socket Layer (SSL) is an encryption-based Internet security protocol for establishing an encrypted link between a server and a client. It provides privacy, authentication, and integrity to Internet communications.

- Authenticates users and servers, ensuring that data is sent to the correct clients and servers.
- Encrypts data to prevent it from being intercepted during transfer.
- Ensures data integrity during transmission.

After SSL is enabled, you can establish an encrypted connection between your client and the instance you want to access to improve data security.

Precautions

- After you enable or disable SSL, the established connection is interrupted. Restart the instance to apply the change.
- Enabling SSL will prolong network connection response time and increase CPU usage. So, evaluate impacts on service performance before enabling SSL.
- The SSL function provided by GeminiDB Redis supports only TLS 1.3 or later.

Enabling SSL

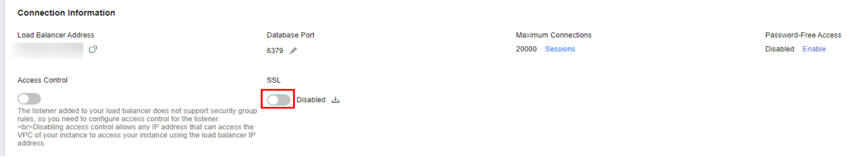
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the **Connection Information** area, click to enable SSL.

Figure 3-53 Enabling SSL



Connection Information

Load Balancer Address

Database Port: 6379

Maximum Connections: 20000 Sessions

Password-Free Access: Disabled Enable

Access Control

The listener added to your load balancer does not support security group rules, so you need to configure access control for the listener.
Disabling access control allows any IP address that can access the VPC of your instance to access your instance using the load balancer IP address.

SSL: Disabled

After SSL is enabled, you can connect to the instance through SSL connections. For details, see **Connecting to a instance Using SSL**.

----End

Disabling SSL

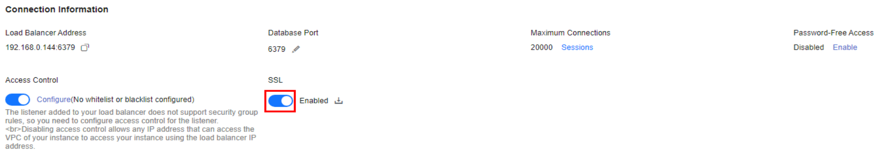
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the **Connection Information** area, click to disable SSL.

Figure 3-54 Disabling SSL



Connection Information

Load Balancer Address: 192.168.0.144:6379

Database Port: 6379

Maximum Connections: 20000 Sessions

Password-Free Access: Disabled Enable

Access Control

Configure(No whitelist or blacklist configured)

The listener added to your load balancer does not support security group rules, so you need to configure access control for the listener.
Disabling access control allows any IP address that can access the VPC of your instance to access your instance using the load balancer IP address.

SSL: Enabled

After SSL is disabled, you can connect to the GeminiDB Redis instance over a non-SSL connection. For details, see **Procedure**.

----End

3.4.11 Connecting to a instance Using SSL

GeminiDB Redis allows you to connect to a GeminiDB Redis instance through Redis-cli in SSL mode for data encryption and higher security. This section describes how to connect to a GeminiDB Redis instance using SSL.

Precautions

- The instances must be in the same VPC subnet as the ECS.
- The ECS must be in a security group that has access to the instances. For details, see **Configuring Security Group Rules for Nodes**.
- After the SSL connection is enabled, download the SSL certificate for your applications to access to the GeminiDB Redis instance.
- If the SSL connection is used, ensure that the Redis client, for example, Redis-cli 6.x, supports SSL.

Prerequisites

An ECS has been created. The following uses a Linux ECS as an example. For details, see **Purchasing an ECS** in *Getting Started with Elastic Cloud Server*.

Procedure

Step 1 Upload the SSL certificate to the ECS.

Step 2 Check the OpenSSL version supported by the ECS OS.

```
openssl version
```

NOTE

- The SSL function provided by GeminiDB Redis supports only TLS 1.3 or later.
- The OpenSSL version in the ECS OS must be 1.1.1 or later so that redis-cli can support TLS 1.3 or later.

- If the OS version is earlier than 1.1.1, perform the following steps to install OpenSSL:

  ```
  wget https://www.openssl.org/source/openssl-1.1.1m.tar.gz
  tar -zxvf openssl-1.1.1m.tar.gz
  cd openssl-1.1.1m/
  ./config --prefix=/usr/local/openssl-1.1.1m_install_dir
  make
  make install
  ```

  After OpenSSL is installed, go to **Step 3**.
- If the OS is 1.1.1 or later, go to **Step 3**.

Step 3 Decompress the client package.

```
tar -xzf redis-6.2.6.tar.gz
```

Step 4 Connect to the instance in the **src** directory.

- If the required OpenSSL version has been installed by performing **Step 2** and the version is earlier than 1.1.1, you can connect to the DB instance using the following method:

  ```
  cd redis-6.2.6
  make BUILD_TLS=yes OPENSSL_PREFIX=/usr/local/openssl-1.1.1m_install_dir
  cd src
  LD_PRELOAD=/usr/local/openssl-1.1.1m_install_dir/lib/libssl.so.1.1:/usr/local/
  ```

```
openssl-1.1.1m_install_dir/lib/libcrypto.so.1.1 ./redis-cli -h <DB_HOST> -p <DB_PORT> -a
<DB_PWD> --tls --cacert <CACERT_PATH>
```

For example:

```
LD_PRELOAD=/usr/local/openssl-1.1.1m_install_dir/lib/libssl.so.1.1:/usr/local/
openssl-1.1.1m_install_dir/lib/libcrypto.so.1.1 ./redis-cli -h 192.168.0.208 -p 8635 -a <DB_PWD> --
tls --cacert ./cacert.crt
```

- If the OpenSSL version in the ECS OS is 1.1.1 or later, you can connect to the DB instance using the following method:

```
cd redis-6.2.6
make BUILD_TLS=yes
cd src
./redis-cli -h <DB_HOST> -p <DB_PORT> -a <DB_PWD> --tls --cacert <CACERT_PATH>
```

For example:

```
./redis-cli -h 192.168.0.208 -p 8635 -a <DB_PWD> --tls --cacert ./cacert.crt
```

Table 3-19 Parameter Description

Parameter	Description
<DB_HOST>	The private IP address of the instance to be accessed. To obtain this IP address, go to the **Instances** page, locate the instance, and click its name. The IP address can be found in the **Private IP Address** field under **Node Information** on the **Basic Information** page. If the instance you has multiple nodes, select the private IP address of any node.
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the port number, perform the following steps: On the **Instances** page, locate the instance and click its name. On the **Basic Information** page, in the **Network Information** area, view the database port.
<DB_PWD>	Specifies the administrator password set when you a GeminiDB Redis instance.
<CACERT_PATH>	Path of the SSL certificate.

Step 5 If information similar to the following is displayed, the connection was successful.

```
IP:port>
```

----End

3.4.12 Changing a Node Security Group

Scenarios

You can change security groups of GeminiDB Redis instances.

Precautions

- If you are adding nodes to a DB instance, the security group of the instance cannot be changed.

Procedure

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose security group you want to change and click its name.

Step 4 In the **Security Group** area, click ✎ to select a security group.

- To submit the change, click ✓. This process takes about 1 to 3 minutes.
- To cancel the change, click ✗.

Step 5 View the modification result.

----End

3.4.13 Enabling or Disabling Private Network Access for a Load Balancer

Scenarios

GeminiDB Redis allows you to enable or disable private network access for a load balancer.

Precautions

- A load balancer address does not support security groups. After instance creation is complete, configure IP address access control. If no whitelist is configured, all IP addresses that can communicate with the VPC can access the instance.

Enabling a Blacklist/Whitelist for a Load Balancer IP Address

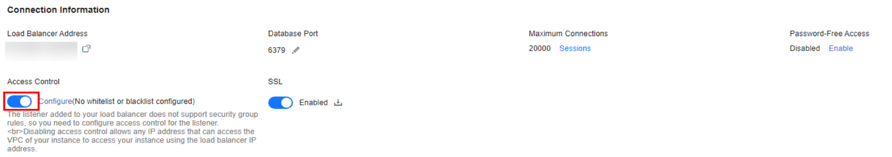
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the **Connection Information** area, click next to **Access Control**.

Figure 3-55 Enabling private network access for a load balancer



Connection Information

Load Balancer Address
[redacted]

Database Port
6379

Maximum Connections
20000 Sessions

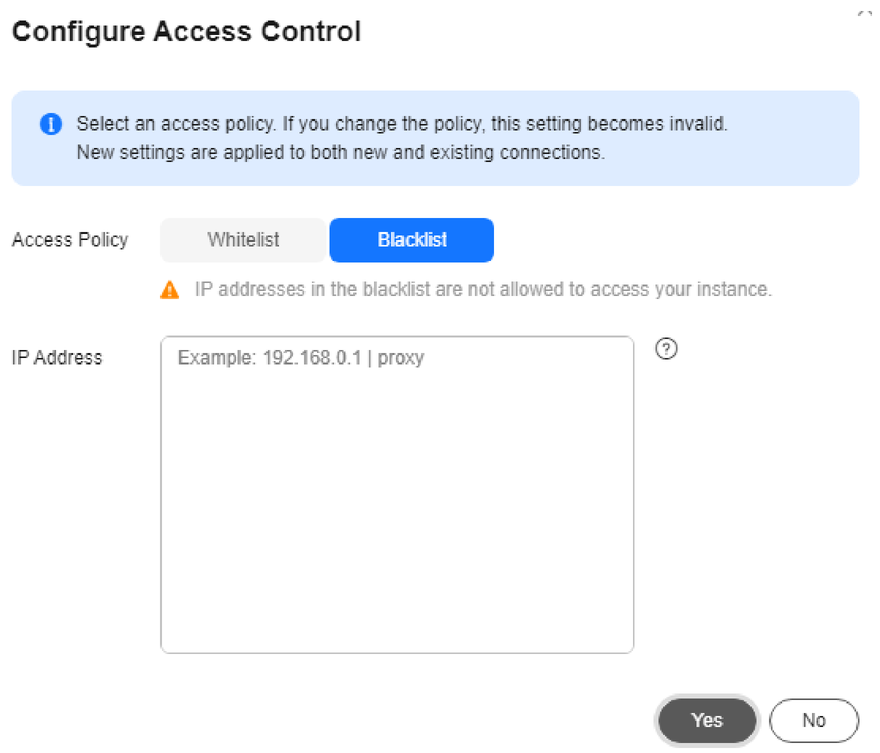
Password-Free Access
Disabled Enable

Access Control
Configure(No whitelist or blacklist configured)
The listener added to your load balancer does not support security group rules, so you need to configure access control for the listener.
Disabling access control allows any IP address that can access the VPC of your instance to access your instance using the load balancer IP address.

SSL
Enabled

Step 5 Select **Blacklist** or **Whitelist** and specify IP addresses in that list.

Figure 3-56 Configuring access control



Configure Access Control

Select an access policy. If you change the policy, this setting becomes invalid. New settings are applied to both new and existing connections.

Access Policy: Whitelist | Blacklist

IP addresses in the blacklist are not allowed to access your instance.

IP Address: Example: 192.168.0.1 | proxy

- Blacklist: The blocklist and allowlist cannot be configured at the same time. If you switch between lists, your previously entered settings will be lost. IP addresses in the blacklist cannot be accessed. Exercise caution when performing this operation.
- Whitelist: The blocklist and allowlist cannot be configured at the same time. If you switch between lists, your previously entered settings will be lost. Only IP addresses in the whitelist are allowed to access the system. Exercise caution when performing this operation.

----End

Disabling Private Network Access for a Load Balancer

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the **Connection Information** area, click next to **Access Control**. In the displayed dialog box, click **Yes**.

Step 5 Check the load balancer address cannot take effect.

----End

3.5 Instance Statuses

The status of a DB instance indicates the health of the instance. You can view the DB instance statuses on the management console.

Table 3-20 DB instance statuses

Status	Description
Available	The instance is available.
Abnormal	The instance is abnormal.
Creating	The instance is being created.
Creation failed	The instance failed to be created.
Restarting	The instance is being restarted.
Resetting password	The administrator password is being reset.
Adding node	Nodes are being added to an instance.
Deleting node	Nodes are being deleted from an instance.
Scaling up	The storage space of an instance is being scaled up.
Changing instance class	The vCPUs and memory of an instance are being changed.
Changing to yearly/monthly	The billing mode is being changed from pay-per-use to yearly/monthly.
Changing to pay-per-use	The billing mode is being changed from yearly/monthly to pay-per-use.
Uploading backup	The backup file is being uploaded.
Backing up	A database backup is being created.
Checking restoration	The backup of the instance is being restored to a new instance.
Configuring SSL	SSL is being enabled or disabled.
Checking changes	The yearly/monthly instance is pending check when its billing mode is changed.

3.6 Instance Lifecycle Management

3.6.1 Restarting an Instance

Scenarios

You may need to restart an instance for routine maintenance.

Precautions

- Only instances in states **Available**, **Abnormal**, or **Checking restoration** can be restarted.
- After you restart an instance, all nodes in the instance are also restarted.
- Restarting an instance will interrupt services. Wait until off-peak hours and ensure that your application can re-connect.
- If you enable operation protection, two-factor authentication is required for sensitive operations to secure your account and cloud products. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Procedure

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance you want to restart and in the **Operation** column choose **Restart** or **More** > **Restart**.

Alternatively, locate the instance you want to restart and click its name. On the displayed **Basic Information** page, click **Restart** in the upper right corner of the page.

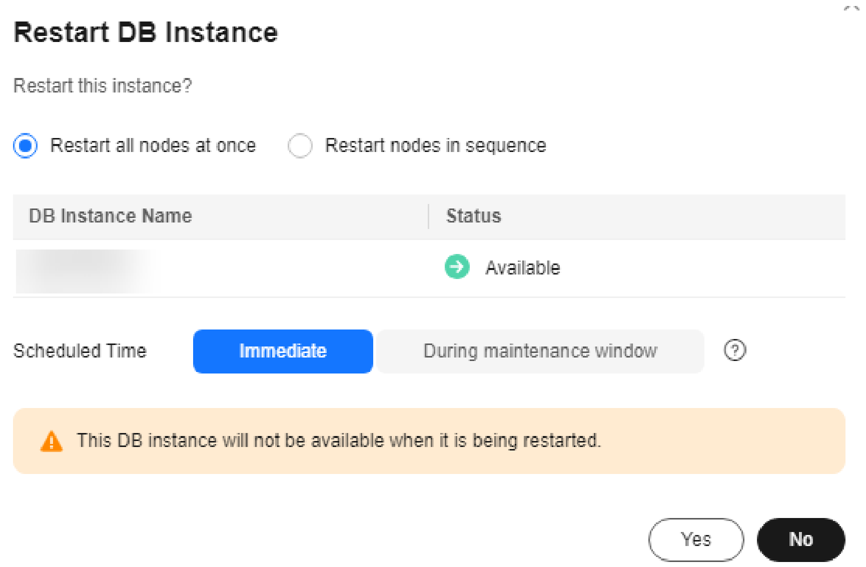
Step 4 If you have enabled operation protection, click **Start Verification** in the **Restart DB Instance** dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

Step 5 In the displayed dialog box, click **Yes** or **Immediate**.

- Instance in classic deployment mode

  For GeminiDB Redis instances in classic deployment mode, you can restart several nodes at the same time or in sequence based on service requirements.

Figure 3-57 Restarting an instance



Restart DB Instance

Restart this instance?

● Restart all nodes at once ○ Restart nodes in sequence

DB Instance Name	Status
[redacted]	Available

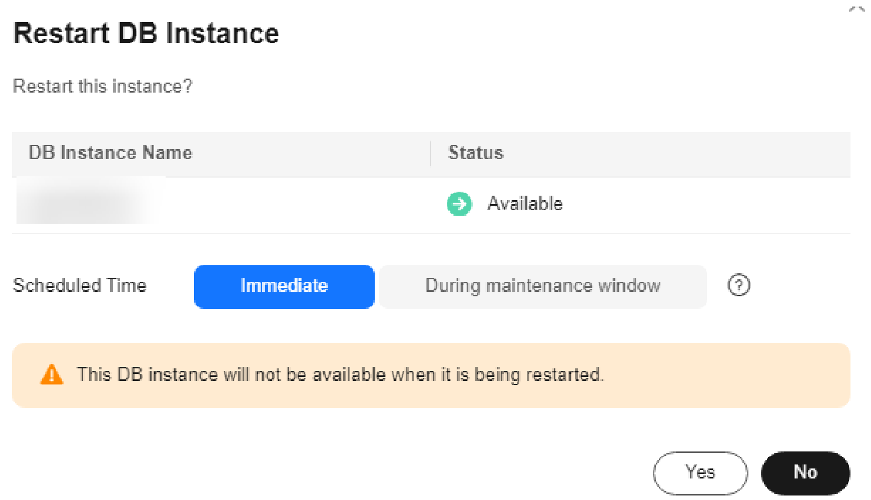
Scheduled Time: Immediate | During maintenance window

This DB instance will not be available when it is being restarted.

- Instance in cloud native deployment mode

  For GeminiDB Redis instances deployed in cloud native mode, click **Yes** or **Immediate**.

Figure 3-58 Restarting an instance



Restart DB Instance

Restart this instance?

DB Instance Name	Status
[redacted]	Available

Scheduled Time: Immediate | During maintenance window

This DB instance will not be available when it is being restarted.

----End

3.6.2 Exporting Instance Information

Scenarios

You can export information about all or selected instances to view and analyze instance information.

Precautions

To enable this function, contact customer service.

Exporting All Instance Information

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click ↗ in the upper right corner of the page. By default, information about all DB instances are exported. In the displayed dialog box, you can select the items to be exported and click **Export**.

Step 4 After the export task is complete, check an XLS file is generated locally.

----End

Exporting Information About Selected Instances

Step 1 On the **Instances** page, select the instances that you want to export or search for required instances by project, compatible API, name, ID, or tag and click ↗ in the upper right corner of the page. In the displayed dialog box, select the items to be exported and click **Export**.

Step 2 After the export task is complete, check an XLS file is generated locally.

----End

3.6.3 Deleting a Pay-per-Use Instance

Scenarios

You can choose to delete a pay-per-use instance on the **Instances** page based on service requirements. To delete a yearly/monthly DB instance, you need to unsubscribe from the order. For details, see **Unsubscribing from a Yearly/ Monthly Instance**.

Precautions

- Instances that an operation is being performed on cannot be deleted. They can be deleted only after the operations are complete.
- If a instance is deleted, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.
- After an instance is deleted, all its data and automated backups are automatically deleted as well and cannot be recovered. You are advised to create a backup before deleting an instance. For details, see **Creating a Manual Backup**.
- After you delete an instance, all of its nodes are deleted.
- A deleted instance will be retained in the recycle bin for a period of time after being released, so you can rebuild the instance and restore data from it.

Procedure

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance that you want to delete and in the **Operation** column choose **Delete** or **More** > **Delete**.

Step 4 If you have enabled operation protection, click **Start Verification** in the **Delete DB Instance** dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

> NOTE
>
> If you enable operation protection, two-factor authentication is required for sensitive operations to secure your account and cloud products. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 5 In the displayed dialog box, click **Yes**.

Deleted instances are not displayed in the instance list any longer.

----End

3.6.4 Recycling an Instance

Unsubscribed yearly/monthly instances and deleted pay-per-use instances are moved to the recycle bin and can be restored.

Precautions

- The recycling bin is enabled by default and cannot be disabled. Instances in the recycle bin are retained for 7 days by default, and this will not incur any charges.
- You can put up to 100 instances into the recycle bin. If the maximum number of instances is reached, you cannot put instances into the recycle bin any more.
- If you delete an instance of full storage, the deleted instance will not be moved to the recycle bin.

Modifying the Recycling Policy

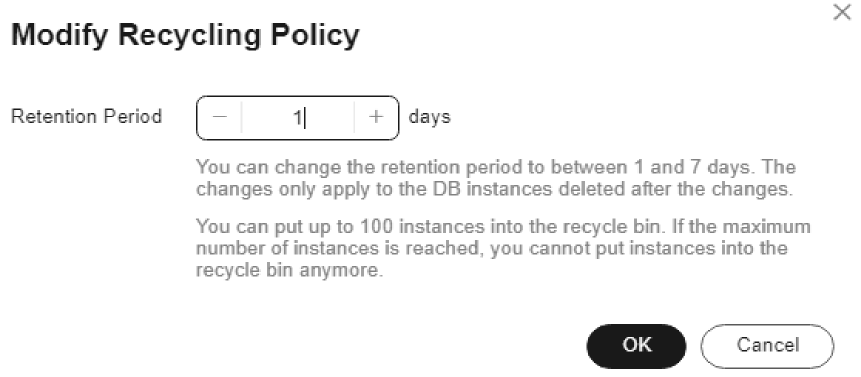
> NOTICE
>
> You can modify the retention period, and the new retention period only takes effect for the instances that are deleted after the modification.

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Recycling Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period from 1 day to 7 days. Then, click **OK**.

Figure 3-59 Modifying the recycling policy



Modify Recycling Policy

Retention Period − 1 + days

You can change the retention period to between 1 and 7 days. The changes only apply to the DB instances deleted after the changes.

You can put up to 100 instances into the recycle bin. If the maximum number of instances is reached, you cannot put instances into the recycle bin anymore.

----End

Rebuilding an Instance

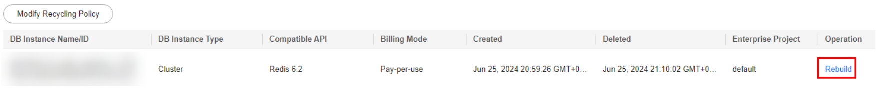
You can rebuild instances from the recycle bin within the retention period to restore data.

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Recycling Bin** page, locate the instance that you want to rebuild and click **Rebuild** in the **Operation** column.

Figure 3-60 Rebuilding an instance



DB Instance Name/ID	DB Instance Type	Compatible API	Billing Mode	Created	Deleted	Enterprise Project	Operation
[redacted]	Cluster	Redis 6.2	Pay-per-use	Jun 25, 2024 20:59:26 GMT+0...	Jun 25, 2024 21:10:02 GMT+0...	default	Rebuild

Step 4 On the displayed page, set required parameters (you are advised to set the specifications to be the same as those of the original instance) and submit the rebuilding task.

----End

3.7 Instance Changes

3.7.1 Modifying the Name of an Instance

Scenarios

This section describes how to modify the name of a GeminiDB Redis instance.

Method 1

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose name you want to modify and click ✎ to the right of the instance.

- To submit the change, click **OK**.
- To cancel the change, click **Cancel**.

NOTE

The instance name:

- Can be the same as an existing instance name.
- Can include 4 to 64 bytes and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_).

Step 4 View the results on the **Instances** page.

----End

Method 2

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance whose name you want to modify and click its name.

Step 4 In the **Instance Information** area on the **Basic Information** page, click ✎ in the **DB Instance Name** field.

- To submit the change, click ✓ .
- To cancel the change, click ✗ .

Step 5 Check the results on the **Instances** page.

----End

3.7.2 Changing the Administrator Password of a GeminiDB Redis Instance

Scenarios

For security reasons, regularly change your administrator password.

Precautions

- You can reset the administrator password only when the **instance status** is **Available**, **Backing up**, or **Scaling up**.

- If you enable operation protection, two-factor authentication is required for sensitive operations to secure your account and cloud products. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Method 1

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose administrator password you want to reset and choose **More** > **Reset Password** in the **Operation** column.

Step 4 Enter and confirm the new administrator password and click **OK**.

The password must be 8 to 32 characters in length and contain any two of uppercase letters, lowercase letters, digits, and the following special characters: ~!@#%^*-_=+?$()&

Step 5 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

----End

Method 2

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose administrator password you want to reset and click its name. The **Basic Information** page is displayed.

Step 4 In the **DB Information** area, click **Reset Password** in the **Administrator** field.

Step 5 Enter and confirm the new administrator password and click **OK**.

The password must be 8 to 32 characters in length and contain any two of uppercase letters, lowercase letters, digits, and the following special characters: ~!@#%^*-_=+?$()&

Step 6 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

----End

3.7.3 Scaling Up Storage Space

Scenarios

This section describes how to scale up storage space of an instance to suit your service requirements.

Precautions

- Storage space can only be scaled up.
- When the disk usage of a GeminiDB Redis instance exceeds 95%, the instance enters the read-only mode. You can only read or delete data from the instance, but cannot write new data into it. To keep services accessible, scale up storage space when the disk usage exceeds 80%.
- **Storage scaling does not interrupt your services. After storage scaling is complete, you do not need to restart your instance.**

Notes on Setting an Instance to Read-Only

To ensure that the instance can still be used if the storage space is about to be used up, the database is set to read-only, and data cannot be modified. If this happens, you can add more storage to restore the database to read/write status.

Table 3-21 Notes on setting an instance to read-only

Storage Space	Description
Less than 600 GB	● When the storage usage reaches 97%, the instance is set to read-only. ● When the storage usage decreases to 85%, the read-only status is automatically disabled for the instance.
Greater than or equal to 600 GB	● The remaining storage space is less than 18 GB, and the instance is set to read-only. ● If the remaining storage space is greater than or equal to 90 GB, the read-only status is automatically disabled for the instance.

Method 1

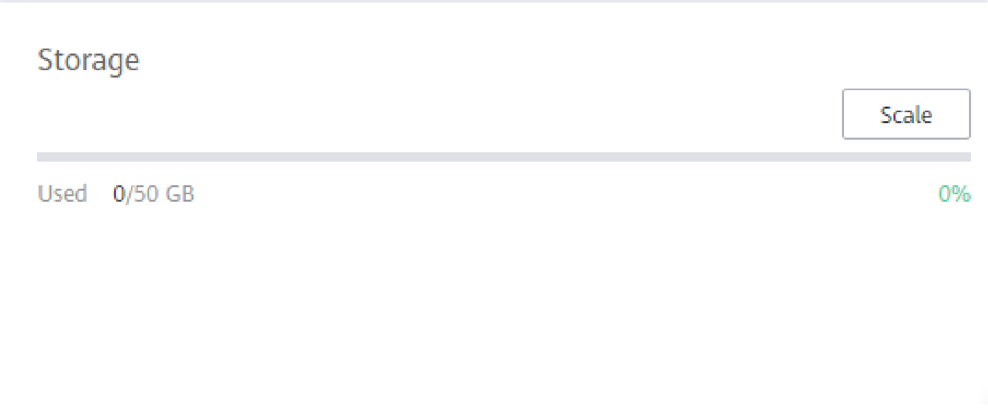
Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the **Storage Space** area on the **Basic Information** page, click **Scale**.

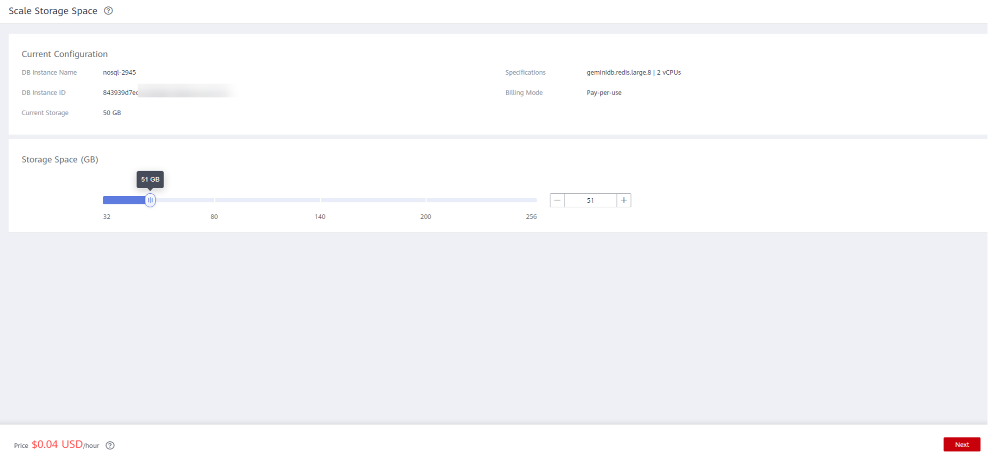
Figure 3-61 Scaling storage



Used 0/50 GB 0%

Step 5 On the displayed page, specify the new storage capacity and click **Next**.

Figure 3-62 Scaling storage



Current Configuration

DB Instance Name nosql-2945

DB Instance ID 843939d7ec[redacted]

Current Storage 50 GB

Specifications geminidb.redis.large.8 | 2 vCPUs

Billing Mode Pay-per-use

Storage Space (GB) 51

Price $0.04 USD/hour

Select at least 1 GB each time you scale up the storage, and the storage size must be an integer.

Step 6 On the displayed page, confirm the storage space.

- For yearly/monthly instances
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit** and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit**.

Step 7 Check the scaling result.

- When the scaling task is ongoing, the instance status is **Scaling up**.
- After the scaling task is complete, the instance status becomes **Available**.

- In the **Storage Space** area on the **Basic Information** page, view the new storage space.

----End

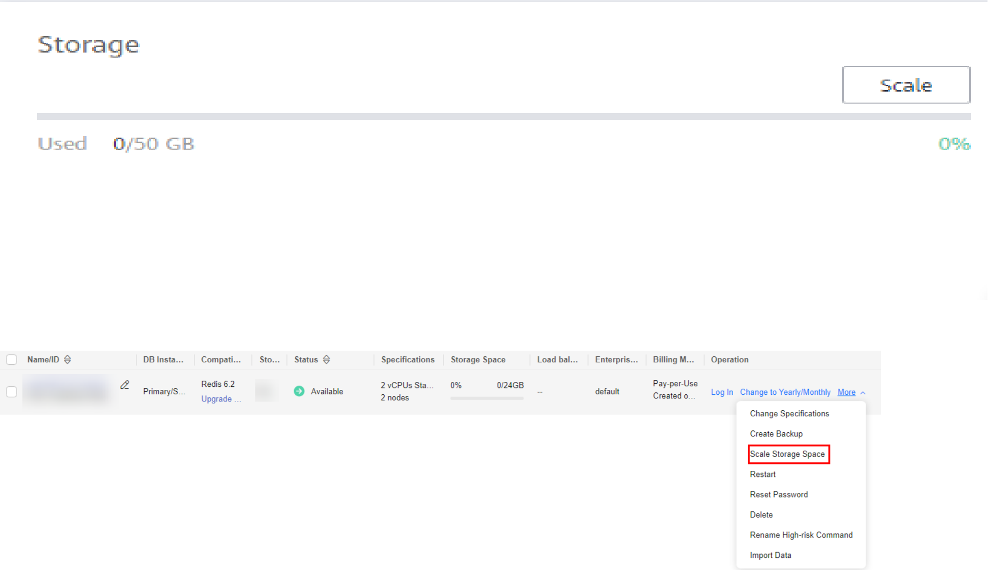
Method 2

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose storage space you want to scale and choose **More** > **Scale Storage Space** in the **Operation** column.

Figure 3-63 Scaling storage



Storage

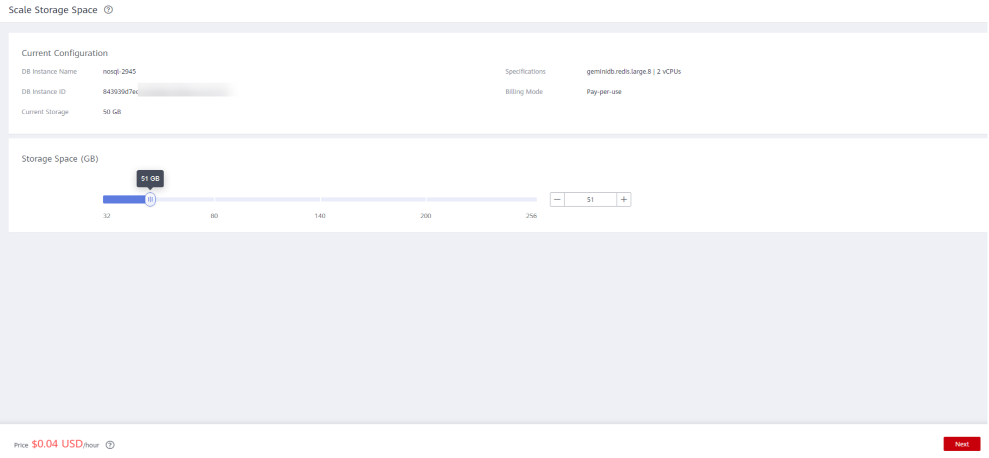
Scale

Used 0/50 GB 0%

☐	Name/ID		DB Insta...	Compati...	Sto...	Status	Specifications	Storage Space	Load bal...	Enterpris...	Billing M...	Operation
☐	[redacted]		Primary/S...	Redis 6.2 Upgrade ...		Available	2 vCPUs Sta... 2 nodes	0% 0/24GB	--	default	Pay-per-Use Created o...	Log In Change to Yearly/Monthly More

Step 4 On the displayed page, specify the new storage capacity and click **Next**.

Figure 3-64 Scaling storage



Scale Storage Space

Current Configuration

DB Instance Name nosql-2945

DB Instance ID 843939d7ec[redacted]

Current Storage 50 GB

Specifications geminidb.redis.large.8 | 2 vCPUs

Billing Mode Pay-per-use

Storage Space (GB)

Price $0.04 USD/hour

Select at least 1 GB each time you scale up the storage, and the storage size must be an integer.

Step 5 On the displayed page, confirm the storage space.

- For yearly/monthly instances
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit** and complete the payment.
- Pay-per-use
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit**.

Step 6 Check the scaling result.

- When the scaling task is ongoing, the instance status is **Scaling up**.
- After the scaling task is complete, the instance status becomes **Available**.
- In the **Storage Space** area on the **Basic Information** page, view the new storage space.

----End

3.7.4 Changing the CPU and Memory Specifications of an Instance

Scenarios

You can increase or decrease the CPU or memory of all nodes in an instance. You can change the vCPU and memory specifications of your instance to meet your service requirements. If an instance is overloaded and compute resources need to be added urgently, you are advised to add compute nodes first.

Precautions

- During online specification change, second-level intermittent disconnection occurs once when the change is performed on a single node. Therefore, the entire instance is intermittently disconnected for several times. The client must have an automatic reconnection mechanism. You are advised to perform the specification change during off-peak hours.
- For a node whose specifications are being changed, its computing tasks are handed over to other nodes. Change specifications of nodes during off-peak hours to prevent the instance from overload.
- To modify specifications of an instance based on another specification ratio, contact customer service.
- After specifications of a standard instance in cloud native deployment mode are changed, the system automatically adjusts the storage capacity to the number of shards multiplied by shard specifications (GB).

Procedure

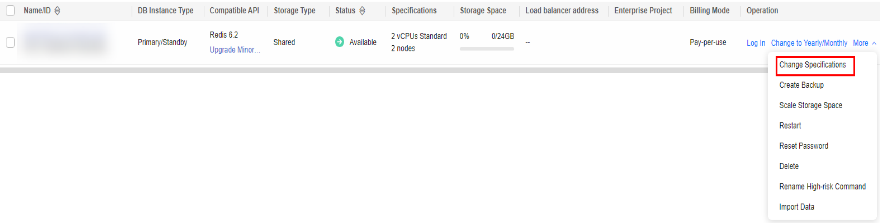
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose specifications you want to change and click **More > Change Specifications** in the **Operation** column.

- Instance in classic deployment mode

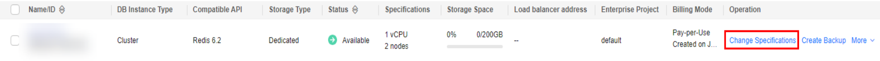
Figure 3-65 Changing specifications



Name/ID	DB Instance Type	Compatible API	Storage Type	Status	Specifications	Storage Space	Load balancer address	Enterprise Project	Billing Mode	Operation
[redacted]	Primary/Standby	Redis 6.2 Upgrade Minor...	Shared	Available	2 vCPUs Standard 2 nodes	0% 0/24GB	--		Pay-per-use	Log In Change to Yearly/Monthly More

- Instance in cloud native deployment mode

Figure 3-66 Changing specifications

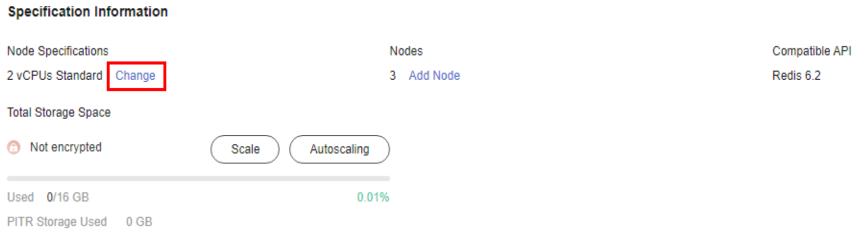


Name/ID	DB Instance Type	Compatible API	Storage Type	Status	Specifications	Storage Space	Load balancer address	Enterprise Project	Billing Mode	Operation
[redacted]	Cluster	Redis 6.2	Dedicated	Available	1 vCPU 2 nodes	0% 0/200GB	--	default	Pay-per-Use Created on J...	Change Specifications Create Backup More

In the **DB Information** area on the **Basic Information** page, click **Change** next to the **Specifications** field.

- Instance in classic deployment mode

Figure 3-67 Changing specifications



Specification Information

Node Specifications
2 vCPUs Standard Change

Nodes
3 Add Node

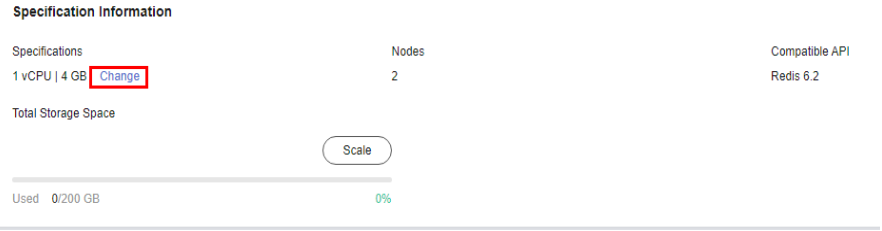
Compatible API
Redis 6.2

Total Storage Space
Not encrypted

Used 0/16 GB 0.01%
PITR Storage Used 0 GB

- Capacity-oriented instance in cloud native deployment mode

Figure 3-68 Changing specifications



Specification Information

Specifications
1 vCPU | 4 GB Change

Nodes
2

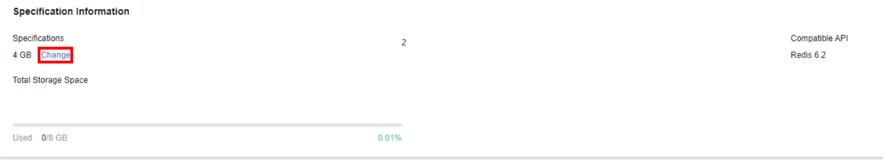
Compatible API
Redis 6.2

Total Storage Space

Used 0/200 GB 0%

- Standard instance in cloud native deployment mode

Figure 3-69 Changing specifications



Specification Information

Specifications
4 GB Change

2

Compatible API
Redis 6.2

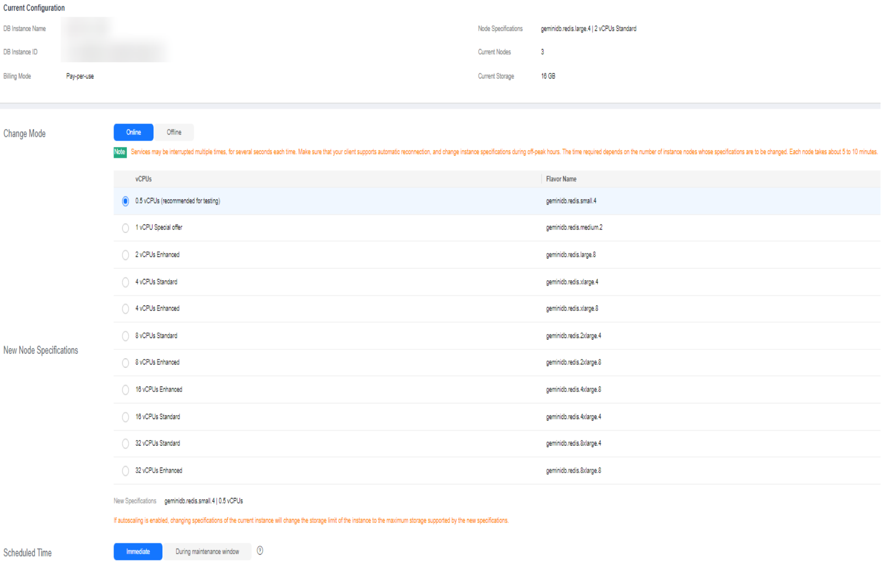
Total Storage Space

Used 0/8 GB 0.01%

Step 4 On the displayed page, select a specification change mode and required specifications, and click **Next**.

- Online change: During the change, instance nodes are upgraded in rolling mode, which has the minimum impact on services. The change duration is positively related to the number of nodes. Each node takes about 5 to 10 minutes. If there are a large number of nodes, wait patiently.
- Offline change: During offline change, all nodes are changed concurrently, which interrupts services for about 10 to 20 minutes. Exercise caution when performing this operation. For your online production services, you are advised to perform the change online.
- Instance in classic deployment mode

Figure 3-70 Changing specifications



Current Configuration

DB Instance Name	[redacted]	Node Specifications	geminidb.redis.large.4 | 2 vCPUs Standard
DB Instance ID	[redacted]	Current Nodes	3
Billing Mode	Pay-per-use	Current Storage	16 GB

Change Mode: Online | Offline

Note Services may be interrupted multiple times, for several seconds each time. Make sure that your client supports automatic reconnection, and change instance specifications during off-peak hours. The time required depends on the number of instance nodes whose specifications are to be changed. Each node takes about 5 to 10 minutes.

New Node Specifications

vCPUs	Flavor Name
● 0.5 vCPUs (recommended for testing)	geminidb.redis.small.4
○ 1 vCPU Special offer	geminidb.redis.medium.2
○ 2 vCPUs Enhanced	geminidb.redis.large.8
○ 4 vCPUs Standard	geminidb.redis.xlarge.4
○ 4 vCPUs Enhanced	geminidb.redis.xlarge.8
○ 8 vCPUs Standard	geminidb.redis.2xlarge.4
○ 8 vCPUs Enhanced	geminidb.redis.2xlarge.8
○ 16 vCPUs Enhanced	geminidb.redis.4xlarge.8
○ 16 vCPUs Standard	geminidb.redis.4xlarge.4
○ 32 vCPUs Standard	geminidb.redis.8xlarge.4
○ 32 vCPUs Enhanced	geminidb.redis.8xlarge.8

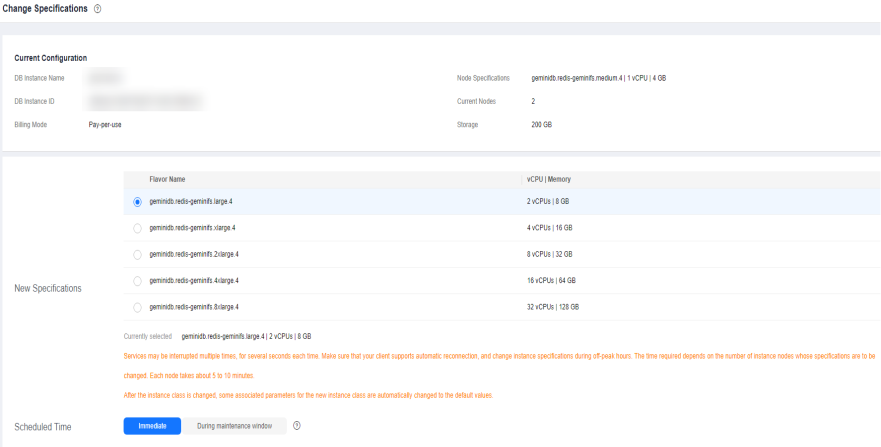
New Specifications geminidb.redis.small.4 | 0.5 vCPUs

If autoscaling is enabled, changing specifications of the current instance will change the storage limit of the instance to the maximum storage supported by the new specifications.

Scheduled Time: Immediate | During maintenance window

- Capacity-oriented instance in cloud native deployment mode

Figure 3-71 Changing specifications



Change Specifications

Current Configuration

DB Instance Name	[redacted]	Node Specifications	geminidb.redis-geminifs.medium.4 | 1 vCPU | 4 GB
DB Instance ID	[redacted]	Current Nodes	2
Billing Mode	Pay-per-use	Storage	200 GB

New Specifications

Flavor Name	vCPU | Memory
● geminidb.redis-geminifs.large.4	2 vCPUs | 8 GB
○ geminidb.redis-geminifs.xlarge.4	4 vCPUs | 16 GB
○ geminidb.redis-geminifs.2xlarge.4	8 vCPUs | 32 GB
○ geminidb.redis-geminifs.4xlarge.4	16 vCPUs | 64 GB
○ geminidb.redis-geminifs.8xlarge.4	32 vCPUs | 128 GB

Currently selected geminidb.redis-geminifs.large.4 | 2 vCPUs | 8 GB

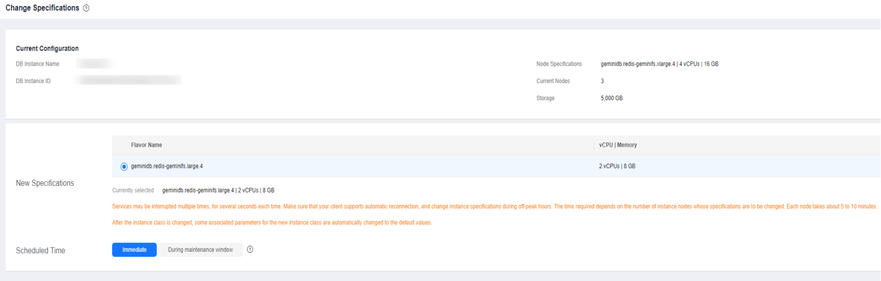
Services may be interrupted multiple times, for several seconds each time. Make sure that your client supports automatic reconnection, and change instance specifications during off-peak hours. The time required depends on the number of instance nodes whose specifications are to be changed. Each node takes about 5 to 10 minutes.

After the instance class is changed, some associated parameters for the new instance class are automatically changed to the default values.

Scheduled Time: Immediate | During maintenance window

- Standard instance in cloud native deployment mode

Figure 3-72 Changing specifications



Change Specifications

Current Configuration

DB Instance Name [redacted]

DB Instance ID [redacted]

Node Specifications geminidb.redis-geminifs.xlarge.4 | 4 vCPUs | 16 GB

Current Nodes 3

Storage 5,000 GB

New Specifications

	Flavor Name	vCPU | Memory
●	geminidb.redis-geminifs.large.4	2 vCPUs | 8 GB

Currently selected geminidb.redis-geminifs.large.4 | 2 vCPUs | 8 GB

Services may be interrupted multiple times, for several seconds each time. Make sure that your client supports automatic reconnection, and change instance specifications during off-peak hours. The time required depends on the number of instance nodes whose specifications are to be changed. Each node takes about 5 to 10 minutes.

After the instance class is changed, some associated parameters for the new instance class are automatically changed to the default values.

Scheduled Time Immediate During maintenance window

Step 5 On the displayed page, confirm the specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** .

Step 6 View the results.

Go to the **Basic Information** page and in the **DB Information** area you can see the new instance specifications.

----End

3.7.5 Adding Nodes

Scenarios

This section describes how to add nodes to an instance to suit your service requirements. You can also delete a node as required. For details, see **Deleting Nodes**.

Precautions

- Adding nodes will trigger fast load balancing, which may cause a request timeout for a few seconds. Enable automatic retry for services.
- You can add nodes only when the instance status is **Available** or **Checking restoration**.
- An instance cannot be deleted when one or more nodes are being added.
- If the storage is insufficient, adding nodes is not supported. Expand the storage first. For details about the storage supported by instances of different specifications, see **Instance Specifications**.
- Currently, only cluster instances support node addition.

Method 1

Step 1 **Log in to the GeminiDB console.**

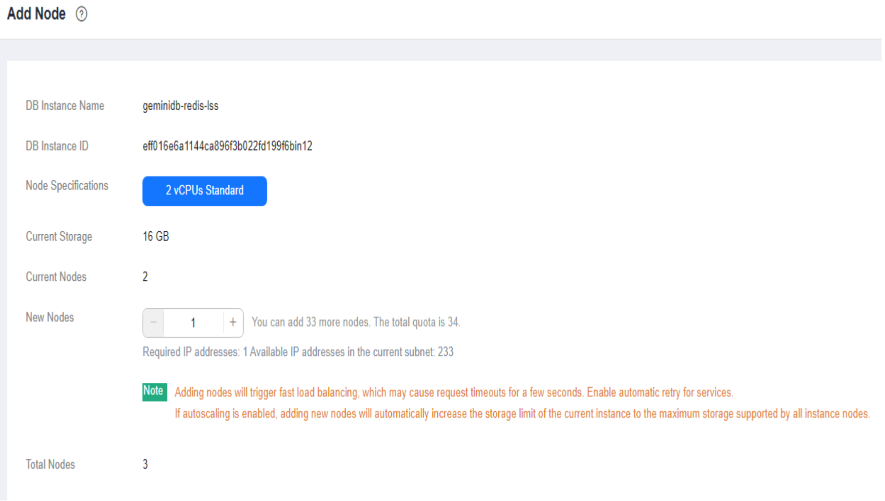
Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance you want to add nodes for and click its name.

Step 4 In the navigation pane, choose **Node Management**.

Step 5 Click **Add Node**, on the displayed page, specify the number of nodes to be added and view the storage of the instance.

- If the storage is sufficient, click **Next** and go to **12**.
- If the storage is insufficient, click **Next** and go to **8**.



Add Node

DB Instance Name geminidb-redis-lss

DB Instance ID eff016e6a1144ca896f3b022fd199f6bin12

Node Specifications 2 vCPUs Standard

Current Storage 16 GB

Current Nodes 2

New Nodes − 1 + You can add 33 more nodes. The total quota is 34.

Required IP addresses: 1 Available IP addresses in the current subnet: 233

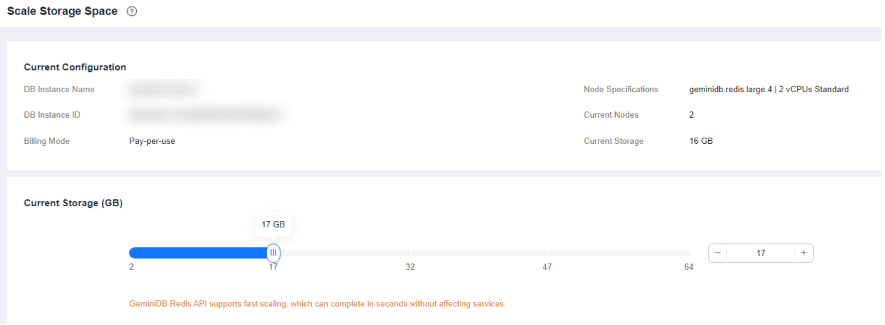
Note Adding nodes will trigger fast load balancing, which may cause request timeouts for a few seconds. Enable automatic retry for services.
If autoscaling is enabled, adding new nodes will automatically increase the storage limit of the current instance to the maximum storage supported by all instance nodes.

Total Nodes 3

New nodes are of the same specifications as existing nodes. Once a new node is added, its specifications cannot be changed.

Step 6 On the **Scale Storage Space** page, select your target storage capacity and click **Next**.

Figure 3-73 Storage change



Scale Storage Space

Current Configuration

DB Instance Name [redacted]

DB Instance ID [redacted]

Billing Mode Pay-per-use

Node Specifications geminidb.redis.large.4 | 2 vCPUs Standard

Current Nodes 2

Current Storage 16 GB

Current Storage (GB)

17 GB

− 17 +

GeminiDB Redis API supports fast scaling, which can complete in seconds without affecting services.

Step 7 After the storage capacity is expanded, go to **Step 3** to add nodes again.

Step 8 On the displayed page, confirm the node configuration details.

- For yearly/monthly instances
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit** and complete the payment.
- For pay-per-use instances
  - If you need to modify your settings, click **Previous**.

– If you do not need to modify your settings, click **Submit**.

----End

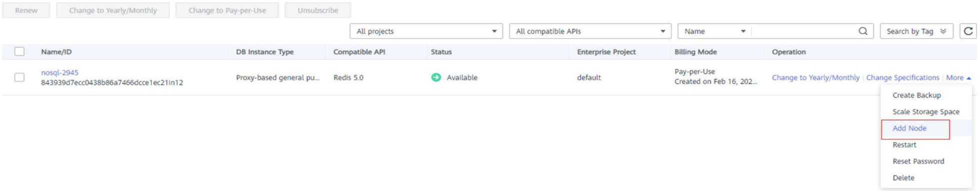
Method 2

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance which you want to add nodes for, click its name, and choose **More** > **Add Node** in the **Operation** column.

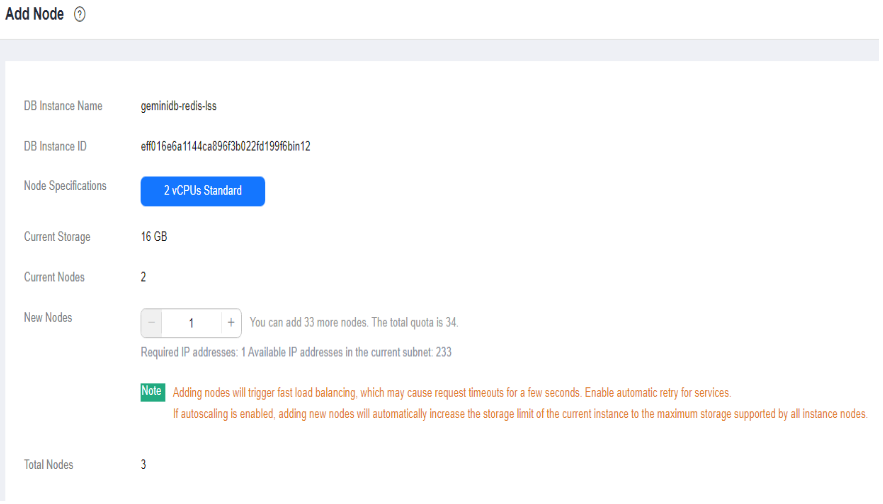
Figure 3-74 Adding nodes



□	Name/ID	DB Instance Type	Compatible API	Status	Enterprise Project	Billing Mode	Operation
□	nosql-2945 843939d7ecc0438b86a7466dcce1ec21in12	Proxy-based general pu...	Redis 5.0	Available	default	Pay-per-Use Created on Feb 16, 202...	Change to Yearly/Monthly | Change Specifications | More

Step 4 On the **Add Node** page, specify the number of nodes to be added and view the storage of the instance.

- If the storage capacity is sufficient, click **Next** and go to **Step 7**.
- If the storage capacity is insufficient, click **Next** and go to **Step 5**.



Add Node

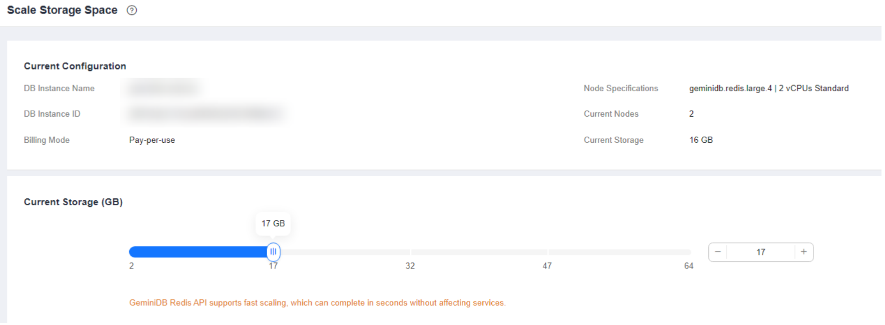
DB Instance Name	geminidb-redis-lss
DB Instance ID	eff016e6a1144ca896f3b022fd199f6bin12
Node Specifications	2 vCPUs Standard
Current Storage	16 GB
Current Nodes	2
New Nodes	1 You can add 33 more nodes. The total quota is 34. Required IP addresses: 1 Available IP addresses in the current subnet: 233
Total Nodes	3

Note Adding nodes will trigger fast load balancing, which may cause request timeouts for a few seconds. Enable automatic retry for services.
If autoscaling is enabled, adding new nodes will automatically increase the storage limit of the current instance to the maximum storage supported by all instance nodes.

New nodes are of the same specifications as existing nodes. Once a new node is added, its specifications cannot be changed.

Step 5 On the **Scale Storage Space** page, select your target storage capacity and click **Next**.

Figure 3-75 Storage change



Scale Storage Space

Current Configuration

DB Instance Name [redacted]

DB Instance ID [redacted]

Billing Mode Pay-per-use

Node Specifications geminidb.redis.large.4 | 2 vCPUs Standard

Current Nodes 2

Current Storage 16 GB

Current Storage (GB)

GeminiDB Redis API supports fast scaling, which can complete in seconds without affecting services.

Step 6 After the storage is scaled up, go to **5** to add nodes again.

Step 7 On the displayed page, confirm the node configuration details.

- For yearly/monthly instances
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit** and complete the payment.
- For pay-per-use instances
  - If you need to modify your settings, click **Previous**.
  - If you do not need to modify your settings, click **Submit**.

----End

3.7.6 Deleting Nodes

Scenarios

You can add or delete nodes for a pay-per-use or yearly/monthly instance to release resources.

Precautions

- Deleted nodes cannot be recovered. Exercise caution when performing this operation.
- If you enable operation protection, two-factor authentication is required for sensitive operations to secure your account and cloud products. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Procedure

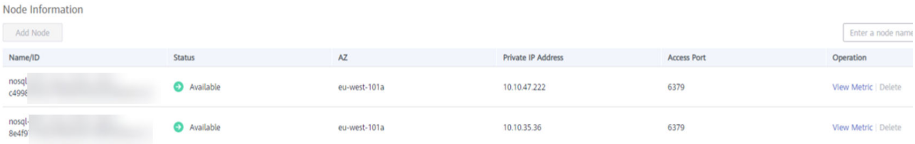
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 On the **Basic Information** page, at the **Node Information** area, locate the node that you want to delete and click **Delete**.

Figure 3-76 Node information



Node Information

Name/ID	Status	AZ	Private IP Address	Access Port	Operation
nosql c4998 [redacted]	Available	eu-west-101a	10.10.47.222	6379	View Metric | Delete
nosql- 8e4f9 [redacted]	Available	eu-west-101a	10.10.35.36	6379	View Metric | Delete

Step 5 If you have enabled operation protection, click **Start Verification** in the **Delete Node** dialog box. On the displayed page, click **Send Code**, enter the verification code, and click **Verify**. The page is closed automatically.

Step 6 In the displayed dialog box, click **Yes**.

- When the node is being deleted, the instance status is **Deleting node**.
- After the node is deleted, the instance status becomes **Available**.

----End

3.7.7 Managing Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage global tags, and other cloud services manage their own tags.

Adding tags to GeminiDB Redis instances helps you better identify and manage them. A DB instance can be tagged during or after it is created.

After a DB instance is tagged, you can search for the tag key or value to quickly query the instance details.

Precautions

- You are advised to set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key. For details about the naming rules of tag keys and tag values, see **Table 3-22**.
- Each instance can have up to 20 tags by default.
- The tag name must comply with the naming rules described in **Table 3-22**.

Table 3-22 Naming rules

Parameter	Requirement	Example Value
Tag key	● Cannot be left blank. ● Must be unique for each instance. ● Contains a maximum of 36 characters. ● Can only consist of digits, letters, underscores (_), and hyphens (-).	Organization

Parameter	Requirement	Example Value
Tag value	● Can be left blank. ● Contains a maximum of 43 characters. ● Can only consist of digits, letters, underscores (_), periods (.), and hyphens (-).	nosql_01

Adding a Tag

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance that you want to add tags to and click its name.

Step 4 In the navigation pane on the left, choose **Tags**.

Step 5 On the **Tags** page, click **Add Tag**. In the displayed dialog box, enter a tag key and value, and click **OK**.

Step 6 View and manage the tag on the **Tags** page.

----End

Editing a Tag

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose tags you want to edit and click its name.

Step 4 In the navigation pane on the left, choose **Tags**.

Step 5 On the **Tags** page, locate the tag to be edited and click **Edit** in the **Operation** column. In the displayed dialog box, change the tag value and click **OK**.

Only the tag value can be edited.

Step 6 View and manage the tag on the **Tags** page.

----End

Deleting a Tag

Step 1 Log in to the GeminiDB console.

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the instance whose tags you want to delete and click its name.

Step 4 In the navigation pane on the left, choose **Tags**.

Step 5 On the **Tags** page, locate the tag to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

Step 6 View that the tag is no longer displayed on the **Tags** page.

----End

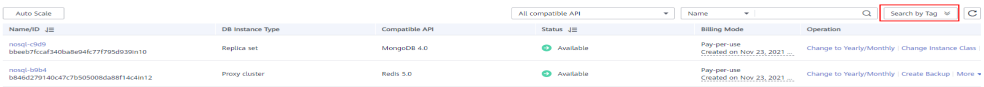
Searching an Instance by Tag

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click **Search by Tag** in the upper right corner of the instance list.

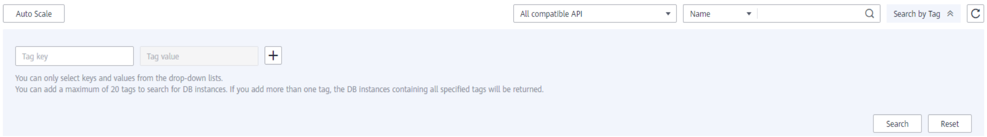
Figure 3-77 Search by Tag



Name/ID	DB Instance Type	Compatible API	Status	Billing Mode	Operation
nosql-c9d9 bbeeb7fccaf340ba8e94fc77f795d939in10	Replica set	MongoDB 4.0	Available	Pay-per-use Created on Nov 23, 2021 ...	Change to Yearly/Monthly | Change Instance Class |
nosql-b9b4 b846d279140c47c7b505008da88f14c4in12	Proxy cluster	Redis 5.0	Available	Pay-per-use Created on Nov 23, 2021 ...	Change to Yearly/Monthly | Create Backup | More

Step 4 Enter a tag key or value and click **Search** to query the instance associated with the tag.

Figure 3-78 Searching by tag key



You can only select keys and values from the drop-down lists.
You can add a maximum of 20 tags to search for DB instances. If you add more than one tag, the DB instances containing all specified tags will be returned.

----End

3.7.8 Updating the OS of an Instance

To improve database performance and security, the OS of a GeminiDB Redis instance needs to be updated in a timely manner.

Every time you upgrade the kernel version of your instance, GeminiDB Redis determines whether to update the OS and selects the right cold patch to upgrade the OS if necessary.

Updating the OS does not change the DB instance version or other information.

In addition, GeminiDB Redis installs hot patches as required to fix major OS vulnerabilities within the maintenance window you specified.

3.8 Audit

3.8.1 Key Operations Supported by CTS

With CTS, you can record GeminiDB Redis key operations for later query, audit, and backtracking.

Table 3-23 GeminiDB Redis key operations

Operation	Resource Type	Trace Name
Creating an instance	instance	NoSQLCreateInstance
Deleting an instance	instance	NoSQLDeleteInstance
Adding nodes	instance	NoSQLEnlargeInstance
Deleting nodes	instance	NoSQLReduceInstance
Restarting an instance	instance	NoSQLRestartInstance
Restoring data to a new instance	instance	NoSQLRestoreNewInstance
Scaling up storage space of an instance	instance	NoSQLExtendInstanceVolume
Resetting the password of an instance	instance	NoSQLResetPassword
Modifying the name of an instance	instance	NoSQLRenameInstance
Binding an EIP	instance	NoSQLResizeInstance
Unbinding an EIP	instance	NoSQLBindEIP
Changing specifications	instance	NoSQLUnBindEIP
Freezing an instance	instance	NoSQLFreezeInstance
Unfreezing an instance	instance	NoSQLUnfreezeInstance
Creating a backup	backup	NoSQLCreateBackup
Deleting a backup	backup	NoSQLDeleteBackup
Setting a backup policy	backup	NoSQLSetBackupPolicy
Adding an instance tag	tag	NoSQLAddTags
Modifying an instance tag	tag	NoSQLModifyInstanceTag
Deleting an instance tag	tag	NoSQLDeleteInstanceTag
Creating a parameter template	parameterGroup	NoSQLCreateConfigurations
Modifying a parameter template	parameterGroup	NoSQLUpdateConfigurations
Modifying instance parameters	parameterGroup	NoSQLUpdateInstanceConfigurations
Replicating a parameter template	parameterGroup	NoSQLCopyConfigurations

Operation	Resource Type	Trace Name
Resetting a parameter template	parameterGroup	NoSQLResetConfigurations
Applying a parameter template	parameterGroup	NoSQLApplyConfigurations
Deleting a parameter template	parameterGroup	NoSQLDeleteConfigurations
Deleting the node that fails to be added	instance	NoSQLDeleteEnlargeFail-Node
Enabling SSL	instance	NoSQLSwitchSSL
Changing the security group of an instance	instance	NoSQLModifySecurityGroup
Modifying the recycling policy	instance	NoSQLModifyRecyclePolicy

3.8.2 Querying Traces

Scenarios

After CTS is enabled, CTS starts recording operations on cloud resources. The CTS console stores the last 7 days of operation records for later query, audit, and backtracking.

This section describes how to query the last 7 days of operation records on the CTS console.

Procedure

Step 1 **Log in to the GeminiDB console.**

Step 2 Click **Service List**. Under **Management & Governance**, click **Cloud Trace Service**.

Step 3 Choose **Trace List** in the navigation pane on the left.

Step 4 Click **Filter** and specify filter criteria as needed. The following filters are available:

- **Trace Type**: Select **Management** or **Data**.
- **Trace Source**, **Resource Type**, and **Search By**

  Select a filter from the drop-down list.

  When you select **Trace name** for **Search By**, you also need to select a specific trace name.

  When you select **Resource ID** for **Search By**, you also need to select or enter a specific resource ID.

  When you select **Resource name** for **Search By**, you also need to select or enter a specific resource name.

- **Operator**: Select a specific operator (a user rather than tenant).
- **Trace Status**: Available options include **All trace statuses**, **normal**, **warning**, and **incident**. You can only select one of them.
- Start time and end time: You can specify a time range for querying traces.

Step 5 Click ⌄ on the left of the record to be queried to extend its details.

Step 6 Locate a trace and click **View Trace** in the **Operation** column.

----End

3.9 Monitoring and Alarm Configuration

3.9.1 GeminiDB Redis Metrics

Description

This section describes GeminiDB Redis metrics reported to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of a monitored object and alarms generated for GeminiDB Redis.

Namespace

SYS.NoSQL

Monitoring Metrics

NOTE

You can view the instance-level and node-level metrics described in **Table 3-24** on each instance node by referring to **Viewing Monitoring Metrics**. The instance-level metrics displayed on each instance node are the same.

Table 3-24 GeminiDB Redis metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
nosql001_cpu_usage	CPU Usage	CPU usage of the monitored system Unit: Percent	0–100	GeminiDB Redis instance nodes	1 minute
nosql002_mem_usage	Memory Usage	Memory usage of the monitored system Unit: Percent	0–100	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
nosql005_disk_usage	Storage Space Usage	Disk usage of the monitored container Unit: Percent	0–100	GeminiDB Redis instances	1 minute
nosql006_disk_total_size	Total Disk Size	Total disk capacity of the monitored container Unit: GB	≥ 0	GeminiDB Redis instances	1 minute
nosql007_disk_used_size	Used Storage Space	Used disk space of the monitored container Unit: GB	≥ 0	GeminiDB Redis instances	1 minute
redis017_proxy_accept	Total Clients Received by Proxy	Total number of clients received by the proxy Unit: count	≥ 0	GeminiDB Redis instance nodes	1 minute
redis018_proxy_request_ps	Request Acceptance Rate	Rate at which the proxy receives client requests Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis019_proxy_response_ps	Proxy Response Rate	Rate at which the proxy returns requests to the client Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis020_proxy_recv_client_bps	Proxy Byte Stream Acceptance Rate	Rate at which the proxy receives byte streams from the client Unit: byte/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis021_proxy_send_client_bps	Proxy Byte Stream Send Rate	Rate at which the proxy sends byte streams to the client Unit: byte/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis032_shard_qps	Shard QPS	QPS of the shard Unit: count	≥ 0	GeminiDB Redis instance nodes	1 minute
redis036_exists_avg_usec	Average Proxy Latency of exists Command	Average latency when the proxy executes the exists command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis037_exists_max_usec	Maximum Proxy Latency of exists Command	Maximum latency when the proxy executes the exists command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis038_exists_p99	Proxy P99 Latency of exists Command	P99 latency when the proxy executes the exists command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis039_exists_qps	Proxy exists Command Rate	Rate at which the proxy executes the exists command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis040_expire_avg_usec	Average Proxy Latency of expire Command	Average latency when the proxy executes the expire command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis041_expire_max_usec	Maximum Proxy Latency of expire Command	Maximum latency when the proxy executes the expire command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis042_expire_p99	Proxy P99 Latency of expire Command	P99 latency when the proxy executes the expire command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis043_expire_qps	Proxy expire Command Rate	Rate at which the proxy executes the expire command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis044_del_avg_usec	Average Proxy Latency of del Command	Average latency when the proxy executes the del command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis045_del_max_usec	Maximum Proxy Latency of del Command	Maximum latency when the proxy executes the del command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis046_del_p99	Proxy P99 Latency of del Command	P99 latency when the proxy executes the del command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis047_del_qps	Proxy del Command Rate	Rate at which the proxy executes the del command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis048_ttl_avg_usec	Average Proxy Latency of ttl Command	Average latency when the proxy executes the ttl command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis049_ttl_max_usec	Maximum Proxy Latency of ttl Command	Maximum latency when the proxy executes the ttl command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis050_ttl_p99	Proxy P99 Latency of ttl Command	P99 latency when the proxy executes the ttl command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis051_ttl_qps	Proxy ttl Command Rate	Rate at which the proxy executes the ttl command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis052_persist_avg_usec	Average Proxy Latency of persist Command	Average latency when the proxy executes the persist command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis053_persist_max_usec	Maximum Proxy Latency of persist Command	Maximum latency when the proxy executes the persist command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis054_persist_p99	Proxy P99 Latency of persist Command	P99 latency when the proxy executes the persist command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis055_persist_qps	Proxy persist Command Rate	Rate at which the proxy executes the persist command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis056_scan_avg_usec	Average Proxy Latency of scan Command	Average latency when the proxy executes the scan command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis057_scan_max_usec	Maximum Proxy Latency of scan Command	Maximum latency when the proxy executes the scan command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis058_scan_p99	Proxy P99 Latency of scan Command	P99 latency when the proxy executes the scan command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis059_scan_qps	Proxy scan Command Rate	Rate at which the proxy executes the scan command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis060_set_avg_usec	Average Proxy Latency of set Command	Average latency when the proxy executes the set command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis061_set_max_usec	Maximum Proxy Latency of set Command	Maximum latency when the proxy executes the set command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis062_set_p99	Proxy P99 Latency of set Command	P99 latency when the proxy executes the set command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis063_set_qps	Proxy set Command Rate	Rate at which the proxy executes the set command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis064_get_avg_usec	Average Proxy Latency of get Command	Average latency when the proxy executes the get command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis065_get_max_usec	Maximum Proxy Latency of get Command	Maximum latency when the proxy executes the get command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis066_get_p99	Proxy P99 Latency of get Command	P99 latency when the proxy executes the get command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis067_get_qps	Proxy get Command Rate	Rate at which the proxy executes the get command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis068_getset_avg_usec	Average Proxy Latency of getset Command	Average latency when the proxy executes the getset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis069_getset_max_usec	Maximum Proxy Latency of getset Command	Maximum latency when the proxy executes the getset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis070_getset_p99	Proxy P99 Latency of getset Command	P99 latency when the proxy executes the getset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis071_getset_qps	Proxy getset Command Rate	Rate at which the proxy executes the getset command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis072_append_avg_usec	Average Proxy Latency of append Command	Average latency when the proxy executes the append command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis073_append_max_usec	Maximum Proxy Latency of append Command	Maximum latency when the proxy executes the append command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis074_append_p99	Proxy P99 Latency of append Command	P99 latency when the proxy executes the append command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis075_append_qps	Proxy append Command Rate	Rate at which the proxy executes the append command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis076_mget_avg_usec	Average Proxy Latency of mget Command	Average latency when the proxy executes the mget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis077_mget_max_usec	Maximum Proxy Latency of mget Command	Maximum latency when the proxy executes the mget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis078_mget_p99	Proxy P99 Latency of mget Command	P99 latency when the proxy executes the mget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis079_mget_qps	Proxy mget Command Rate	Rate at which the proxy executes the mget command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis080_mset_avg_usec	Average Proxy Latency of mset Command	Average latency when the proxy executes the mset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis081_mset_max_usec	Maximum Proxy Latency of mset Command	Maximum latency when the proxy executes the mset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis082_mset_p99	Proxy P99 Latency of mset Command	P99 latency when the proxy executes the mset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis083_mset_qps	Proxy mset Command Rate	Rate at which the proxy executes the mset command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis084_getrange_avg_usec	Average Proxy Latency of getrange Command	Average latency when the proxy executes the getrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis085_getrange_max_usec	Maximum Proxy Latency of getrange Command	Maximum latency when the proxy executes the getrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis086_getrange_p99	Proxy P99 Latency of getrange Command	P99 latency when the proxy executes the getrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis087_getrange_qps	Proxy getrange Command Rate	Rate at which the proxy executes the getrange command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis088_setrange_avg_usec	Average Proxy Latency of setrange Command	Average latency when the proxy executes the setrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis089_setrange_max_usec	Maximum Proxy Latency of setrange Command	Maximum latency when the proxy executes the setrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis090_setrange_p99	Proxy P99 Latency of setrange Command	P99 latency when the proxy executes the setrange command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis091_setrange_qps	Proxy setrange Command Rate	Rate at which the proxy executes the setrange command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis092_substr_avg_usec	Average Proxy Latency of substr Command	Average latency when the proxy executes the substr command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis093_substr_max_usec	Maximum Proxy Latency of substr Command	Maximum latency when the proxy executes the substr command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis094_substr_p99	Proxy P99 Latency of substr Command	P99 latency when the proxy executes the substr command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis095_substr_qps	Proxy substr Command Rate	Rate at which the proxy executes the substr command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis096_strlen_avg_usec	Average Proxy Latency of strlen Command	Average latency when the proxy executes the strlen command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis097_strlen_max_usec	Maximum Proxy Latency of strlen Command	Maximum latency when the proxy executes the strlen command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis098_strlen_p99	Proxy P99 Latency of strlen Command	P99 latency when the proxy executes the strlen command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis099_strlen_qps	Proxy strlen Command Rate	Rate at which the proxy executes the strlen command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis100_incr_avg_usec	Average Proxy Latency of incr Command	Average latency when the proxy executes the incr command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis101_incr_max_usec	Maximum Proxy Latency of incr Command	Maximum latency when the proxy executes the incr command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis102_incr_p99	Proxy P99 Latency of incr Command	P99 latency when the proxy executes the incr command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis103_incr_qps	Proxy incr Command Rate	Rate at which the proxy executes the incr command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis104_decr_avg_usec	Average Proxy Latency of decr Command	Average latency when the proxy executes the decr command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis105_decr_max_usec	Maximum Proxy Latency of decr Command	Maximum latency when the proxy executes the decr command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis106_decr_p99	Proxy P99 Latency of decr Command	P99 latency when the proxy executes the decr command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis107_decr_qps	Proxy decr Command Rate	Rate at which the proxy executes the decr command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis108_hset_avg_usec	Average Proxy Latency of hset Command	Average latency when the proxy executes the hset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis109_hset_max_usec	Maximum Proxy Latency of hset Command	Maximum latency when the proxy executes the hset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis110_hset_p99	Proxy P99 Latency of hset Command	P99 latency when the proxy executes the hset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis111_hset_qps	Proxy hset Command Rate	Rate at which the proxy executes the hset command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis112_hget_avg_usec	Average Proxy Latency of hget Command	Average latency when the proxy executes the hget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis113_hget_max_usec	Maximum Proxy Latency of hget Command	Maximum latency when the proxy executes the hget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis114_hget_p99	Proxy P99 Latency of hget Command	P99 latency when the proxy executes the hget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis115_hget_qps	Proxy hget Command Rate	Rate at which the proxy executes the hget command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis116_hmset_avg_usec	Average Proxy Latency of hmset Command	Average latency when the proxy executes the hmset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis117_hmset_max_usec	Maximum Proxy Latency of hmset Command	Maximum latency when the proxy executes the hmset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis118_hmset_p99	Proxy P99 Latency of hmset Command	P99 latency when the proxy executes the hmset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis119_hmset_qps	Proxy hmset Command Rate	Rate at which the proxy executes the hmset command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis120_hmget_avg_usec	Average Proxy Latency of hmget Command	Average latency when the proxy executes the hmget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis121_hmget_max_usec	Maximum Proxy Latency of hmget Command	Maximum latency when the proxy executes the hmget command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis122_hmget_p99	Proxy P99 Latency of hmget Command	P99 latency when the proxy executes the hmget command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis123_hmget_qps	Proxy hmget Command Rate	Rate at which the proxy executes the hmget command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis124_hdel_avg_usec	Average Proxy Latency of hdel Command	Average latency when the proxy executes the hdel command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis125_hdel_max_usec	Maximum Proxy Latency of hdel Command	Maximum latency when the proxy executes the hdel command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis126_hdel_p99	Proxy P99 Latency of hdel Command	P99 latency when the proxy executes the hdel command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis127_hdel_qps	Proxy hdel Command Rate	Rate at which the proxy executes the hdel command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis128_hgetall_avg_usec	Average Proxy Latency of hgetall Command	Average latency when the proxy executes the hgetall command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis129_hgetall_max_usec	Maximum Proxy Latency of hgetall Command	Maximum latency when the proxy executes the hgetall command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis130_hgetall_p99	Proxy P99 Latency of hgetall Command	P99 latency when the proxy executes the hgetall command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis131_hgetall_qps	Proxy hgetall Command Rate	Rate at which the proxy executes the hgetall command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis132_hexists_avg_usec	Average Proxy Latency of hexists Command	Average latency when the proxy executes the hexists command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis133_hexists_max_usec	Maximum Proxy Latency of hexists Command	Maximum latency when the proxy executes the hexists command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis134_hexists_p99	Proxy P99 Latency of hexists Command	P99 latency when the proxy executes the hexists command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis135_hexists_qps	Proxy hexists Command Rate	Rate at which the proxy executes the hexists command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis136_hincrby_avg_usec	Average Proxy Latency of hincrby Command	Average latency when the proxy executes the hincrby command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis137_hincrby_max_usec	Maximum Proxy Latency of hincrby Command	Maximum latency when the proxy executes the hincrby command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis138_hincrby_p99	Proxy P99 Latency of hincrby Command	P99 latency when the proxy executes the hincrby command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis139_hincrby_qps	Proxy hincrby Command Rate	Rate at which the proxy executes the hincrby command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis140_hkeys_avg_usec	Average Proxy Latency of hkeys Command	Average latency when the proxy executes the hkeys command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis141_hkeys_max_usec	Maximum Proxy Latency of hkeys Command	Maximum latency when the proxy executes the hkeys command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis142_hkeys_p99	Proxy P99 Latency of hkeys Command	P99 latency when the proxy executes the hkeys command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis143_hkeys_qps	Proxy hkeys Command Rate	Rate at which the proxy executes the hkeys command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis144_hlen_avg_usec	Average Proxy Latency of hlen Command	Average latency when the proxy executes the hlen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis145_hlen_max_usec	Maximum Proxy Latency of hlen Command	Maximum latency when the proxy executes the hlen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis146_hlen_p99	Proxy P99 Latency of hlen Command	P99 latency when the proxy executes the hlen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis147_hlen_qps	Proxy hlen Command Rate	Rate at which the proxy executes the hlen command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis148_hstrlen_avg_usec	Average Proxy Latency of hstrlen Command	Average latency when the proxy executes the hstrlen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis149_hstrlen_max_usec	Maximum Proxy Latency of hstrlen Command	Maximum latency when the proxy executes the hstrlen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis150_hstrlen_p99	Proxy P99 Latency of hstrlen Command	P99 latency when the proxy executes the hstrlen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis151_hstrlen_qps	Proxy hstrlen Command Rate	Rate at which the proxy executes the hstrlen command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis152_hvals_avg_usec	Average Proxy Latency of hvals Command	Average latency when the proxy executes the hvals command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis153_hvals_max_usec	Maximum Proxy Latency of hvals Command	Maximum latency when the proxy executes the hvals command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis154_hvals_p99	Proxy P99 Latency of hvals Command	P99 latency when the proxy executes the hvals command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis155_hvals_qps	Proxy hvals Command Rate	Rate at which the proxy executes the hvals command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis156_hscan_avg_usec	Average Proxy Latency of hscan Command	Average latency when the proxy executes the hscan command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis157_hscan_max_usec	Maximum Proxy Latency of hscan Command	Maximum latency when the proxy executes the hscan command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis158_hscan_p99	Proxy P99 Latency of hscan Command	P99 latency when the proxy executes the hscan command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis159_hscan_qps	Proxy hscan Command Rate	Rate at which the proxy executes the hscan command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis160_lpush_avg_usec	Average Proxy Latency of lpush Command	Average latency when the proxy executes the lpush command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis161_lpush_max_usec	Maximum Proxy Latency of lpush Command	Maximum latency when the proxy executes the lpush command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis162_lpush_p99	Proxy P99 Latency of lpush Command	P99 latency when the proxy executes the lpush command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis163_lpush_qps	Proxy lpush Command Rate	Rate at which the proxy executes the lpush command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis164_lpop_avg_usec	Average Proxy Latency of lpop Command	Average latency when the proxy executes the lpop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis165_lpop_max_usec	Maximum Proxy Latency of lpop Command	Maximum latency when the proxy executes the lpop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis166_lpop_p99	Proxy P99 Latency of lpop Command	P99 latency when the proxy executes the lpop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis167_lpop_qps	Proxy lpop Command Rate	Rate at which the proxy executes the lpop command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis168_rpush_avg_usec	Average Proxy Latency of rpush Command	Average latency when the proxy executes the rpush command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis169_rpush_max_usec	Maximum Proxy Latency of rpush Command	Maximum latency when the proxy executes the rpush command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis170_rpush_p99	Proxy P99 Latency of rpush Command	P99 latency when the proxy executes the rpush command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis171_rpush_qps	Proxy rpush Command Rate	Rate at which the proxy executes the rpush command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis172_rpop_avg_usec	Average Proxy Latency of rpop Command	Average latency when the proxy executes the rpop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis173_rpop_max_usec	Maximum Proxy Latency of rpop Command	Maximum latency when the proxy executes the rpop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis174_rpop_p99	Proxy P99 Latency of rpop Command	P99 latency when the proxy executes the rpop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis175_rpop_qps	Proxy rpop Command Rate	Rate at which the proxy executes the rpop command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis176_rpoplpush_avg_usec	Average Proxy Latency of rpoplpush Command	Average latency when the proxy executes the rpoplpush command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis177_rpoplpush_max_usec	Maximum Proxy Latency of rpoplpush Command	Maximum latency when the proxy executes the rpoplpush command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis178_rpoplpush_p99	Proxy P99 Latency of rpoplpush Command	P99 latency when the proxy executes the rpoplpush command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis179_rpoplpush_qps	Proxy rpoplpush Command Rate	Rate at which the proxy executes the rpoplpush command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis180_llen_avg_usec	Average Proxy Latency of llen Command	Average latency when the proxy executes the llen command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis181_llen_max_usec	Maximum Proxy Latency of llen Command	Maximum latency when the proxy executes the llen command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis182_llen_p99	Proxy P99 Latency of llen Command	P99 latency when the proxy executes the llen command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis183_llen_qps	Proxy llen Command Rate	Rate at which the proxy executes the llen command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis184_lindex_avg_usec	Average Proxy Latency of lindex Command	Average latency when the proxy executes the lindex command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis185_lindex_max_usec	Maximum Proxy Latency of lindex Command	Maximum latency when the proxy executes the lindex command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis186_lindex_p99	Proxy P99 Latency of lindex Command	P99 latency when the proxy executes the lindex command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis187_lindex_qps	Proxy lindex Command Rate	Rate at which the proxy executes the lindex command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis188_linsert_avg_usec	Average Proxy Latency of linsert Command	Average latency when the proxy executes the linsert command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis189_linsert_max_usec	Maximum Proxy Latency of linsert Command	Maximum latency when the proxy executes the linsert command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis190_linsert_p99	Proxy P99 Latency of linsert Command	P99 latency when the proxy executes the linsert command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis191_linsert_qps	Proxy linsert Command Rate	Rate at which the proxy executes the linsert command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis192_lrange_avg_usec	Average Proxy Latency of lrange Command	Average latency when the proxy executes the lrange command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis193_lrange_max_usec	Maximum Proxy Latency of lrange Command	Maximum latency when the proxy executes the lrange command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis194_lrange_p99	Proxy P99 Latency of lrange Command	P99 latency when the proxy executes the lrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis195_lrange_qps	Proxy lrange Command Rate	Rate at which the proxy executes the lrange command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis196_lrem_avg_usec	Average Proxy Latency of lrem Command	Average latency when the proxy executes the lrem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis197_lrem_max_usec	Maximum Proxy Latency of lrem Command	Maximum latency when the proxy executes the lrem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis198_lrem_p99	Proxy P99 Latency of lrem Command	P99 latency when the proxy executes the lrem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis199_lrem_qps	Proxy lrem Command Rate	Rate at which the proxy executes the lrem command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis200_lset_avg_usec	Average Proxy Latency of lset Command	Average latency when the proxy executes the lset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis201_lset_max_usec	Maximum Proxy Latency of lset Command	Maximum latency when the proxy executes the lset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis202_lset_p99	Proxy P99 Latency of lset Command	P99 latency when the proxy executes the lset command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis203_lset_qps	Proxy lset Command Rate	Rate at which the proxy executes the lset command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis204_ltrim_avg_usec	Average Proxy Latency of ltrim Command	Average latency when the proxy executes the ltrim command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis205_ltrim_max_usec	Maximum Proxy Latency of ltrim Command	Maximum latency when the proxy executes the ltrim command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis206_ltrim_p99	Proxy P99 Latency of ltrim Command	P99 latency when the proxy executes the ltrim command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis207_ltrim_qps	Proxy ltrim Command Rate	Rate at which the proxy executes the ltrim command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis208_sadd_avg_usec	Average Proxy Latency of sadd Command	Average latency when the proxy executes the sadd command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis209_sadd_max_usec	Maximum Proxy Latency of sadd Command	Maximum latency when the proxy executes the sadd command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis210_sadd_p99	Proxy P99 Latency of sadd Command	P99 latency when the proxy executes the sadd command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis211_sadd_qps	Proxy sadd Command Rate	Rate at which the proxy executes the sadd command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis212_spop_avg_usec	Average Proxy Latency of spop Command	Average latency when the proxy executes the spop command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis213_spop_max_usec	Maximum Proxy Latency of spop Command	Maximum latency when the proxy executes the spop command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis214_spop_p99	Proxy P99 Latency of spop Command	P99 latency when the proxy executes the spop command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis215_spop_qps	Proxy spop Command Rate	Rate at which the proxy executes the spop command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis216_scard_avg_usec	Average Proxy Latency of scard Command	Average latency when the proxy executes the scard command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis217_scard_max_usec	Maximum Proxy Latency of scard Command	Maximum latency when the proxy executes the scard command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis218_scard_p99	Proxy P99 Latency of scard Command	P99 latency when the proxy executes the scard command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis219_scard_qps	Proxy scard Command Rate	Rate at which the proxy executes the scard command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis220_smembers_avg_usec	Average Proxy Latency of smembers Command	Average latency when the proxy executes the smembers command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis221_smembers_max_usec	Maximum Proxy Latency of smembers Command	Maximum latency when the proxy executes the smembers command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis222_smembers_p99	Proxy P99 Latency of smembers Command	P99 latency when the proxy executes the smembers command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis223_smembers_qps	Proxy smembers Command Rate	Rate at which the proxy executes the smembers command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis224_srem_avg_usec	Average Proxy Latency of srem Command	Average latency when the proxy executes the srem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis225_srem_max_usec	Maximum Proxy Latency of srem Command	Maximum latency when the proxy executes the srem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis226_srem_p99	Proxy P99 Latency of srem Command	P99 latency when the proxy executes the srem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis227_srem_qps	Proxy srem Command Rate	Rate at which the proxy executes the srem command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis228_sunion_avg_usec	Average Proxy Latency of sunion Command	Average latency when the proxy executes the sunion command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis229_sunion_max_usec	Maximum Proxy Latency of sunion Command	Maximum latency when the proxy executes the sunion command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis230_sunion_p99	Proxy P99 Latency of sunion Command	P99 latency when the proxy executes the sunion command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis231_sunion_qps	Proxy sunion Command Rate	Rate at which the proxy executes the sunion command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis232_sinter_avg_usec	Average Proxy Latency of sinter Command	Average latency when the proxy executes the sinter command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis233_sinter_max_usec	Maximum Proxy Latency of sinter Command	Maximum latency when the proxy executes the sinter command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis234_sinter_p99	Proxy P99 Latency of sinter Command	P99 latency when the proxy executes the sinter command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis235_sinter_qps	Proxy sinter Command Rate	Rate at which the proxy executes the sinter command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis236_sismember_avg_usec	Average Proxy Latency of sismember Command	Average latency when the proxy executes the sismember command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis237_sismember_max_usec	Maximum Proxy Latency of sismember Command	Maximum latency when the proxy executes the sismember command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis238_sismember_p99	Proxy P99 Latency of sismember Command	P99 latency when the proxy executes the sismember command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis239_sismember_qps	Proxy sismember Command Rate	Rate at which the proxy executes the sismember command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis240_sdiff_avg_usec	Average Proxy Latency of sdiff Command	Average latency when the proxy executes the sdiff command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis241_sdiff_max_usec	Maximum Proxy Latency of sdiff Command	Maximum latency when the proxy executes the sdiff command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis242_sdiff_p99	Proxy P99 Latency of sdiff Command	P99 latency when the proxy executes the sdiff command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis243_sdiff_qps	Proxy sdiff Command Rate	Rate at which the proxy executes the sdiff command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis244_srandmember_avg_usec	Average Proxy Latency of srandmember Command	Average latency when the proxy executes the srandmember command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis245_srandmember_max_usec	Maximum Proxy Latency of srandmember Command	Maximum latency when the proxy executes the srandmember command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis246_srandmember_p99	Proxy P99 Latency of srandmember Command	P99 latency when the proxy executes the srandmember command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis247_srandmember_qps	Proxy srandmember Command Rate	Rate at which the proxy executes the srandmember command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis248_zadd_avg_usec	Average Proxy Latency of zadd Command	Average latency when the proxy executes the zadd command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis249_zadd_max_usec	Maximum Proxy Latency of zadd Command	Maximum latency when the proxy executes the zadd command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis250_zadd_p99	Proxy P99 Latency of zadd Command	P99 latency when the proxy executes the zadd command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis251_zadd_qps	Proxy zadd Command Rate	Rate at which the proxy executes the zadd command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis252_zcard_avg_usec	Average Proxy Latency of zcard Command	Average latency when the proxy executes the zcard command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis253_zcard_max_usec	Maximum Proxy Latency of zcard Command	Maximum latency when the proxy executes the zcard command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis254_zcard_p99	Proxy P99 Latency of zcard Command	P99 latency when the proxy executes the zcard command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis255_zcard_qps	Proxy zcard Command Rate	Rate at which the proxy executes the zcard command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis256_zscan_avg_usec	Average Proxy Latency of zscan Command	Average latency when the proxy executes the zscan command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis257_zscan_max_usec	Maximum Proxy Latency of zscan Command	Maximum latency when the proxy executes the zscan command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis258_zscan_p99	Proxy P99 Latency of zscan Command	P99 latency when the proxy executes the zscan command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis259_zscan_qps	Proxy zscan Command Rate	Rate at which the proxy executes the zscan command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis260_zincrby_avg_usec	Average Proxy Latency of zincrby Command	Average latency when the proxy executes the zincrby command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis261_zincrby_max_usec	Maximum Proxy Latency of zincrby Command	Maximum latency when the proxy executes the zincrby command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis262_zincrby_p99	Proxy P99 Latency of zincrby Command	P99 latency when the proxy executes the zincrby command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis263_zincrby_qps	Proxy zincrby Command Rate	Rate at which the proxy executes the zincrby command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis264_zrevrange_avg_usec	Average Proxy Latency of zrevrange Command	Average latency when the proxy executes the zrevrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis265_zrevrange_max_usec	Maximum Proxy Latency of zrevrange Command	Maximum latency when the proxy executes the zrevrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis266_zrevrange_p99	Proxy P99 Latency of zrevrange Command	P99 latency when the proxy executes the zrevrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis267_zrevrange_qps	Proxy zrevrange Command Rate	Rate at which the proxy executes the zrevrange command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis268_zrange_avg_usec	Average Proxy Latency of zrange Command	Average latency when the proxy executes the zrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis269_zrange_max_usec	Maximum Proxy Latency of zrange Command	Maximum latency when the proxy executes the zrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis270_zrange_p99	Proxy P99 Latency of zrange Command	P99 latency when the proxy executes the zrange command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis271_zrange_qps	Proxy zrange Command Rate	Rate at which the proxy executes the zrange command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis272_zcount_avg_usec	Average Proxy Latency of zcount Command	Average latency when the proxy executes the zcount command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis273_zcount_max_usec	Maximum Proxy Latency of zcount Command	Maximum latency when the proxy executes the zcount command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis274_zcount_p99	Proxy P99 Latency of zcount Command	P99 latency when the proxy executes the zcount command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis275_zcount_qps	Proxy zcount Command Rate	Rate at which the proxy executes the zcount command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis276_zrem_avg_usec	Average Proxy Latency of zrem Command	Average latency when the proxy executes the zrem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis277_zrem_max_usec	Maximum Proxy Latency of zrem Command	Maximum latency when the proxy executes the zrem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis278_zrem_p99	Proxy P99 Latency of zrem Command	P99 latency when the proxy executes the zrem command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis279_zrem_qps	Proxy zrem Command Rate	Rate at which the proxy executes the zrem command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis280_zscore_avg_usec	Average Proxy Latency of zscore Command	Average latency when the proxy executes the zscore command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis281_zscore_max_usec	Maximum Proxy Latency of zscore Command	Maximum latency when the proxy executes the zscore command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis282_zscore_p99	Proxy P99 Latency of zscore Command	P99 latency when the proxy executes the zscore command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis283_zscore_qps	Proxy zscore Command Rate	Rate at which the proxy executes the zscore command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis284_zrank_avg_usec	Average Proxy Latency of zrank Command	Average latency when the proxy executes the zrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis285_zrank_max_usec	Maximum Proxy Latency of zrank Command	Maximum latency when the proxy executes the zrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis286_zrank_p99	Proxy P99 Latency of zrank Command	P99 latency when the proxy executes the zrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis287_zrank_qps	Proxy zrank Command Rate	Rate at which the proxy executes the zrank command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis288_zrevrank_avg_usec	Average Proxy Latency of zrevrank Command	Average latency when the proxy executes the zrevrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis289_zrevrank_max_usec	Maximum Proxy Latency of zrevrank Command	Maximum latency when the proxy executes the zrevrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis290_zrevrank_p99	Proxy P99 Latency of zrevrank Command	P99 latency when the proxy executes the zrevrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis291_zrevrank_qps	Proxy zrevrank Command Rate	Rate at which the proxy executes the zrevrank command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis292_zlexcount_avg_usec	Average Proxy Latency of zlexcount Command	Average latency when the proxy executes the zlexcount command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis293_zlexcount_max_usec	Maximum Proxy Latency of zlexcount Command	Maximum latency when the proxy executes the zlexcount command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis294_zlexcount_p99	Proxy P99 Latency of zlexcount Command	P99 latency when the proxy executes the zlexcount command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis295_zlexcount_qps	Proxy zlexcount Command Rate	Rate at which the proxy executes the zlexcount command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis296_zpopmax_avg_usec	Average Proxy Latency of zpopmax Command	Average latency when the proxy executes the zpopmax command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis297_zpopmax_max_usec	Maximum Proxy Latency of zpopmax Command	Maximum latency when the proxy executes the zpopmax command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis298_zpopmax_p99	Proxy P99 Latency of zpopmax Command	P99 latency when the proxy executes the zpopmax command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis299_zpopmax_qps	Proxy zpopmax Command Rate	Rate at which the proxy executes the zpopmax command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis300_zpopmin_avg_usec	Average Proxy Latency of zpopmin Command	Average latency when the proxy executes the zpopmin command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis301_zpopmin_max_usec	Maximum Proxy Latency of zpopmin Command	Maximum latency when the proxy executes the zpopmin command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis302_zpopmin_p99	Proxy P99 Latency of zpopmin Command	P99 latency when the proxy executes the zpopmin command Unit: µs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis303_zpopmin_qps	Proxy zpopmin Command Rate	Rate at which the proxy executes the zpopmin command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis304_zremrangebyrank_avg_usec	Average Proxy Latency of zremrangebyrank Command	Average latency when the proxy executes the zremrangebyrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis305_zremrangebyrank_max_usec	Maximum Proxy Latency of zremrangebyrank Command	Maximum latency when the proxy executes the zremrangebyrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis306_zremrangebyrank_p99	Proxy P99 Latency of zremrangebyrank Command	P99 latency when the proxy executes the zremrangebyrank command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis307_zremrangebyrank_qps	Proxy zremrangebyrank Command Rate	Rate at which the proxy executes the zremrangebyrank command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis308_zremrangebyscore_avg_usec	Average Proxy Latency of zremrangebyscore Command	Average latency when the proxy executes the zremrangebyscore command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis309_zremrangebyscore_max_usec	Maximum Proxy Latency of zremrangebyscore Command	Maximum latency when the proxy executes the zremrangebyscore command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis310_zremrangebyscore_p99	Proxy P99 Latency of zremrangebyscore Command	P99 latency when the proxy executes the zremrangebyscore command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis311_zremrangebyscore_qps	Proxy zremrangebyscore Command Rate	Rate at which the proxy executes the zremrangebyscore command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis312_zremrangebylex_avg_usec	Average Proxy Latency of zremrangebylex Command	Average latency when the proxy executes the zremrangebylex command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis313_zremrangebylex_max_usec	Maximum Proxy Latency of zremrangebylex Command	Maximum latency when the proxy executes the zremrangebylex command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis314_zremrangebylex_p99	Proxy P99 Latency of zremrangebylex Command	P99 latency when the proxy executes the zremrangebylex command Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis315_zremrangebylex_qps	Proxy zremrangebylex Command Rate	Rate at which the proxy executes the zremrangebylex command Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis316_all_avg_usec	Average Proxy Latency of Commands	Average latency when the proxy executes commands Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis317_all_max_usec	Maximum Proxy Latency of Commands	Maximum latency when the proxy executes commands Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis318_all_p99	Proxy P99 Latency of Commands	P99 latency when the proxy executes all commands Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis319_all_qps	Proxy Command Rate	Rate at which the proxy executes commands Unit: count/s	≥ 0	GeminiDB Redis instance nodes	1 minute
redis661_rsync_ops	rsync Rate	Rate that rsync transfers data in a collection period Unit: count	≥ 0	GeminiDB Redis instance nodes	1 minute
redis662_rsync_wal_size	Size of WAL Files to Be Synchronized	Size of WAL files to be synchronized by rsync in a collection period Unit: byte	≥ 0	GeminiDB Redis instance nodes	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis663_rsync_push_cost	Average Push Time	Average time required for rsync to push data in a collection period Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis664_rsync_send_cost	Average Send Time	Average time required for rsync to send data in a collection period Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis665_rsync_max_push_cost	Maximum Push Time	Maximum time required for a push operation in a collection period Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute
redis666_rsync_max_send_cost	Maximum Send Time	Maximum time required for a send operation in a collection period Unit: μs	≥ 0	GeminiDB Redis instance nodes	1 minute

Dimensions

Key	Value
redis_cluster_id	Cluster ID of the GeminiDB Redis instance
redis_node_id	Node ID of the GeminiDB Redis instance

3.9.2 Configuring Alarm Rules

Scenarios

Setting alarm rules allows you to customize objects to be monitored and notification policies so that you can closely monitor your instances.

Alarm rules include the alarm rule name, instance, metric, threshold, monitoring interval, and whether to send notifications. This section describes how to set alarm rules.

Procedure

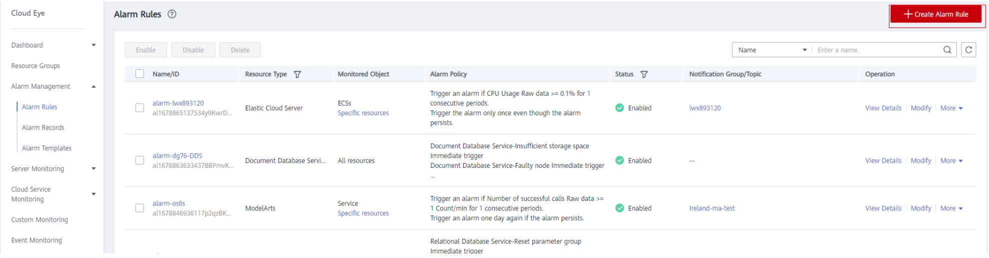
Step 1 Log in to the management console.

Step 2 Click **Service List**. Under **Management & Deployment**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.

Step 4 On the **Alarm Rules** page, click **Create Alarm Rule**.

Figure 3-79 Creating an alarm rule

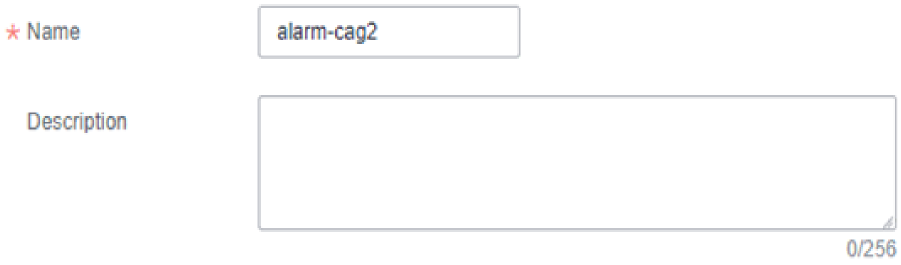


Name/ID	Resource Type	Monitored Object	Alarm Policy	Status	Notification Group/Topic	Operation
alarm-lwx893120 al1678865137534y9KwrD...	Elastic Cloud Server	ECSs Specific resources	Trigger an alarm if CPU Usage Raw data >= 0.1% for 1 consecutive periods. Trigger the alarm only once even though the alarm persists.	Enabled	lwx893120	View Details | Modify | More
alarm-dg76-DDS al1678863633437BBPmvK...	Document Database Servi...	All resources	Document Database Service-Insufficient storage space Immediate trigger Document Database Service-Faulty node Immediate trigger ...	Enabled	--	View Details | Modify | More
alarm-os6s al1678846936117p2qzBK...	ModelArts	Service Specific resources	Trigger an alarm if Number of successful calls Raw data >= 1 Count/min for 1 consecutive periods. Trigger an alarm one day again if the alarm persists.	Enabled	Ireland-ma-test	View Details | Modify | More
			Relational Database Service-Reset parameter group Immediate trigger			

Step 5 Set alarm parameters.

1. Configure basic alarm information.

Figure 3-80 Configuring basic information for an alarm rule



* Name	alarm-cag2
Description	0/256

Table 3-25 Basic alarm rule information

Parameter	Description	Example Value
Name	Name of the rule. The system generates a random name and you can modify it.	alarm-cag2

Parameter	Description	Example Value
Description	(Optional) Alarm rule description.	-

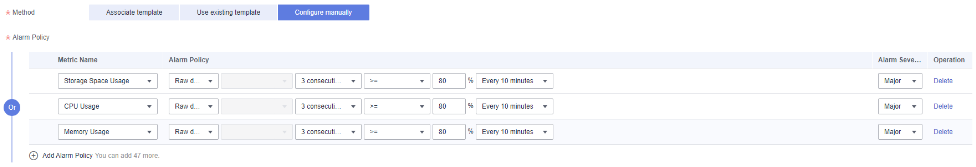
2. Select objects to be monitored and specify the monitoring scope.

Table 3-26 Parameter description

Parameter	Description	Example Value
Alarm Type	Alarm type that the alarm rule is created for. The value can be **Metric** or **Event**.	Metric
Resource Type	Type of the resource the alarm rule is created for. Select **GeminiDB**.	-
Dimension	Metric dimension of the alarm rule. Select **Redis-Redis Nodes**.	-
Monitoring Scope	Monitoring scope the alarm rule applies to. **NOTE** – If you select **Resource groups** and any resource in the group meets the alarm policy, an alarm notification will be sent. – After you select **Specific resources**, select one or more resources and click » to add them to the box on the right.	All resources
Group	This parameter is mandatory when **Monitoring Scope** is set to **Resource groups**.	-

3. Configure an alarm policy.

Figure 3-81 Configuring the alarm policy



* Method: Associate template | Use existing template | Configure manually

* Alarm Policy

Metric Name	Alarm Policy						Alarm Seve...	Operation
Storage Space Usage	Raw d...		3 consecuti...	>=	80 %	Every 10 minutes	Major	Delete
CPU Usage	Raw d...		3 consecuti...	>=	80 %	Every 10 minutes	Major	Delete
Memory Usage	Raw d...		3 consecuti...	>=	80 %	Every 10 minutes	Major	Delete

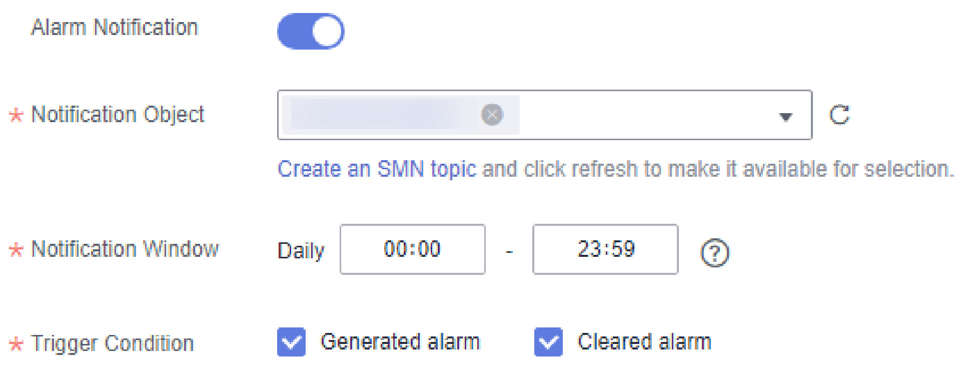
⊕ Add Alarm Policy You can add 47 more.

Table 3-27 Parameter description

Parameter	Description	Example Value
Method	Select **Associate template**, **Use existing template**, or **Configure manually**. NOTE If you set **Monitoring Scope** to **Specific resources**, you can set **Method** to **Use existing template**.	Configure manually
Template	Select the template to be used. This parameter is available only when you select **Use existing template** for **Method**.	-
Alarm Policy	Policy for triggering an alarm. You can configure the threshold, consecutive periods, alarm interval, and alarm severity based on service requirements. – **Metric Name**: specifies the name of the metric configured in the alarm rule. The following metrics are recommended: **Storage Space Usage**, which is used to monitor the storage usage of GeminiDB Redis instances. If the storage usage is greater than 80%, scale up the storage in a timely manner by referring to **Scaling Up Storage Space**. **CPU Usage** and **Memory Usage**, which are used to monitor the compute resource usage of each GeminiDB Redis instance node. If the CPU usage or memory usage is greater than 80%, you can **add nodes** or **upgrade node specifications** in a timely manner. For more metrics, see **GeminiDB Redis Metrics**. – **Alarm Severity**: specifies the severity of the alarm. Valid values are **Critical**, **Major**, **Minor**, and **Informational**. NOTE A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm is triggered.	Take the CPU usage as an example. The alarm policy configured in **Figure 3-81** indicates that a major alarm notification will be sent to users every 10 minutes if the original CPU usage reaches 80% or above for three consecutive periods.

4. Configure alarm notification information.

Figure 3-82 Configuring alarm notification information



Alarm Notification

* Notification Object

Create an SMN topic and click refresh to make it available for selection.

* Notification Window Daily 00:00 - 23:59

* Trigger Condition ☑ Generated alarm ☑ Cleared alarm

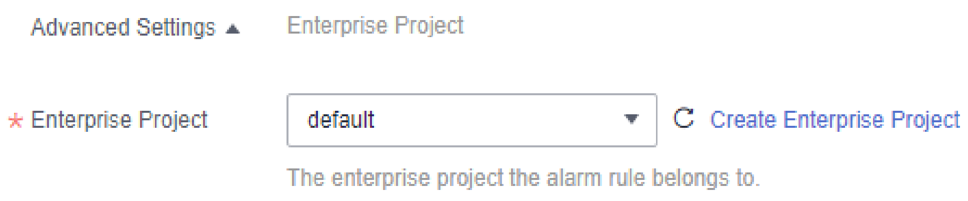
Table 3-28 Parameter description

Parameter	Description	Example Value
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/ HTTPS message. Enabling alarm notification is recommended. When the metric data reaches the threshold set in the alarm rule, Cloud Eye immediately notifies you through SMN that an exception has occurred.	Enabled **Alarm Notification**.
Notification Object	Specifies the object that receives alarm notifications. You can select the account contact or a topic. – Account contact is the mobile phone number and email address provided for registration. – **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it.	-

Parameter	Description	Example Value
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. For example, if **Notification Window** is set to **00:00-8:00**, Cloud Eye sends notifications only within 00:00-08:00.	-
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.	-

5. Configure advanced settings.

Figure 3-83 Advanced settings


Advanced Settings ▲ Enterprise Project

* Enterprise Project: default

The enterprise project the alarm rule belongs to.

Table 3-29 Parameter description

Parameter	Description	Example Value
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.	default

Step 6 After the configuration is complete, click **Create**.

When the metric data reaches the threshold set in the alarm rule, Cloud Eye immediately notifies you through SMN that an exception has occurred.

----End

3.9.3 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors GeminiDB Redis instance running statuses. You can view the GeminiDB Redis monitoring metrics on the management console.

Monitored data requires a period of time for transmission and display. The status of the monitored object displayed on the Cloud Eye page is the status obtained 5 to 10 minutes before. You can view the monitored data of a newly created DB instance 5 to 10 minutes later.

Prerequisites

- The DB instance is running properly.

  Cloud Eye does not display the metrics of a faulty or deleted DB instance. You can view the monitoring information only after the instance is restarted or recovered.

- The DB instance has been properly running for at least 10 minutes.

  The monitoring data and graphics are available for a new DB instance after the instance runs for at least 10 minutes.

Method 1

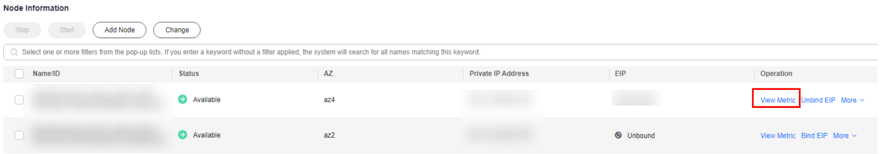
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance whose metrics you want to view and click its name.

Step 4 In the navigation pane on the left, choose **Node Management**. In the **Node Information** area, click **View Metric** in the **Operation** column.

Figure 3-84 Viewing metrics



Select one or more filters from the pop-up lists. If you enter a keyword without a filter applied, the system will search for all names matching this keyword.

□ Name/ID	Status	AZ	Private IP Address	EIP	Operation
□ [redacted]	Available	az4	[redacted]	[redacted]	View Metric Unbind EIP More ⌄
□ [redacted]	Available	az2	[redacted]	Unbound	View Metric Bind EIP More ⌄

Step 5 In the monitoring area, you can select a duration to view the monitoring data.

You can view the monitoring data of the service in the last 1, 3, or 12 hours.

To view the monitoring curve in a longer time range, click ⤡ to enlarge the graph.

----End

3.9.4 Configuring a Dashboard

Dashboards, serving as custom monitoring platforms, allow you to view metrics.

This section describes how to configure a dashboard for GeminiDB Redis.

Procedure

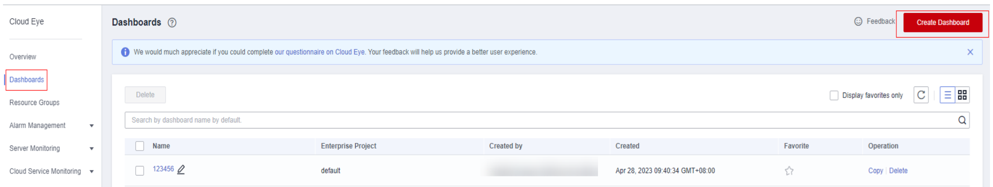
Step 1 Log in to the GeminiDB console.

Step 2 In the service list, click **Cloud Eye** to go to the Cloud Eye console.

Step 3 Create a monitoring dashboard.

1. In the navigation pane on the left, choose **Dashboards**. On the displayed page, click **Create Dashboard**.

   Figure 3-85 Creating a dashboard

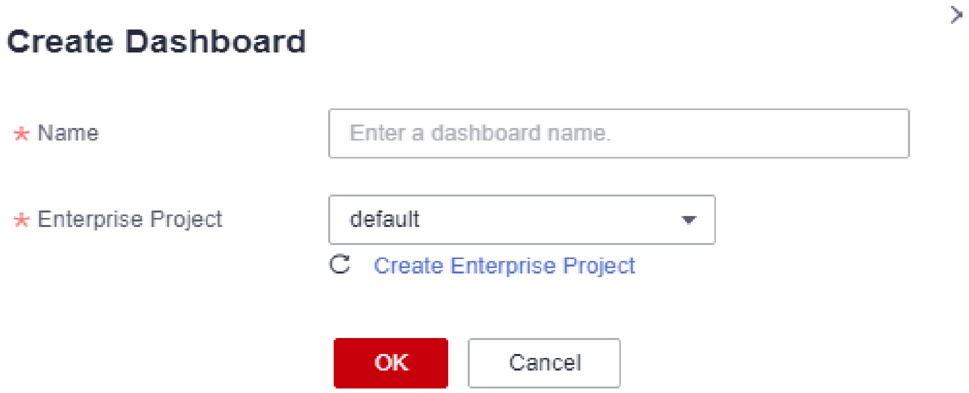


We would much appreciate if you could complete our questionnaire on Cloud Eye. Your feedback will help us provide a better user experience.

Name	Enterprise Project	Created by	Created	Favorite	Operation
123456	default	[redacted]	Apr 28, 2023 09:40:34 GMT+08:00	☆	Copy | Delete

2. In the displayed **Create Dashboard** dialog box, set required parameters.

   Figure 3-86 Configuring parameters



* Name: Enter a dashboard name.

* Enterprise Project: default

Table 3-30 Parameter description

Parameter	Description
Name	Dashboard name. The name can include a maximum of 128 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Enterprise Project	If you associate a monitoring dashboard with an enterprise project, only users who have the permissions of the enterprise project can manage the monitoring dashboard. **NOTE** The enterprise project feature is available only in some regions.

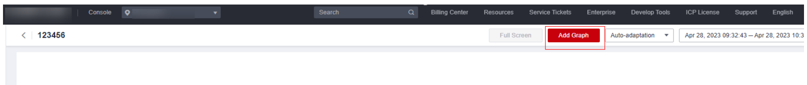
3. Click **OK**.

Step 4 Add a graph to the monitoring dashboard.

After a dashboard is created, you can add graphs to monitor your GeminiDB Redis instances.

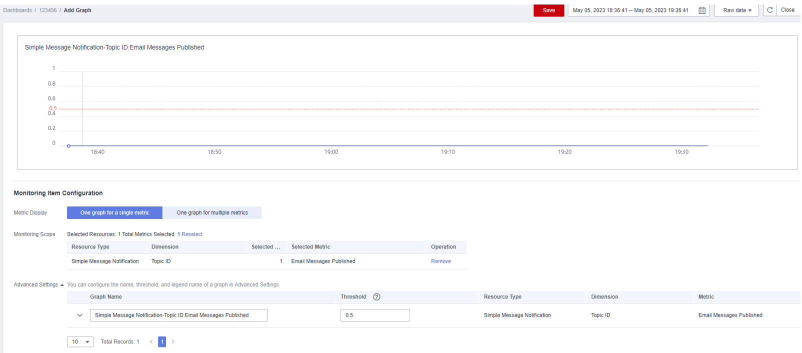
1. In the navigation pane on the left, choose **Dashboard**. Locate the dashboard that you want to add a graph to, and click **Add Graph**.

Figure 3-87 Adding a graph



2. On the **Add Graph** page, select a line chart or bar chart.
   - A curve chart reflects changes and peak values of a metric over time.
   - A bar chart reflects metric data of top-ranked resources of the same type, helping you to understand upper and lower limits of a metric.
3. At the **Monitoring Item Configuration** area, configure required parameters by referring to **Table 3-31**.

Figure 3-88 Monitored item configuration


Dashboards / 123456 / Add Graph
Save
May 05, 2023 18:36:41 — May 05, 2023 19:36:41
Raw data
Close
Simple Message Notification-Topic ID:Email Messages Published
1
0.8
0.6
0.5
0.4
0.2
0
18:40
18:50
19:00
19:10
19:20
19:30

Monitoring Item Configuration

Metric Display: One graph for a single metric | One graph for multiple metrics

Monitoring Scope: Selected Resources: 1 Total Metrics Selected: 1 Reselect

Resource Type	Dimension	Selected ...	Selected Metric	Operation
Simple Message Notification	Topic ID	1	Email Messages Published	Remove

Advanced Settings ▲ You can configure the name, threshold, and legend name of a graph in Advanced Settings

Graph Name	Threshold	Resource Type	Dimension	Metric
Simple Message Notification-Topic ID:Email Messages Published	0.5	Simple Message Notification	Topic ID	Email Messages Published

10 Total Records: 1 < 1 >

Table 3-31 Parameter description

Parameter	Description
Metric Display	– **One graph for a single metric**: One or more graphs can be generated, and all monitoring items in each graph represent the same metric. – **One graph for multiple metrics**: One graph is generated for multiple metrics, and monitoring items can represent different metrics.
Monitoring Scope	Specify resources and metrics.
Graph Name	Specifies the title of the graph to be added. The name can contain a maximum of 128 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. Example: CPU usage
Threshold	Configure a threshold to generate an auxiliary line. Data points higher than the line are highlighted in red.

Parameter	Description
Legend Name	The legend name is displayed on the line in a monitoring graph and can be changed. If you do not configure the legend name, it is displayed in the following format by default: *monitored object* (*resource type*) - *metric*: *monitored data*.

NOTE

When you add a graph, select **One graph for a single metric**. Then a graph is generated for each metric, making it easy for you to view and analyze monitored data. If you need multiple metrics, add monitoring graphs.

4. On the selected dashboard, view metric trends on the added graph.

----End

3.10 Data Backup

3.10.1 Overview

GeminiDB Redis allows you to back up instances to protect your data. After an instance is deleted, the manual backup data is retained. Automatic backup data is released together with instances. Backup data cannot be downloaded or exported.

Backup Methods

Both automatic backup and manual backup are supported.

- Automated backup

  You can **modify backup policy** on the GeminiDB console, and the system will automatically back up your instance data based on the time window and backup cycle you configure in the backup policy and will store the data for a length of time you specify.

  Automated backups cannot be manually deleted. You can adjust their retention period by referring to **Modifying an Automated Backup Policy**, and backups that expire will be automatically deleted.

- Manual backup

  A manual backup is a full backup of a DB instance and can be retained until you manually delete it. Manual backup can be triggered at any time to meet your service requirements.

  Regularly backing up your database is recommended. If your database becomes faulty or data is corrupted, you can restore it from backup.

Table 3-32 Backup methods

Method	Scenario
Automated backup	After you set a backup policy, the system automatically backs up your database based on the policy. You can also modify the policy based on service requirements.
Manual backup	You can manually create full backups for your instance based on service requirements.

How Backup Works

GeminiDB Redis API takes snapshots of persistent data in seconds an d then stores them as compressed packages in OBS, without using any of the storage space of your instance. GeminiDB Redis API consumes a few compute resources during backup, so it is normal if the instance CPU usage and memory usage increase slightly.

Redis Community Edition is slow in backup and jitter may happen in performance. By contrast, GeminiDB Redis API backs up data faster, and almost no jitter occurs during the backup.

Backup Storage

Backups are stored in OBS buckets to provide disaster recovery and save storage space.

After you purchase an instance, GeminiDB Redis will provide additional backup storage of the same size as what you purchased. For example, if you purchase an instance with 100 GB of storage, you will obtain additional 100 GB of storage free of charge. If the backup data does not exceed 100 GB, it is stored on OBS free of charge. If there is more than 100 GB of data, you will be billed at standard OBS rates.

3.10.2 Managing Automated Backups

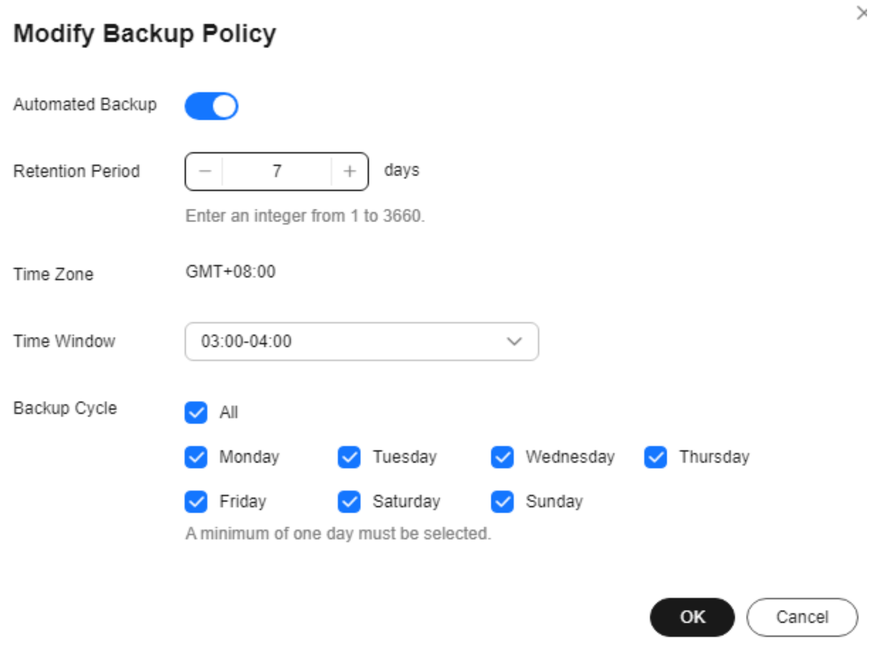
GeminiDB Redis allows you to create automated backups to protect your data. If a database or table is deleted, maliciously or accidentally, backups can help recover your data.

Configuring an Automated Backup Policy

Automated backups are generated based on a backup policy and saved as packages in OBS buckets to secure and protect your data. Regularly backing up your database is recommended. If your database becomes faulty or data is corrupted, you can restore it from backup. Backing up data affects the database read and write performance, so you are advised to set the automated backup time window to off-peak hours.

When you create an instance, automated backup is enabled by default.

Figure 3-89 Modifying backup policies



Modify Backup Policy

Automated Backup

Retention Period 7 days

Enter an integer from 1 to 3660.

Time Zone GMT+08:00

Time Window 03:00-04:00

Backup Cycle ☑ All

☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday

☑ Friday ☑ Saturday ☑ Sunday

A minimum of one day must be selected.

- **Retention Period**: Automated backup files are saved for seven days by default. Full backups are retained till the retention period expires.
  - Extending the retention period improves data reliability. You can extend the retention period as needed.
  - If you shorten the retention period, the new backup policy takes effect for existing backups. Any automated backups (including full and incremental backups) that have expired will be automatically deleted. Manual backups will not be automatically deleted but you can delete them manually.

  NOTE

  - If the retention period is less than seven days, the system automatically backs up data daily.
  - The system checks existing automated backups and deletes any backups that exceed the backup retention period you configure.
  - **Time Window**: A one-hour period the backup will be scheduled for, such as 04:00–05:00. The backup time is in GMT format. If the DST or standard time is switched, the backup time segment changes with the time zone.

  If **Retention Period** is set to **2**, full and incremental backups that have been stored for more than two days will be automatically deleted. For instance, a backup generated on Monday will be deleted on Wednesday; or a backup generated on Tuesday will be deleted on Thursday.

  Policy for automatically deleting full backups:

  To ensure data integrity, even after the retention period expires, the most recent backup will be retained, for example,

  If **Backup Cycle** was set to **Monday** and **Tuesday** and the **Retention Period** was set to **2**:

  - A full backup generated on Monday will be automatically deleted on Thursday. The reasons are as follows:

The full backup generated on Monday expires on Wednesday, but it is the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday.

  - The full backup generated on Tuesday will be automatically deleted on the following Wednesday. The reasons are as follows:

    The backup generated on Tuesday will expire on Thursday, but as it is the last backup, so it will be retained until a new backup expires. The next backup will be generated on the following Monday and will expire on the following Wednesday. So the full backup generated on Tuesday will not be automatically deleted until the following Wednesday.

- **Backup Cycle**: All options are selected by default.
  - **All**: Each day of the week is selected. The system automatically backs up data every day.
  - Select a cycle: You can select one or more days in a week. The system automatically backs up data at the specified time.

  NOTE

  A full backup starts within one hour of the time you specify. The amount of time required for the backup depends on the amount of data to be backed up. The more data has to be backed up, the longer it will take.

- After the DB instance is created, you can modify the automated backup policy as needed. You can change the time window after the DB instance is created. The system backs up data based on the automated backup policy you have set.
- If the automated backup policy is disabled, any automated backups in progress stop immediately.

Modifying an Automated Backup Policy

Step 1 **Log in to the GeminiDB console.**

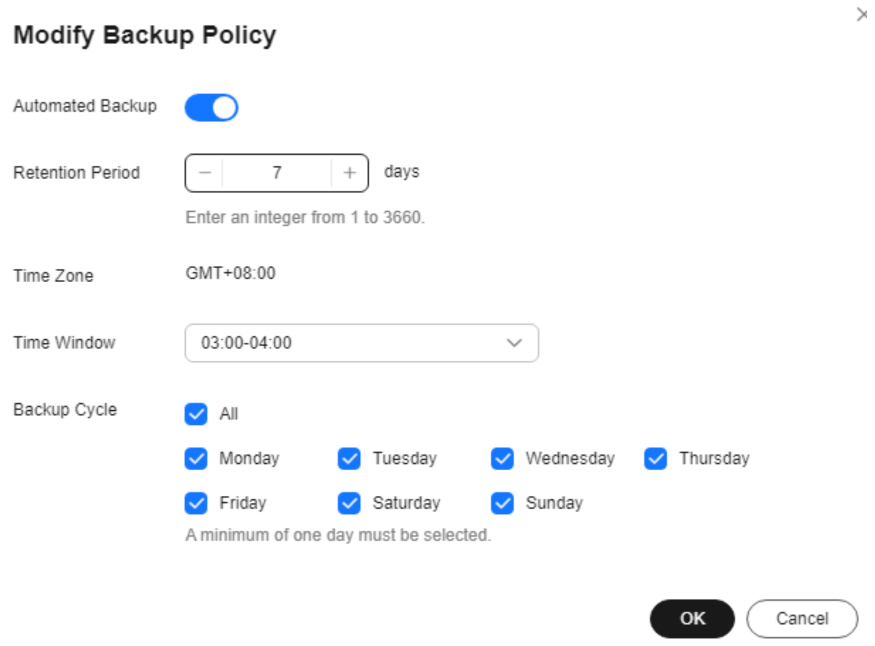
Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance whose backup policy you want to modify.

Step 4 Choose **Backups & Restorations** in the navigation pane one the left, and click **Modify Backup Policy**. In the displayed dialog box, set the backup policy. Then, click **OK**.

For details about how to set a backup policy, see **Configuring an Automated Backup Policy**.

Figure 3-90 Modifying a backup policy



Modify Backup Policy

Automated Backup: on

Retention Period: 7 days

Enter an integer from 1 to 3660.

Time Zone: GMT+08:00

Time Window: 03:00-04:00

Backup Cycle: ☑ All

☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday

☑ Friday ☑ Saturday ☑ Sunday

A minimum of one day must be selected.

Step 5 Check or manage the generated backups on the **Backups** or **Backups & Restorations** page.

----End

Disabling Automated Backup

Step 1 **Log in to the GeminiDB console.**

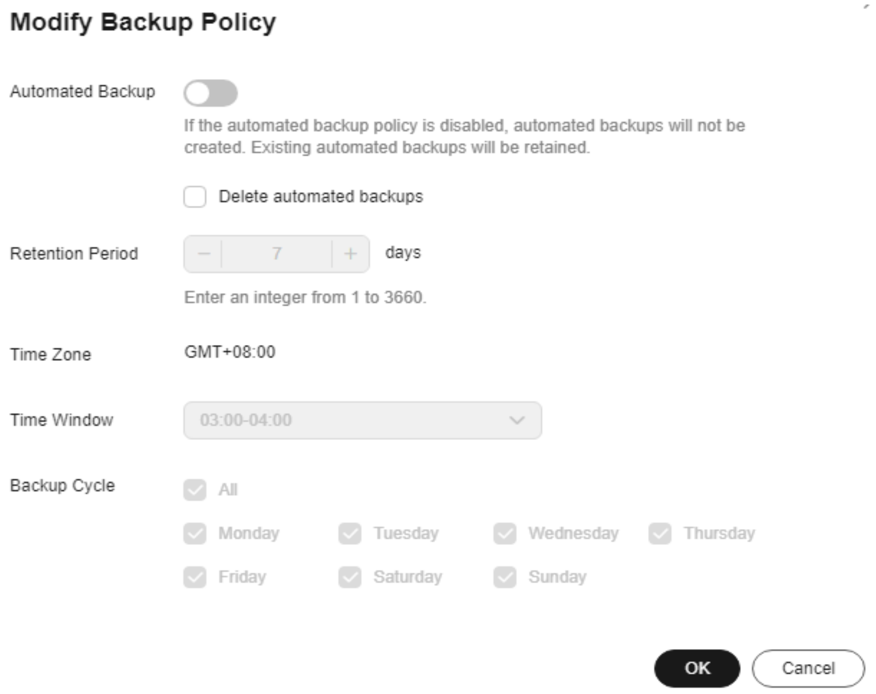
Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance whose backup policy you want to modify.

Step 4 Choose **Backups & Restorations** in the navigation pane one the left, and click **Modify Backup Policy**.

Step 5 In the displayed dialog box, click to disable the backup policy and click **OK**.

Figure 3-91 Disabling backup policies



Modify Backup Policy

Automated Backup ○

If the automated backup policy is disabled, automated backups will not be created. Existing automated backups will be retained.

☐ Delete automated backups

Retention Period − 7 + days

Enter an integer from 1 to 3660.

Time Zone GMT+08:00

Time Window 03:00-04:00

Backup Cycle ☑ All

☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday

☑ Friday ☑ Saturday ☑ Sunday

When disabling the automated backup policy, you can decide whether to delete the automated backups by selecting **Delete automated backups**.

- If you select it, all backup files within the retention period will be deleted. No automated backups are displayed in the backup list until you enable the automated backup policy again.
- If you do not select it, all backup files within the retention period will be retained, but you can still manually delete them later if needed. For details, see **Deleting an Automated Backup**.

If automated backup is disabled, any automated backups in progress stop immediately.

----End

Deleting an Automated Backup

If automated backup is disabled, you can delete stored automated backups to free up storage space.

If automated backup is enabled, the system will delete automated backups as they expire. You cannot delete them manually.

NOTICE

Deleted backups cannot be recovered. Exercise caution when performing this operation.

- **Method 1**

a. **Log in to the GeminiDB console.**

b. In the service list, choose **Databases** > **GeminiDB**.

c. On the **Instances** page, click the instance whose backup you want to delete.

d. On the **Backups & Restorations** page, locate the backup you want to delete and click **Delete**.

e. In the **Delete Backup** dialog box, confirm the backup details and click **Yes**.

- **Method 2**

  a. **Log in to the GeminiDB console.**

  b. In the service list, choose **Databases** > **GeminiDB**.

  c. On the **Backups** page, locate the backup that you want to delete and click **Delete**.

  d. In the **Delete Backup** dialog box, confirm the backup details and click **Yes**.

Setting the Policy for Restoring Data to a Specified Time Point

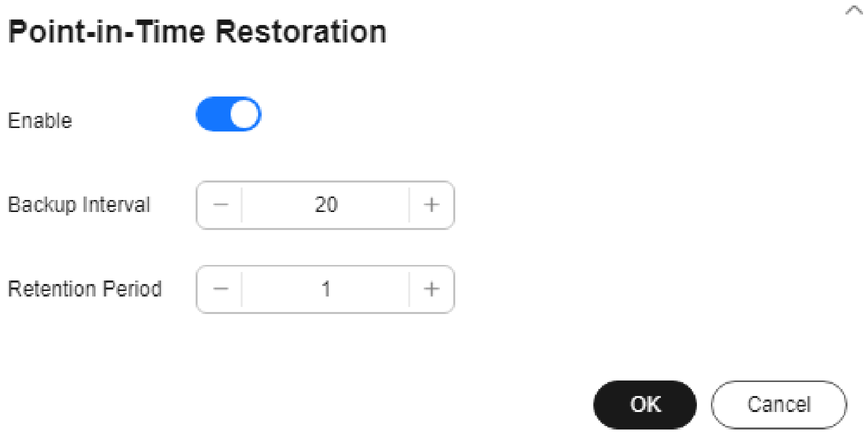
Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the target instance.

Step 4 Choose **Backups & Restorations** in the navigation pane one the left, and click **Point in Time Restoration**. After the setting is complete, click **OK**.

Figure 3-92 Setting the policy for restoring data to a specified time point



Point-in-Time Restoration

Enable: on

Backup Interval: 20

Retention Period: 1

- You can toggle on or off **Enable** to enable or disable the backup function.
- **Backup Interval** refers to the time interval, in minutes, for automated backups. The value ranges from 5 to 120. For example, if the initial backup is scheduled for 04:00, the subsequent backup will take place at 04:05.

- **Retention Period** determines how long automated backups are kept in days. The value ranges from 1 to 7. Full backups are retained till the retention period expires.

----End

3.10.3 Managing Manual Backups

GeminiDB Redis API allows you to manually back up instances whose status is **Available** to protect your data. If a database or table is deleted, maliciously or accidentally, backups can help recover your data.

NOTE

- Manual backups are full backups.

Creating a Manual Backup

Step 1 Log in to the GeminiDB console.

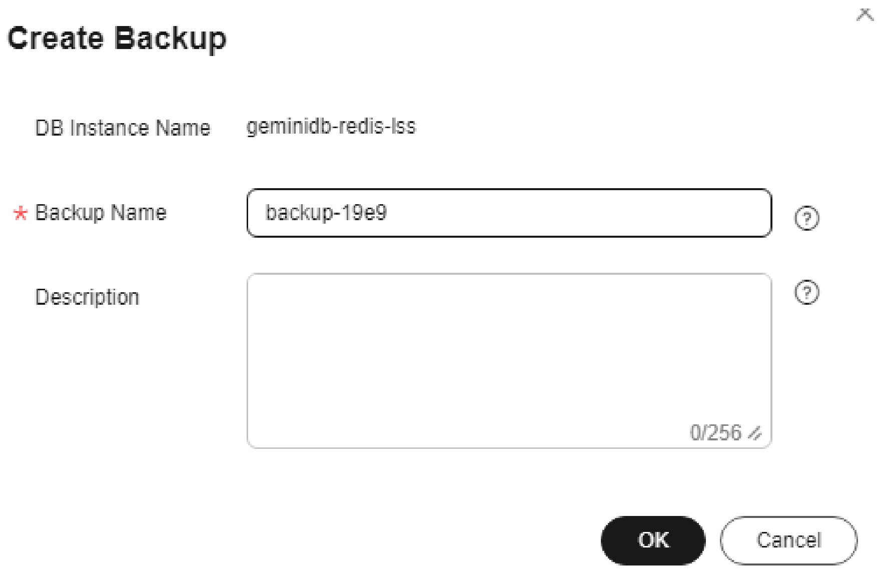
Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 Create a manual backup.

Method 1

On the **Instances** page, locate the instance you want to back up and choose **More** > **Create Backup** in the **Operation** column.

Figure 3-93 Creating a manual backup



Create Backup

DB Instance Name geminidb-redis-lss

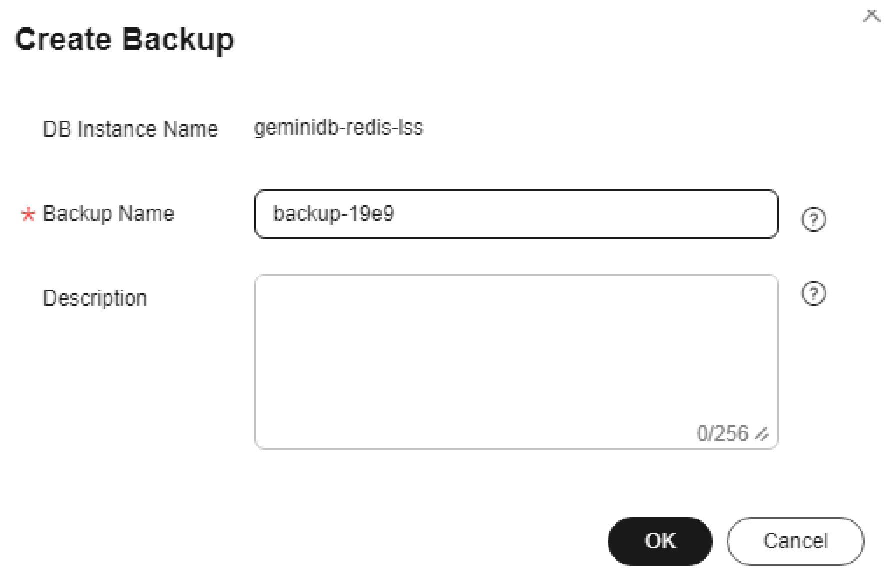
* Backup Name backup-19e9

Description

Method 2

1. On the **Instances** page, click the instance you want to back up.
2. Choose **Backups & Restorations** in the navigation pane on the left, and click **Create Backup**.

Figure 3-94 Creating a manual backup



Create Backup

DB Instance Name: geminidb-redis-lss

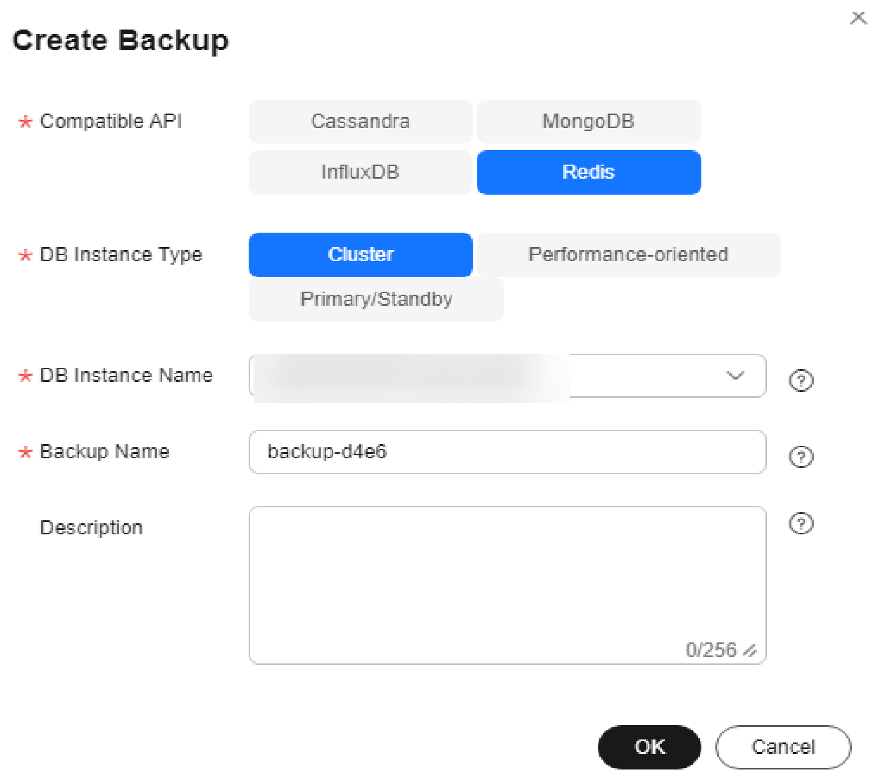
* Backup Name: backup-19e9

Description: 0/256

Method 3

In the navigation pane on the left, choose **Backups**. On the displayed page, click **Create Backup**.

Figure 3-95 Creating a manual backup



Create Backup

* Compatible API: Cassandra, MongoDB, InfluxDB, Redis (selected)

* DB Instance Type: Cluster (selected), Performance-oriented, Primary/Standby

* DB Instance Name: [redacted]

* Backup Name: backup-d4e6

Description: 0/256

Step 4 In the displayed dialog box, specify a backup name and description and click **OK**.

Table 3-33 Parameter description

Parameter	Description
DB Instance Name	Must be the name of the DB instance to be backed up and cannot be modified.
Backup Name	Must be 4 to 64 characters long and start with a letter. It is case-insensitive and contains only letters, digits, hyphens (-), and underscores (_).
Description	Can include a maximum of 256 characters and cannot contain line breaks and the following special characters: >!<"&'=

Step 5 View the backup status.

- When the backup is being created, query the backup status on the **Backups** or **Backups & Restorations** page. The backup status is **Backing up**.
- After the backup is created, the backup status changes to **Completed**.

----End

Deleting a Manual Backup

If you do not need a manual backup any longer, delete it on the **Backups** or **Backups & Restorations** page.

Deleted backups are not displayed in the backup list.

NOTICE

Deleted backups cannot be recovered. Exercise caution when performing this operation.

Method 1

1. **Log in to the GeminiDB console.**
2. In the service list, choose **Databases** > **GeminiDB**.
3. On the **Instances** page, locate the instance whose backup you want to delete and click its name.
4. Choose **Backups & Restorations** in the navigation pane on the left, locate the backup you want to delete, and click **Delete** in the **Operation** column.
5. In the displayed dialog box, confirm the backup details and click **Yes**.

Method 2

1. **Log in to the GeminiDB console.**
2. In the service list, choose **Databases** > **GeminiDB**.
3. On the **Backups** page, locate the backup that you want to delete and click **Delete**.

4. In the displayed dialog box, confirm the backup details and click **Yes**.

3.11 Data Restoration

3.11.1 Restoration Methods

GeminiDB Redis supports multiple forms of data restoration. You can select one based on service requirements.

Table 3-34 Restoration methods

Method	Scenario
Restoring Data to a New Instance	You can restore an existing backup file to a new instance.

3.11.2 Restoring Data to a New Instance

Scenarios

GeminiDB Redis allows you to use an existing backup to restore data to a new instance.

Procedure

Step 1 Log in to the GeminiDB console.

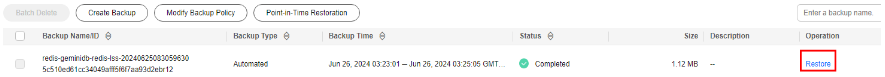
Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 Restore a DB instance from the backup.

Method 1

1. On the **Instances** page, locate the instance whose backup you want to restore and click its name.
2. Choose **Backups & Restorations** in the navigation pane on the left, locate the backup that you want to restore and click **Restore** in the **Operation** column.

Figure 3-96 Restoration

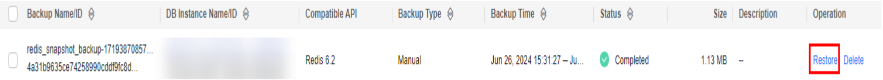


Backup Name/ID	Backup Type	Backup Time	Status	Size	Description	Operation
redis-geminidb-redis-lss-20240625083059630 5c510ed61cc34049afff5f6f7aa93d2ebr12	Automated	Jun 26, 2024 03:23:01 — Jun 26, 2024 03:25:05 GMT...	Completed	1.12 MB	--	Restore

Method 2

On the **Backups** page, locate the backup that you want to restore and click **Restore** in the **Operation** column.

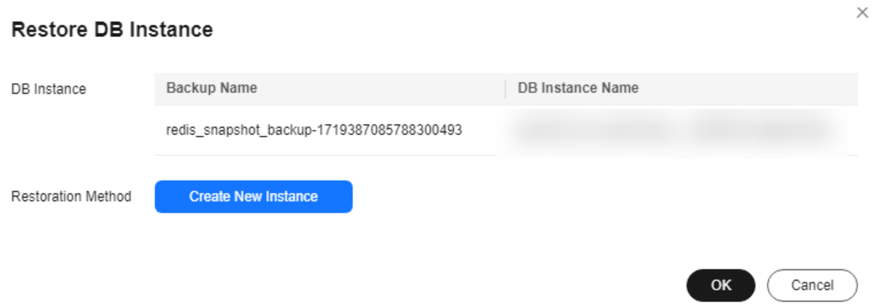
Figure 3-97 Restoration



□ Backup Name/ID	DB Instance Name/ID	Compatible API	Backup Type	Backup Time	Status	Size	Description	Operation
□ redis_snapshot_backup-17193870857... 4a31b9635ce74258990cddf9fc8d...	[redacted]	Redis 6.2	Manual	Jun 26, 2024 15:31:27 – Ju...	Completed	1.13 MB	--	Restore Delete

Step 4 In the displayed dialog box, confirm the current instance details and restoration method and click **OK**.

Figure 3-98 Restoring data to a new DB instance



Restore DB Instance

DB Instance

Backup Name	DB Instance Name
redis_snapshot_backup-1719387085788300493	[redacted]

Restoration Method: Create New Instance

- The default API type and DB engine version are the same as those of the original instance and cannot be changed.
- The new instance must have no less than nodes than the original instance.
- GeminiDB automatically calculates the minimum storage space required for restoration based on the size of the selected backup file. The storage capacity depends on the instance specifications, and must be an integer.
- You need to set a new administrator password.
- To modify other parameters, see the description of buying instances of other DB APIs in *Getting Started*.

Step 5 View the restoration results.

A new instance is created using the backup data. The status of the new instance changes from **Creating** to **Available**.

After the restoration, the system will perform a full backup.

The new DB instance is independent from the original one.

----End

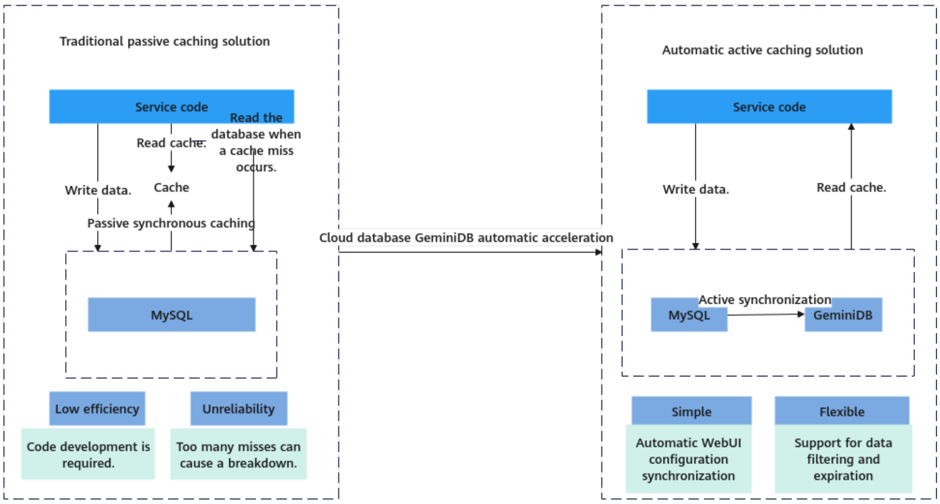
3.12 Memory Acceleration

3.12.1 Memory Acceleration Overview

GeminiDB Redis API offers memory acceleration to enhance the conventional cache solution. With this feature, users can set up rules on the GUI to cache MySQL data automatically, thereby speeding up MySQL access.

The conventional cache solution is inefficient and unreliable as it necessitates users to create code for writing MySQL data to the cache. The active cache solution with cloud data memory acceleration (DB Cache) supports visualized configuration on the GUI, making it easier to set up. Once the configuration is done, data can be synchronized automatically. DB Cache also supports data filtering and expiration time setting, which enhances development efficiency and data reliability.

Figure 3-99 Diagram


Traditional passive caching solution
Service code
Read cache.
Read the database when a cache miss occurs.
Write data.
Cache
Passive synchronous caching
MySQL
Low efficiency
Unreliability
Code development is required.
Too many misses can cause a breakdown.
Cloud database GeminiDB automatic acceleration
Automatic active caching solution
Service code
Write data.
Read cache.
Active synchronization
MySQL
GeminiDB
Simple
Flexible
Automatic WebUI configuration synchronization
Support for data filtering and expiration

3.12.2 Enabling Memory Acceleration

To enable memory acceleration for an existing MySQL instance, you must first create a GeminiDB instance.

Precautions

After memory acceleration is enabled, commands such as RESET MASTER and FLUSH LOGS to delete binlogs on MySQL instances are not allowed to be executed.

Procedure

Step 1 Log in to the management console.

Step 2 Click ≡ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the navigation pane on the left, choose **Memory Acceleration**. On the displayed page, click **Create GeminiDB Instance**.

Step 5 On the displayed page, set required parameters and click **Submit**.

Table 3-35 Basic information

Parameter	Description
Instance Class	CPU and memory of the instance. For details, see **Table 3-36**.
Database Port	Port number for accessing the instance. You can specify a port number based on your requirements. The port number ranges from 1024 to 65535 except 2180, 2887, 3887, 6377, 6378, 6380, 8018, 8079, 8091, 8479, 8484, 8999, 12017, 12333, and 50069. If you do not specify a port number, port 6379 is used by default. **NOTE** You cannot change the database port after an instance is created.
DB Instance Name	The instance name: - Can be the same as an existing instance name. - Can include 4 to 64 bytes and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_).
Database Password	Database password set by the user. - Must be 8 to 32 characters long. - Can include two of the following: uppercase letters, lowercase letters, digits, and special characters: ~!@#%^*-_=+? - For security reasons, set a strong password. The system will verify the password strength. Keep your password secure. The system cannot retrieve it if it is lost.
Confirm Password	Enter the administrator password again.

Table 3-36 GeminiDB Redis instance specifications

Storage (GB)	Nodes	vCPUs	QPS	Maximum Connections per Single-node Instance	Databases
24	2	2	40,000	10,000	1,000
32	2	2	40,000	10,000	1,000
48	2	4	80,000	20,000	1,000
64	2	4	80,000	20,000	1,000

Storage (GB)	Nodes	vCPUs	QPS	Maximum Connections per Single-node Instance	Databases
96	2	8	160,000	20,000	1,000
128	2	16	320,000	20,000	1,000

----End

3.12.3 Managing Mapping Rules

You can create mapping rules to automatically synchronize data from MySQL instances to GeminiDB instances. This section describes how to create, modify, and delete a mapping rule, and provides an example of creating a mapping rule.

Precautions

- Currently, only hashes from MySQL can be converted to GeminiDB Redis API.
- The Redis key prefix + key separator of a new rule cannot be the subprefix of the Redis key prefix + key separator of an existing rule, and vice versa. For example, if the prefix of a new rule is **pre1**: and the key separator is a comma (,), and the prefix of the existing rule is **pre1** and the separator is a colon (:), the new rule cannot be created.
- If the table name of a MySQL instance in the mapping rule is changed, you need to reconfigure the mapping rule.
- Currently, the ENUM, SET, and JSON data cannot be synchronized.
- Renaming or deleting one or more fields in the key field of a mapping rule renders the rule invalid.

Creating a mapping rule

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the navigation pane on the left, choose **Memory Acceleration**. In the **Mapping Rule** area, click **Create Mapping Rule**.

Step 5 On the displayed page, configure required parameters.

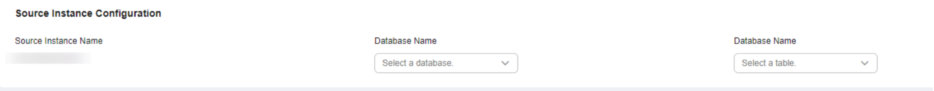
1. Set the rule name.

   Rule Name: Set the name of the mapping rule. The rule name must be unique within a GeminiDB instance and cannot exceed 256 characters or include number signs (#).

2. Configure source instance information.

- **Database Name**: Select the database of the instance to be accelerated.
- **Table Name**: Select the table in the acceleration instance.

Figure 3-100 Configuring source instance information



Source Instance Configuration

Source Instance Name [redacted]

Database Name: Select a database.

Database Name: Select a table.

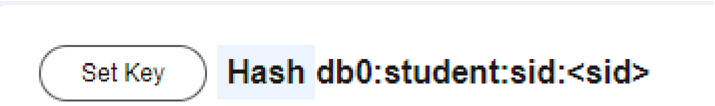
3. Configure acceleration instance information.
   - **Redis Key Prefix**: This parameter is optional. The default format is database name:data table name:field name 1:field name 2.... It can contain up to 1024 characters. If you have created a custom prefix, it will take precedence over the default one.
   - **Value Storage Type**: Data type of the cache. Currently, only hashes are supported.
   - **Database No. (0-999)**: ID of the database that stores cached data in the acceleration instance. The default value is **0**.
   - **TTL (s) Default value: 30 days** : Expiration time of cached data in the acceleration instance. The default value is 30 days (2,592,000 seconds). If you enter **-1**, the cached data will not expire.
   - **Key Delimiter**: Separator among the Redis key prefix, key, and key fields. It is a single character in length.
4. Click **Set Key**, select the key field of the acceleration instance, and click **OK**.

   NOTE

   If an acceleration instance key consists of multiple source instance fields, the key must be unique (a unique index must be created for these fields in a MySQL instance). You can click **Up** or **Down** to adjust the sequence of each field in the key.

   After the setting is complete, the key is displayed.

   Figure 3-101 Key structure

   Set Key Hash db0:student:sid:<sid>

5. Configure the domain-value of the acceleration instance.

   Select fields required in the source instance and copy them to the fields of the acceleration instance.
6. After setting the parameters, click **Submit**.

----End

Example

1. Create database **db1** in the source MySQL instance and create table **students** in db1. The SQL statements are as follows:

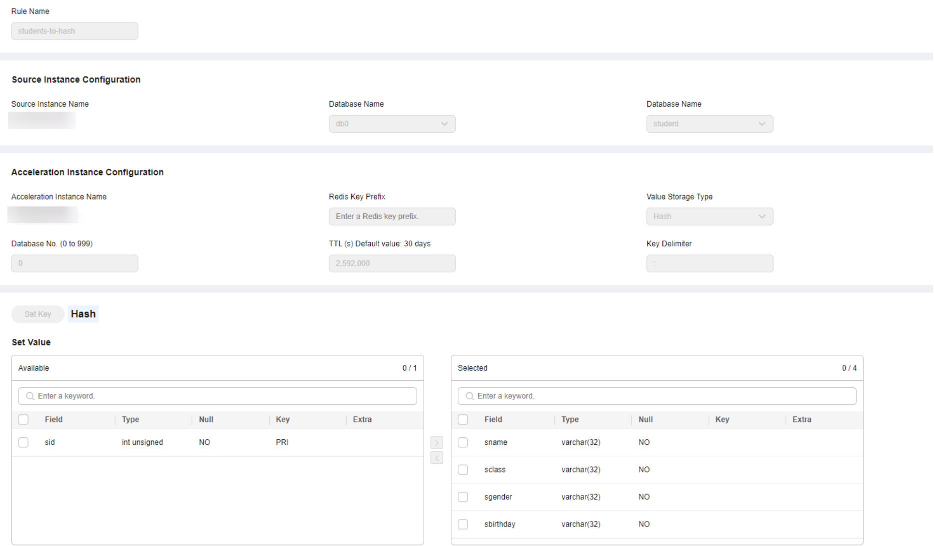
```
mysql> CREATE DATABASE db1;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE TABLE db1.students(
    sid INT UNSIGNED PRIMARY KEY AUTO_INCREMENT NOT NULL,
    sname VARCHAR(20),
    sclass INT,
    sgender VARCHAR(10),
    sbirthday DATE
    );
Query OK, 0 rows affected (0.00 sec)

mysql> DESC db1.students;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| sid       | int unsigned | NO   | PRI | NULL    | auto_increment |
| sname     | varchar(20)  | YES  |     | NULL    |                |
| sclass    | int          | YES  |     | NULL    |                |
| sgender   | varchar(10)  | YES  |     | NULL    |                |
| sbirthday | date         | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

2. After the table is created, on the memory acceleration page, create a mapping rule to convert each row in the students table into a Redis hash. The key of a hash consists of database name:data table name:sid:<sid value>. The fields are **sname**, **sclass**, **sgender**, and **sbirthday**.

Figure 3-102 Configuring mapping rules



Rule Name: students-to-hash

Source Instance Configuration

Source Instance Name: [redacted]

Database Name: db0

Database Name: student

Acceleration Instance Configuration

Acceleration Instance Name: [redacted]

Redis Key Prefix: Enter a Redis key prefix.

Value Storage Type: Hash

Database No. (0 to 999): 0

TTL (s) Default value: 30 days: 2,592,000

Key Delimiter: :

Set Key Hash

Set Value

Available 0 / 1

Enter a keyword.

	Field	Type	Null	Key	Extra
□	sid	int unsigned	NO	PRI	

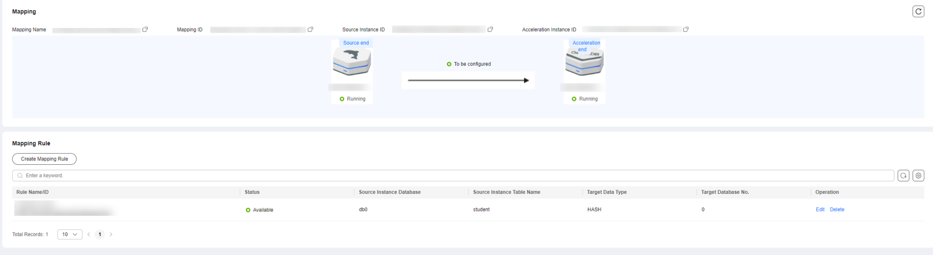
Selected 0 / 4

Enter a keyword.

	Field	Type	Null	Key	Extra
□	sname	varchar(32)	NO		
□	sclass	varchar(32)	NO		
□	sgender	varchar(32)	NO		
□	sbirthday	varchar(32)	NO		

3. After a mapping rule is created, check the mapping rule and mapping information.

Figure 3-103 Mapping information



Mapping

Mapping Name [redacted] Mapping ID [redacted] Source Instance ID [redacted] Acceleration Instance ID [redacted]

Mapping Rule

Create Mapping Rule

Enter a keyword.

Rule Name/ID	Status	Source Instance Database	Source Instance Table Name	Target Data Type	Target Database No.	Operation
[redacted]	Available	db0	student	HASH	0	Edit Delete

Total Records: 1 10 < 1 >

4. Insert a new data record to the **students** table in the MySQL instance.

```
mysql> INSERT INTO db1.students (sname, sclass, sgender, sbirthday) VALUES ('zhangsan', 1, 'male',
'2015-05-20');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM db1.students;
+-----+----------+--------+---------+------------+
| sid | sname    | sclass | sgender | sbirthday  |
+-----+----------+--------+---------+------------+
|   1 | zhangsan |      1 | male    | 2015-05-20 |
+-----+----------+--------+---------+------------+
1 row in set (0.00 sec)
```

5. After the mapping rule is created, the data is automatically synchronized to the GeminiDB instance. Run commands in the GeminiDB instance to query the data.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:1"

127.0.0.1:6379> HGETALL db1:students:sid:1
1) "sbirthday"
2) "2015-05-20"
3) "sclass"
4) "1"
5) "sgender"
6) "male"
7) "sname"
8) "zhangsan"
```

6. Insert a new data record to the **students** table in the MySQL instance.

```
mysql> INSERT INTO db1.students (sname, sclass, sgender, sbirthday) VALUES ('lisi', 10, 'male',
'2015-05-22');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM db1.students;
+-----+----------+--------+---------+------------+
| sid | sname    | sclass | sgender | sbirthday  |
+-----+----------+--------+---------+------------+
|   1 | zhangsan |      1 | male    | 2015-05-20 |
|   2 | lisi     |     10 | male    | 2015-05-22 |
+-----+----------+--------+---------+------------+
2 rows in set (0.00 sec)
```

7. New data will be synchronized to the GeminiDB instance.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:1"
2) "db1:students:sid:2"

127.0.0.1:6379> HGETALL db1:students:sid:2
1) "sbirthday"
2) "2015-05-22"
3) "sclass"
4) "10"
5) "sgender"
6) "male"
7) "sname"
```

8. Update data in the **students** table in the MySQL instance.

```
mysql> UPDATE db1.students SET sclass=12, sname='wangwu' WHERE sid = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM db1.students;
+-----+--------+--------+---------+------------+
| sid | sname  | sclass | sgender | sbirthday  |
+-----+--------+--------+---------+------------+
|   1 | wangwu |     12 | male    | 2015-05-20 |
|   2 | lisi   |     10 | male    | 2015-05-22 |
+-----+--------+--------+---------+------------+
2 rows in set (0.00 sec)
```

9. Data in the GeminiDB instance is updated.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:1"
2) "db1:students:sid:2"

127.0.0.1:6379> HGETALL db1:students:sid:1
1) "sbirthday"
2) "2015-05-20"
3) "sclass"
4) "12"
5) "sgender"
6) "male"
7) "sname"
8) "wangwu"
```

10. Delete data from the **students** table in the MySQL instance.

```
mysql> DELETE FROM db1.students WHERE sid = 1;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM db1.students;
+-----+-------+--------+---------+------------+
| sid | sname | sclass | sgender | sbirthday  |
+-----+-------+--------+---------+------------+
|   2 | lisi  |     10 | male    | 2015-05-22 |
+-----+-------+--------+---------+------------+
1 row in set (0.00 sec)
```

11. The data is deleted from the GeminiDB instance.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:2"
```

Modifying a Mapping Rule

Step 1 Log in to the management console.

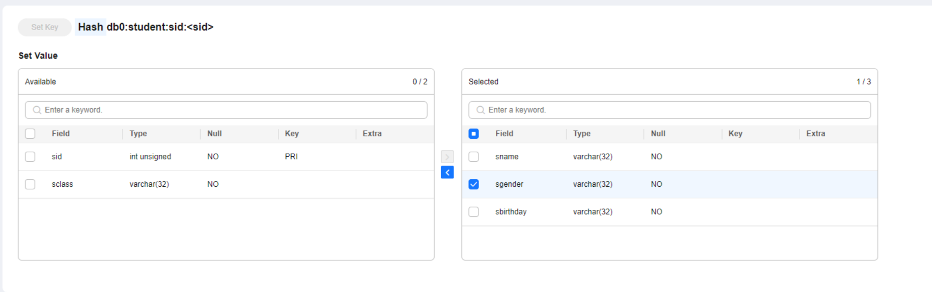
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the navigation pane on the left, choose **Memory Acceleration**. In the **Mapping Rule** area, locate the target rule and click **Edit** in the **Operation** column.

Step 5 After editing the fields, click **Submit**.

Figure 3-104 Editing a mapping rule



Set Key Hash db0:student:sid:<sid>

Set Value

Available 0 / 2

Enter a keyword.

□	Field	Type	Null	Key	Extra
□	sid	int unsigned	NO	PRI	
□	sclass	varchar(32)	NO		

Selected 1 / 3

Enter a keyword.

■	Field	Type	Null	Key	Extra
□	sname	varchar(32)	NO		
☑	sgender	varchar(32)	NO		
□	sbirthday	varchar(32)	NO		

----End

Deleting a Mapping Rule

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance.

Step 4 In the navigation pane on the left, choose **Memory Acceleration**. In the **Mapping Rule** area, locate the target rule and click **Delete** in the **Operation** column.

----End

3.12.4 Memory Acceleration Management

You can view the mapping list on the **Memory Acceleration Management** page and remove mappings.

Precautions

- After a mapping is removed, service applications cannot obtain the latest data in the source database through the acceleration instance. A free GeminiDB instance will be re-billed once the mapping is removed.
- The corresponding mapping rule will be cleared after a mapping is removed.
- If the source instance or acceleration instance is not normal, the mapping cannot be removed.

Querying the Mapping List

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 In the navigation pane on the left, choose **Memory Acceleration Management**. On the displayed page, search for your target mapping by keyword (such as the mapping name or mapping ID).

----End

Unmapping

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 In the navigation pane on the left, choose **Memory Acceleration Management**. On the displayed page, locate the target mapping and click **Remove** in the **Operation**.

Step 4 In the displayed dialog box, click **OK**.

----End

4 Data Migration

4.1 Overview of the Redis Data Migration Solution

This section describes how to migrate services to GeminiDB Redis. If you have any questions about the migration, submit a service ticket to obtain technical support.

Migration Tool

- Data Replication Service (**DRS**): supports full data migration and incremental data migration and provides secure, stable, and reliable migration links.
- **Redis-Shake** tool: is an open-source migration tool that supports migration modes such as full scanning (rump), data restoration (restore), and incremental synchronization (sync). You can download the tool to an ECS and use CLI to facilitate migration.

Required Permissions

- Ensure that the database port is enabled in the security group of the GeminiDB Redis instance.

Migration Scenarios

Table 4-1 Migration scenarios

No.	Source	Destination	Migration Solution
1	Alibaba Cloud Redis/Tair	GeminiDB Redis	**Migrating the Alibaba Cloud Database Redis/Tair To GeminiDB Redis**
3	Self-built Redis/ Codis	GeminiDB Redis	**From On-Premises Redis to GeminiDB Redis API**
4	RDB	GeminiDB Redis	**Restoring RDB Files to GeminiDB Redis API (Recommended)**

5	Self-built Kvrocks	GeminiDB Redis	**From Kvrocks to GeminiDB Redis API**
6	Self-built Pika	GeminiDB Redis	**From Pika to GeminiDB Redis API**
7	Self-built SSDB	GeminiDB Redis	**From SSDB to GeminiDB Redis API**
8	Self-built LevelDB	GeminiDB Redis	**From LevelDB to GeminiDB Redis API**
9	Self-built RocksDB	GeminiDB Redis	**From Kvrocks to GeminiDB Redis API**
10	AWS ElasticCache for Redis	GeminiDB Redis	**Migrating AWS Elastic Cache for Redis Databases To GeminiDB Redis**

4.2 Verifying Redis Data Consistency After Migration

After the migration is complete, you can check the consistency of Redis data.

Constraints

- The Redis migration has been completed, or the incremental migration has started.
- The Redis-Full-Check open-source tool must be deployed on the ECS instance, and the ECS instance can communicate with the source and target networks.
- If the migration task is in the incremental state, data consistency cannot be ensured due to network latency between the source and target ends. If conditions permit, you are advised to stop writing data to the source end and then perform the verification.
- When Redis is used, an expiration time is usually set for keys. During migration, setting a key expiration time affects data consistency. If verification results show that data is inconsistent, the possible cause is that the key expiration time is inconsistent.
- During the migration, DTS writes temporary probing keys to Redis on the destination end. Non-service data may be detected during data verification, which is normal.

Procedure

Step 1 Log in to the ECS and ensure that the ECS can connect to the source and destination Redis databases.

Step 2 Deploy **Redis-Full-Check**.

Step 3 Verify data.

```
/redis-full-check -s {Source IP address}:{Source port} -p {Source password} -t
{Destination IP address}:{Destination port} -a {Destination password} -m 1
```

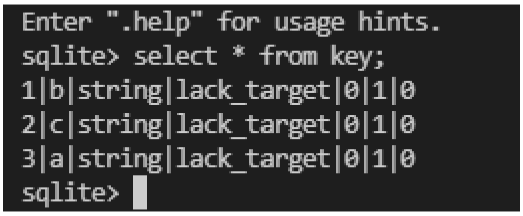
Table 4-2 Parameter description

Parameter	Description	Example Value
-s	Source Redis connection address and port number	-s 10.0.0.1:6379
-p	Password of the source Redis database.	-
-t	Destination Redis connection address and port number	-t 10.0.0.2:6379
-a	Password of the destination Redis database.	-
-m	Verification mode: 1. Verify all key-value pairs. 2. Only value length is verified. 3. Only key integrity is verified. 4. All key values are verified, but only the length of big keys is verified. By default, the second verification mode is used.	-m 1
-q	Maximum QPS. The default value is **15000**.	-q 5000
-d	Name of the file for saving the verification result. The default value **is result.db**.	-d result.db

Step 4 View the verification result file.

By default, three rounds of verification are performed and three verification result files are generated. Generally, you only need to view the last verification result file.

- Run the **sqlite3 result.db.3** command.
- Run the **select * from key** command.
- Check whether abnormal keys exist.



```
Enter ".help" for usage hints.
sqlite> select * from key;
1|b|string|lack_target|0|1|0
2|c|string|lack_target|0|1|0
3|a|string|lack_target|0|1|0
sqlite>
```

----End

4.3 Migrating the Alibaba Cloud Database Redis/Tair To GeminiDB Redis

This section describes how to migrate Alibaba Cloud databases Redis or Tair to GeminiDB Redis.

Migration Principles

- The data migration function of the Alibaba Cloud data migration tool DTS is used to migrate data from Alibaba Cloud Redis to other Redis services. This tool avoids the restrictions of shielding the sync and psync commands of Alibaba Cloud Redis and migrates data from Alibaba Cloud Redis to Huawei Cloud GeminiDB Redis.

Precautions

- The source end on Alibaba Cloud needs to communicate with the destination end on Huawei Cloud. Ensure that a private line is enabled or that binding a public IP address is performed.
- The Alibaba Cloud DTS data migration function is charged in real time. Before using this function, ensure that your Alibaba Cloud account balance is sufficient.
- The Huawei Cloud GeminiDB Redis capacity must be greater than or equal to the memory capacity of the Alibaba Cloud Redis database.
- Ensure that the security group configuration on the source and target ends is enabled.
- Some Redis databases on Alibaba Cloud are special. For example, Tair hybrid storage does not support online full and incremental migration. You can complete the migration by scanning all the data.

Preparations

- Migrating data using a public IP address
  - Purchase a Huawei Cloud EIP in advance. The bandwidth must be greater than the source database traffic.
  - Bind the EIP to a Huawei Cloud GeminiDB Redis node.
  - When configuring DTS, ensure that the destination database is connected through a public IP address.
- Migrating data using a private line
  - Purchase an Alibaba ECS in advance to ensure that it can connect to Huawei Cloud GeminiDB Redis.

- Configure data forwarding to forward the traffic received by the local port to the destination end, implementing migration from Alibaba Cloud Redis to GeminiDB Redis.

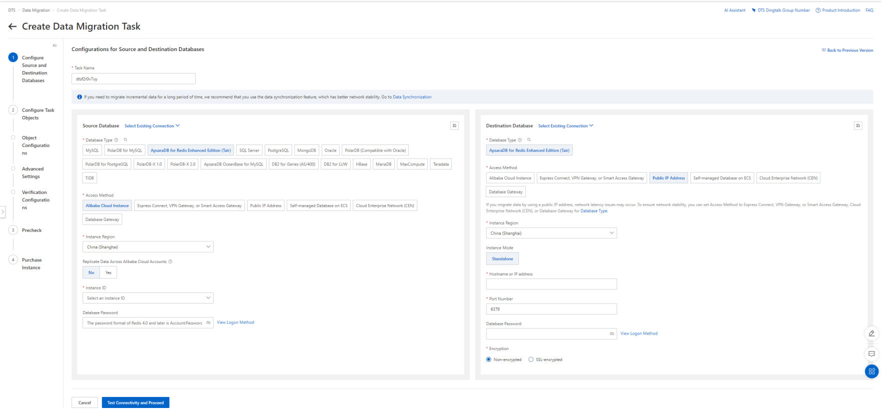
  ssh -g -L (*Forwarding port***): (***LB IP address of Huawei Redis***): (***Huawei Redis port***) -N -f root@ (***Local ECS IP Address***)**

- When configuring DTS, ensure that the destination database is connected through a self-built ECS database.

Purchasing the DTS Data Synchronization Function

Step 1 Select the Redis service on Alibaba Cloud as the source end. If an EIP is used for migration, select a public IP address as the destination end and enter the EIP as the host name. If Direct Connect is used for migration, select the self-built Redis database on ECS as the destination end, set the host name to the IP address of the ECS, set the port number to the forwarding port number, enter the database password, and click the test link. If no exception occurs during the test, the next page is displayed. Otherwise, check whether the entire link is normal and whether the whitelist configuration is correct.

Figure 4-1 Source and destination configuration information



Create Data Migration Task

Configurations for Source and Destination Databases

* Task Name: dtsf2r0v7uy

If you need to migrate incremental data for a long period of time, we recommend that you use the data synchronization feature, which has better network stability. Go to Data Synchronization

Source Database

* Database Type: ApsaraDB for Redis Enhanced Edition (Tair)

* Access Method: Alibaba Cloud Instance

* Instance Region: China (Shanghai)

Replicate Data Across Alibaba Cloud Accounts: No

* Instance ID: Select an instance ID

Database Password: The password format of Redis 4.0 and later is Account:Passworc

Destination Database

* Database Type: ApsaraDB for Redis Enhanced Edition (Tair)

* Access Method: Public IP Address

If you migrate data by using a public IP address, network latency issues may occur. To ensure network stability, you can set Access Method to Express Connect, VPN Gateway, or Smart Access Gateway, Cloud Enterprise Network (CEN), or Database Gateway for Database Type.

* Instance Region: China (Shanghai)

Instance Mode: Standalone

* Hostname or IP address:

* Port Number: 6379

Database Password:

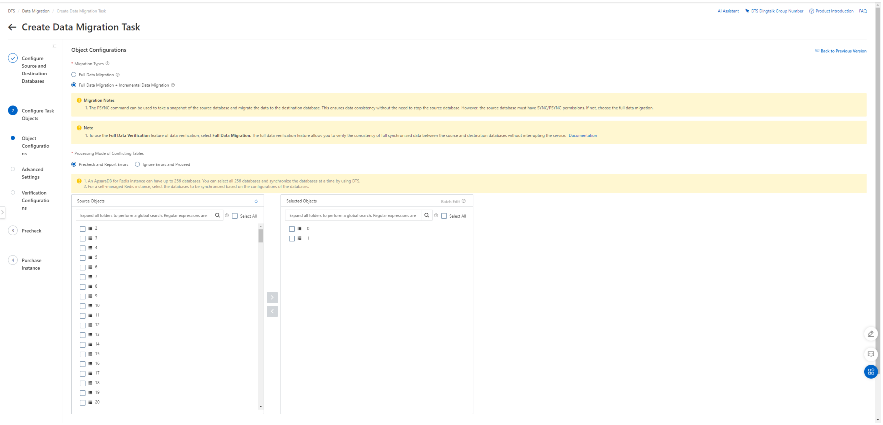
* Encryption: ● Non-encrypted ○ SSL-encrypted

DTS / Data Migration / Create Data Migration Task
AI Assistant
DTS Dingtalk Group Number
Product Introduction
FAQ
Back to Previous Version
1 Configure Source and Destination Databases
2 Configure Task Objects
Object Configurations
Advanced Settings
Verification Configurations
3 Precheck
4 Purchase Instance
Select Existing Connection
MySQL
PolarDB for MySQL
ApsaraDB for Redis Enhanced Edition (Tair)
SQL Server
PostgreSQL
MongoDB
Oracle
PolarDB (Compatible with Oracle)
PolarDB for PostgreSQL
PolarDB-X 1.0
PolarDB-X 2.0
ApsaraDB OceanBase for MySQL
DB2 for iSeries (AS/400)
DB2 for LUW
HBase
MariaDB
MaxCompute
Teradata
TiDB
Alibaba Cloud Instance
Express Connect, VPN Gateway, or Smart Access Gateway
Public IP Address
Self-managed Database on ECS
Cloud Enterprise Network (CEN)
Database Gateway
No
Yes
View Logon Method
Cancel
Test Connectivity and Proceed

Step 2 Select **Full Data Migration** or **Full Data Migration + Incremental Data Migration**. Select **Pre-check and Report Errors** and select the database to be migrated.

> ⚠ CAUTION
>
> If you use the multi-DB function, select all databases to be migrated. If the multi-DB function is not used, select only **DB0**.

Figure 4-2 Database to be migrated



← Create Data Migration Task

Object Configurations

* Migration Types

○ Full Data Migration

● Full Data Migration + Incremental Data Migration

Migration Notes

1. The PSYNC command can be used to take a snapshot of the source database and migrate the data to the destination database. This ensures data consistency without the need to stop the source database. However, the source database must have SYNC/PSYNC permissions. If not, choose the full data migration.

Note

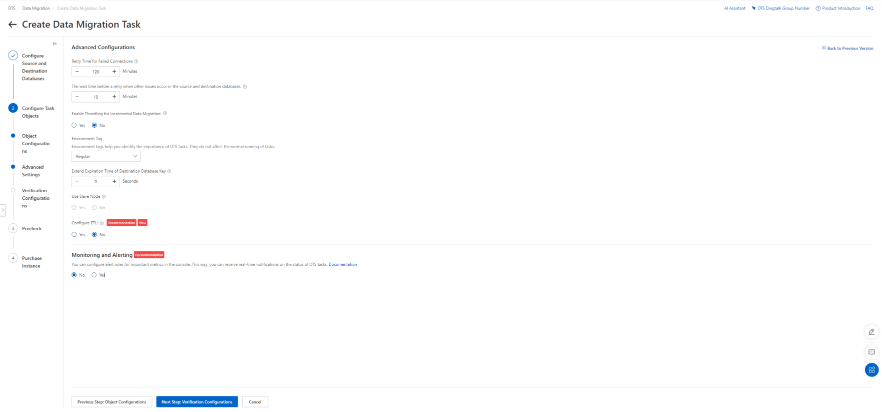
1. To use the **Full Data Verification** feature of data verification, select **Full Data Migration**. The full data verification feature allows you to verify the consistency of full synchronized data between the source and destination databases without interrupting the service. Documentation

* Processing Mode of Conflicting Tables

● Precheck and Report Errors ○ Ignore Errors and Proceed

1. An ApsaraDB for Redis instance can have up to 256 databases. You can select all 256 databases and synchronize the databases at a time by using DTS.
2. For a self-managed Redis instance, select the databases to be synchronized based on the configurations of the databases.

Figure 4-3 Configure advanced settings.



DTS / Data Migration / Create Data Migration Task
AI Assistant
DTS Dingtalk Group Number
Product Introduction
FAQ
Configure Source and Destination Databases
Configure Task Objects
Object Configurations
Advanced Settings
Verification Configurations
Precheck
Purchase Instance
Back to Previous Version
Recommendation
New
Previous Step: Object Configurations
Next Step: Verification Configurations
Cancel

← Create Data Migration Task

Advanced Configurations

Retry Time for Failed Connections: 120 Minutes

The wait time before a retry when other issues occur in the source and destination databases.: 10 Minutes

Enable Throttling for Incremental Data Migration: ○ Yes ● No

Environment Tag

Environment tags help you identify the importance of DTS tasks. They do not affect the normal running of tasks.

Regular

Extend Expiration Time of Destination Database Key: 0 Seconds

Use Slave Node: ○ Yes ○ No

Configure ETL: ○ Yes ● No

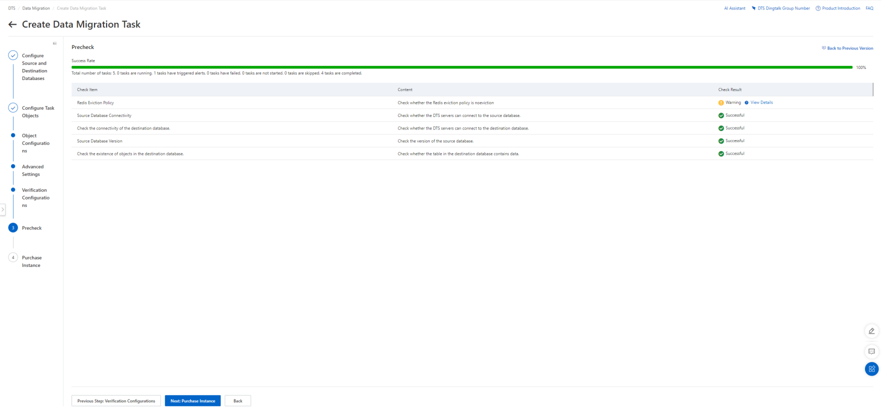
Monitoring and Alerting

You can configure alert rules for important metrics in the console. This way, you can receive real-time notifications on the status of DTS tasks. Documentation

● No ○ Yes

Step 3 After the pre-check is complete, click **Next: Purchase Instance**.

Figure 4-4 Pre-check



DTS / Data Migration / Create Data Migration Task
AI Assistant
DTS Dingtalk Group Number
Product Introduction
FAQ
Configure Source and Destination Databases
Configure Task Objects
Object Configurations
Advanced Settings
Verification Configurations
Precheck
Purchase Instance
Back to Previous Version
View Details
Previous Step: Verification Configurations
Next: Purchase Instance
Back

← Create Data Migration Task

Precheck

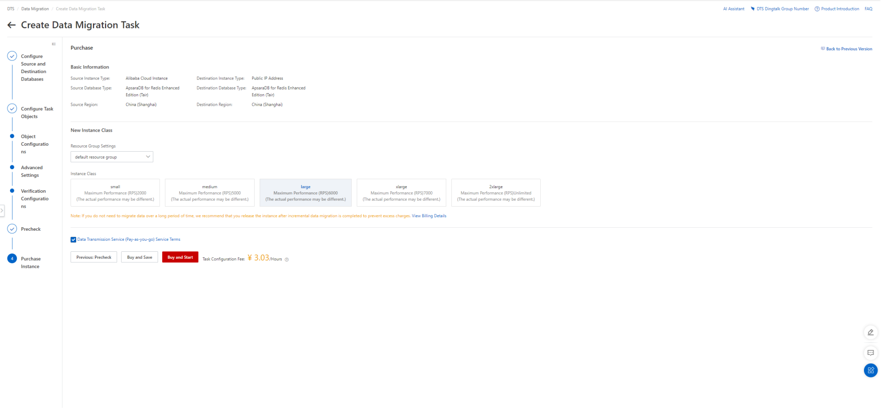
Success Rate 100%

Total number of tasks: 5. 0 tasks are running. 1 tasks have triggered alerts. 0 tasks have failed. 0 tasks are not started. 0 tasks are skipped. 4 tasks are completed.

Check Item	Content	Check Result
Redis Eviction Policy	Check whether the Redis eviction policy is noeviction	Warning
Source Database Connectivity	Check whether the DTS servers can connect to the source database.	Successful
Check the connectivity of the destination database.	Check whether the DTS servers can connect to the destination database.	Successful
Source Database Version	Check the version of the source database.	Successful
Check the existence of objects in the destination database.	Check whether the table in the destination database contains data.	Successful

Step 4 Select the bandwidth for the migration and click **Buy and Start**.

Figure 4-5 Bandwidth configuration



DTS / Data Migration / Create Data Migration Task

AI Assistant DTS Dingtalk Group Number Product Introduction FAQ

← Create Data Migration Task

- Configure Source and Destination Databases
- Configure Task Objects
- Object Configurations
- Advanced Settings
- Verification Configurations
- Precheck
- 4 Purchase Instance

Purchase

Back to Previous Version

Basic Information

Source Instance Type:	Alibaba Cloud Instance	Destination Instance Type:	Public IP Address
Source Database Type:	ApsaraDB for Redis Enhanced Edition (Tair)	Destination Database Type:	ApsaraDB for Redis Enhanced Edition (Tair)
Source Region:	China (Shanghai)	Destination Region:	China (Shanghai)

New Instance Class

Resource Group Settings: default resource group

Instance Class

- small: Maximum Performance (RPS)2000 (The actual performance may be different.)
- medium: Maximum Performance (RPS)5000 (The actual performance may be different.)
- large: Maximum Performance (RPS)6000 (The actual performance may be different.)
- xlarge: Maximum Performance (RPS)7000 (The actual performance may be different.)
- 2xlarge: Maximum Performance (RPS)Unlimited (The actual performance may be different.)

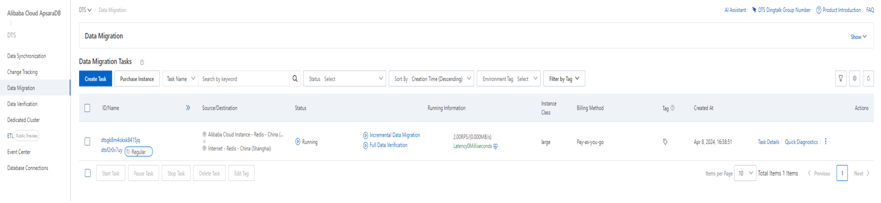
Note: If you do not need to migrate data over a long period of time, we recommend that you release the instance after incremental data migration is completed to prevent excess charges. View Billing Details

☑ Data Transmission Service (Pay-as-you-go) Service Terms

Previous: Precheck | Buy and Save | Buy and Start | Task Configuration Fee: ¥ 3.03/Hours

Step 5 If **Full Data Migration + Incremental Data Migration** is selected, the migration task will not automatically end. If there is no delay (0 ms), the full synchronization is complete.

Figure 4-6 Task status



Alibaba Cloud ApsaraDB > DTS: Data Synchronization, Change Tracking, Data Migration, Data Verification, Dedicated Cluster, ETL Public Preview, Event Center, Database Connections

DTS / Data Migration

AI Assistant DTS Dingtalk Group Number Product Introduction FAQ

Data Migration Show

Data Migration Tasks

Create Task | Purchase Instance | Task Name | Search by keyword | Status Select | Sort By Creation Time (Descending) | Environment Tag Select | Filter by Tag

ID/Name	Source/Destination	Status	Running Information		Instance Class	Billing Method	Tag	Created At	Actions
dtsgk8m4okxk8415jq dtsf2r0v7uy Regular	Alibaba Cloud Instance - Redis - China (... Internet - Redis - China (Shanghai)	Running	Incremental Data Migration Full Data Verification	2.00RPS/(0.000MB/s) Latency0Milliseconds	large	Pay-as-you-go		Apr 8, 2024, 16:38:51	Task Details | Quick Diagnostics

Start Task | Pause Task | Stop Task | Delete Task | Edit Tag

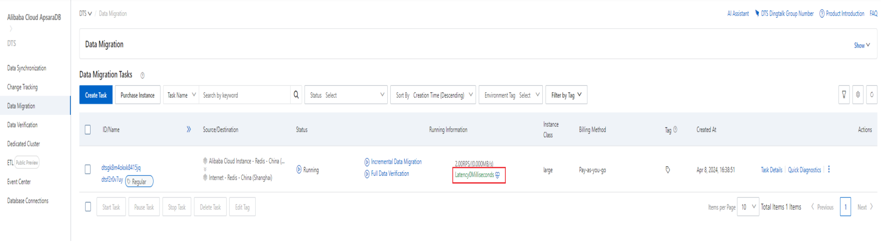
Items per Page 10 | Total Items 1 Items | Previous 1 Next

----End

Stopping the DTS Data Synchronization Service

Step 1 After the Redis service migration, stop the data synchronization task.

Figure 4-7 Stopping the data migration task



Alibaba Cloud ApsaraDB > DTS: Data Synchronization, Change Tracking, Data Migration, Data Verification, Dedicated Cluster, ETL Public Preview, Event Center, Database Connections

DTS / Data Migration

AI Assistant DTS Dingtalk Group Number Product Introduction FAQ

Data Migration Show

Data Migration Tasks

Create Task | Purchase Instance | Task Name | Search by keyword | Status Select | Sort By Creation Time (Descending) | Environment Tag Select | Filter by Tag

ID/Name	Source/Destination	Status	Running Information		Instance Class	Billing Method	Tag	Created At	Actions
dtsgk8m4okxk8415jq dtsf2r0v7uy Regular	Alibaba Cloud Instance - Redis - China (... Internet - Redis - China (Shanghai)	Running	Incremental Data Migration Full Data Verification	2.00RPS/(0.000MB/s) Latency0Milliseconds	large	Pay-as-you-go		Apr 8, 2024, 16:38:51	Task Details | Quick Diagnostics

Start Task | Pause Task | Stop Task | Delete Task | Edit Tag

Items per Page 10 | Total Items 1 Items | Previous 1 Next

----End

4.4 From On-Premises Redis to GeminiDB Redis API

You can use DRS or Redis-Shake to migrate data from self-built Redis to GeminiDB Redis. This section describes how to use Redis-Shake to migrate data from an on-premises Redis database to a GeminiDB Redis instance.

Migration Principles

Use Redis-Shake to migrate data from an on-premises Redis instance (source) to a GeminiDB Redis instance (destination). Full and incremental migrations are both supported. The source can be a single-node, primary/standby, or cluster instance, or an RDB file.

- Full migration: Redis-Shake works as a slave node for the source, obtains data of an RDB file generated by the source, and then parses the data and sends it to the destination by running commands. You can also use an RDB file as the source to import snapshot data generated at a specific time point.
- Incremental migration: After full migration is complete, Redis-Shake continues sending incremental data to the destination by running commands until you stop Redis-Shake.

Precautions

- If data synchronization between master and slave Redis nodes is disconnected, stop Redis-Shake, clear all data in the destination, and retry a migration. To ensure a smooth synchronization, migrate data during off-peak hours and set a large value for parameter **client-output-buffer-limit** to increase the ring buffer size for incremental synchronization.
- Redis-Shake does not write data into the source, but may have a temporary impact on the source performance.
- If the migration involves multiple databases, ensure that source databases are correctly mapped to destination databases to prevent unexpected data overwriting.
- Streaming data cannot be migrated.
- Ensure that network communication among Redis-Shake, the source instance, and the destination instance is normal.
- To get technical support, choose **Service Tickets** > **Create Service Ticket** in the upper right corner of the console and contact customer service.

Migrating Data from an Open-Source Redis Single-Node or Primary/Standby Instance to a GeminiDB Redis Instance

You can import a file similar to the above or perform the following operations to migrate data from an open-source single-node or primary/standby Redis instance to a GeminiDB Redis instance.

Step 1 Deploy the required migration tool.

1. Obtain the **Redis-Shake package**.

NOTE

Download the Redis-Shake release package and decompress it.

2. Modify the **Redis-Shake.conf** configuration file and configuring the following items:

   log.level = info #Default log level. A printed INFO log contains migration progress information, based on which you can judge whether the migration is complete.

   source.address = <host>:<port> # IP address and port of the host where the open-source Redis instance is deployed.

   source.password_raw = ******* #** Password for logging in to the source instance.

   source.type = standalone # Type of the source instance.

   target.address = <host>:8635 # Address for logging in to the destination instance.

   target.password_raw = ******* #** Password for logging in to the destination instance.

   target.version = 5.0 # Version of the destination Redis instance.

   target.type = **standalone** # Type of the destination instance.

   target.db = -1 # Specific database on the destination that all data will be migrated to. If this parameter is set to **-1**, a mapping relationship is established between migrated databases and databases in the source instance.

3. Specify whether data of the destination is overwritten.

   key_exists = none

   NOTE

   If there are duplicate keys on the source and destination, specify whether data of the destination is overwritten. The options are as follows:

   - **rewrite** indicates that the source overwrites the destination.
   - **none** indicates that the migration process exists once duplicate keys are detected.
   - **ignore** indicates that keys in the source are retained and keys in the destination are ignored. This value does not take effect in rump mode.

   none is recommended. There will be no duplicate data because the source is an RDB file. If the migration process exits unexpectedly, contact customer service.

Step 2 Migrate data.

Migration starting command:

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

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

```
sync rdb done
```

- If the following information is displayed, no new data is incremented. You can stop the migration process to disconnect incremental synchronization:

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

Step 3 Verify data.

Download and decompress **RedisFullCheck** and use it to verify data by referring to **Migrating Data from an Open-Source Redis Single-Node or Primary/Standby Instance to a GeminiDB Redis Instance**.

```
./redis-full-check -s SOURCE_IP:SOURCE_PORT -p SOURCE_PWD -t TARGET_IP:8635 -a TARGET_PWD
```

If the following information is displayed, the migration is successful, and data is consistent between the source and destination:

```
all finish successfully, totally 0 key(s) and 0 field(s) conflict
```

----End

Migrating Data from an Open-Source Redis Cluster Instance to a GeminiDB Redis Instance

Configure the following items in the configuration file:

source.address = <host1>:<port1>,<host2>:<port2>,<host2>:<port2> # IP addresses and ports of hosts at the source.

source.type = cluster # Cluster type of the source.

For other steps, see **Migrating Data from an Open-Source Redis Single-Node or Primary/Standby Instance to a GeminiDB Redis Instance**.

Migrating Data from an Open-Source Codis Cluster Instance to a GeminiDB Redis Instance

Obtain host IP addresses and ports of all shards of the Codis cluster instance and configure the configuration file as follows:

source.address = <host1>:<port1>,<host2>:<port2>,<host2>:<port2> # IP addresses and ports of hosts at the source.

source.type = cluster # Cluster type of the source.

For other steps, see **Migrating Data from an Open-Source Redis Single-Node or Primary/Standby Instance to a GeminiDB Redis Instance**.

Fully Scanning Data on and Migrating It from an Open-Source Redis Instance to a GeminiDB Redis Instance

If data cannot be migrated with any of the above methods, try rump of Redis-Shake to scan databases one by one and migrate them.

Step 1 Deploy the required migration tool.

1. Obtain the **Redis-Shake package**.

   NOTE

   Download the Redis-Shake release package and decompress it.

2. Modify the **Redis-Shake.conf** configuration file and configuring the following items:

log.level = info #Default log level. A printed INFO log contains migration progress information, based on which you can judge whether the migration is complete.

source.address = <host>:<port> # IP address and port of the host where the open-source Redis instance is deployed.

source.password_raw = *** #** Password for logging in to the source instance.

source.type = standalone # Type of the source instance.

target.address = <host>:8635 # Address for logging in to the destination instance.

target.password_raw = *** #** Password for logging in to the destination instance.

target.version = 5.0 # Version of the destination Redis instance.

target.type = standalone # Type of the destination instance.

target.db = -1 # Specific database on the destination that all data will be migrated to. If this parameter is set to **-1**, a mapping relationship is established between migrated databases and databases in the source instance.

3. Specify whether data of the destination is overwritten.

key_exists = none

NOTE

If there are duplicate keys on the source and destination, specify whether data of the destination is overwritten. The options are as follows:

- **rewrite** indicates that the source overwrites the destination.
- **none** indicates that the migration process exists once duplicate keys are detected.
- **ignore** indicates that keys in the source are retained and keys in the destination are ignored. This value does not take effect in rump mode.

none is recommended. There will be no duplicate data because the source is an RDB file. If the migration process exits unexpectedly, contact customer service.

Step 2 Migrate data.

Migration starting command:

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

- If information similar to the following is displayed, synchronizing full data is complete.

```
dbRumper[0] executor[0] finish
```

Step 3 Verify data.

Download and decompress **RedisFullCheck** and use it to verify data.

```
./redis-full-check -s SOURCE_IP:SOURCE_PORT -p SOURCE_PWD -t TARGET_IP:8635 -a TARGET_PWD
```

If the following information is displayed, the migration is successful, and data is consistent between the source and destination:

```
all finish successfully, totally 0 key(s) and 0 field(s) conflict
```

----End

4.5 Migration from an RDB to a GeminiDB Redis Instance Using a Migration Tool

Importing an RDB File to a GeminiDB Redis Instance

Step 1 Deploy the required migration tool.

1. Obtain the **redis-shake package**.

   NOTE

   Download the Redis-Shake release package and decompress it.

2. Modify the **Redis-Shake.conf** configuration file and configuring the following items:

   log.level = info #Default log level. A printed INFO log contains migration progress information, based on which you can judge whether the migration is complete.

   source.rdb.input = /xx/xx.rdb # Absolute path of the source RDB file.

   target.address = <host>:6379 # Address for logging in to the destination instance.

   target.password_raw = ***** # Password for logging in to the destination.

   target.version = **5.0** # Version of the destination Redis instance.

   target.type = **standalone** # Type of the destination instance.

   target.db = **-1 #** Specific database on the destination that all data will be migrated to. If this parameter is set to **-1**, a mapping relationship is established between migrated databases and databases in the source instance.

   target.dbmap = #Configure the database migration mapping. The value of **target.db** must be **-1**, for example, **0-5**. **1-3** indicates that data in source database **db0** will be written to destination database **db5** and data in source database **db1** will be written to destination database **db3**.

   big_key_threshold = 52428800 # Big key threshold. If the number of value bytes corresponding to a key exceeds the threshold, data is written in batches.

   resume_from_break_point = **false** #Disable resumable download. This function is unavailable.

3. Specify whether data of the destination is overwritten.

   key_exists = none

   NOTE

   If there are duplicate keys on the source and destination, specify whether data of the destination is overwritten. The options are as follows:

   - **rewrite** indicates that the source overwrites the destination.
   - **none** indicates that the migration process exists once duplicate keys are detected.
   - **ignore** indicates that keys in the source are retained and keys in the destination are ignored. This value does not take effect in rump mode.

   none is recommended. There will be no duplicate data because the source is an RDB file. If the migration process exits unexpectedly, contact customer service.

Step 2 Migrate data.

Migration starting command:

./redis-shake.linux -conf=redis-shake.conf -type=restore

NOTE

Use the restore mode because the source is an RDB file.

Stop the migration process after the migration is complete.

Step 3 Verify data.

Data is obtained from the RDB file. Therefore, you need to check the GeminiDB Redis data at the destination end from the service perspective.

----End

Importing the AOF File To GeminiDB Redis

Step 1 Upload the generated AOF file to an ECS.

Step 2 Start the open-source Redis 5.0 single-node process on the ECS to load the AOF file and wait till the process is started. Ensure that the startup directory of the open-source Redis is the same as the directory containing the AOF file.

Step 3 Run the SAVE command to generate an RDB file. Place the RDB file in the startup directory of the open-source Redis.

Step 4 Stop the open-source Redis 5.0 process.

Step 5 Perform the migration by following **Importing an RDB File to a GeminiDB Redis Instance**.

----End

4.6 Restoring RDB Files to GeminiDB Redis API (Recommended)

Scenarios

Redis data of other vendors or self-hosted Redis can be migrated to GeminiDB Redis API.

You need to download the source Redis data, then upload the data to an OBS bucket in the same region as the GeminiDB Redis instance, and create a data import task on the GeminiDB console to import the data to the GeminiDB Redis instance.

Precautions

- Importing data will overwrite data of the current database.
- Importing backups generated by a later-version Redis instance to an earlier one may fail.

- Before importing backups, ensure that resource-intensive commands (such as FLUSHALL, KEYS, and HGETALL) have been disabled on the target Redis instance.
- If a backup contains multi-DB data, its database count cannot exceed what is supported by the target Redis instance.
- Only .rdb files can be imported.

Creating an OBS Bucket and Uploading Backups

Perform the following steps if the backup is smaller than 5 GB:

Step 1 Create an OBS bucket.

When creating an OBS bucket, configure the following parameters as requested.

1. **Region**:

   The OBS bucket must be in the same region as the destination Redis instance.
2. **Storage Class**: Available options are **Standard**, **Infrequent Access**, and **Archive**.

   Do not select **Archive**. Otherwise, the backup may fail to be imported.
3. Click **Create Now**.

Step 2 In the bucket list, click the bucket created in **Step 1**.

Step 3 In the navigation pane, choose **Objects**.

Step 4 On the **Objects** tab page, click **Upload Object**.

Step 5 Specify **Storage Class**.

Do not select **Archive**. Otherwise, the backup may fail to be imported.

Step 6 Upload the objects.

Drag files or folders to the **Upload Object** area or click **add file**.

A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.

Step 7 (Optional) Select **KMS encryption** to encrypt the uploaded files.

Step 8 Click **Upload**.

----End

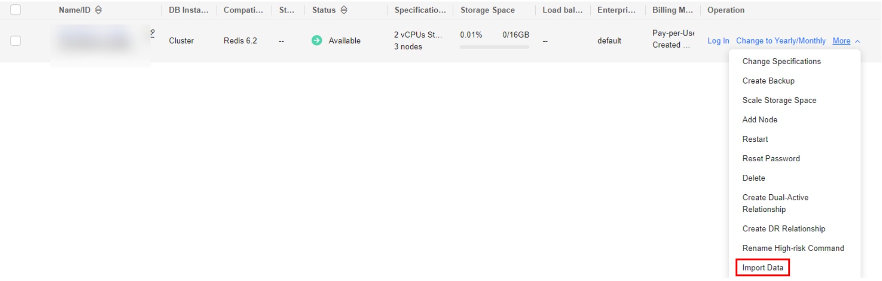
Importing Backups

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, locate the target instance and choose **More** > **Import Data** in the **Operation** column.

Figure 4-8 Importing data



□	Name/ID	DB Insta...	Compati...	St...	Status	Specificatio...	Storage Space	Load bal...	Enterpri...	Billing M...	Operation
□	[redacted]	Cluster	Redis 6.2	--	Available	2 vCPUs St... 3 nodes	0.01% 0/16GB	--	default	Pay-per-Use Created ...	Log In Change to Yearly/Monthly More

Step 4 On the **Data Import** page, specify **OBS Bucket** to which a backup have been uploaded.

Step 5 Click **Add Backup** and select the backups to be imported.

- A maximum of 128 backups can be added at a time.
- To delete a backup, locate the target backup and click **Delete** in the **Operation** column.
- To delete all backups, select **Clear** for **Backup**.

Step 6 Click **Create Now**.

Step 7 Confirm the data import and click **OK**.

⚠ CAUTION

Importing data will overwrite data of the current database.

----End

4.7 From Kvrocks to GeminiDB Redis API

Kvrocks is an open-source NoSQL key-value database that is compatible with the Redis ecosystem. It uses namespace to partition data based on the underlying RocksDB. However, it is relatively weak in cluster management. Kvrocks needs to cooperate with other components to create clusters and does not support some Redis commands, such as stream and hyperloglog that are frequently used in message flow and statistics scenarios.

GeminiDB Redis API is a cloud-native NoSQL database with decoupled compute and storage and full compatibility with Redis. To ensure data security and reliability, it provides multi-copy, strict consistency based on a shared storage pool. It provides high compatibility, cost-effectiveness, high reliability, elastic scalability, high availability, and hitless scale-out. GeminiDB Redis API functions as good as Redis Cluster does and is 100% compatible with native APIs. You can migrate your on-premises Redis databases to GeminiDB Redis ones without modifying any code. In addition to adapting to the Kvrocks service, it can also overcome the disadvantages of weak management capability and low compatibility with Redis.

This section describes how to migrate data from Kvrocks to GeminiDB Redis API.

Migration Principles

The open-source tool kvrocks2redis is used to migrate data from Kvrocks to GeminiDB Redis API. At the code layer, Kvrocks namespace is adapted to the source GeminiDB Redis database.

The migration process consists of two phases: full migration and incremental migration. During full migration that is first performed, snapshots are created for Kvrocks and the corresponding data version (seq) is recorded. Then, the complete data files are parsed into Redis commands and written to GeminiDB Redis API. After the full migration is complete, the incremental migration starts. The migration tool cyclically sends PSYNC commands to Kvrocks and continuously forwards the obtained incremental data to GeminiDB Redis API.

Precautions

- Kvrocks2redis needs to extract data from Kvrocks to local files, parse commands from the files, and send the commands to the target GeminiDB Redis instance. During this process, the performance of the source DB may be affected, but no data is compromised theoretically.
- If a fault occurs when the migration tool is running, the migration tool automatically stops to facilitate fault locating.
- For security purposes, GeminiDB Redis API does not provide database clearing commands. Ensure that no data exists in the database before the migration.

Prerequisites

- Deploy the kvrocks2redis on an independent host.
- Ensure that the source DB, target DB, and migration tool can communicate with each other.
- Back up data of the source Kvrocks instance in advance.
- Clear all data on the destination GeminiDB Redis instance.

Procedure

To migrate data from Kvrocks to GeminiDB Redis API, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console and contact customer service.

4.8 From Pika to GeminiDB Redis API

Pika is a persistent large-capacity Redis storage service. It breaks through the memory bottleneck of Redis due to the large amount of data. However, it is relatively weak in cluster management, and requires twemproxy or codis to shard static data. Compared with the Redis community edition, the database performance is significantly lowered because Pika stores all data in disks.

GeminiDB Redis API is a cloud-native NoSQL database with decoupled compute and storage and full compatibility with Redis. To ensure data security and reliability, it provides multi-copy, strict consistency based on a shared storage pool.

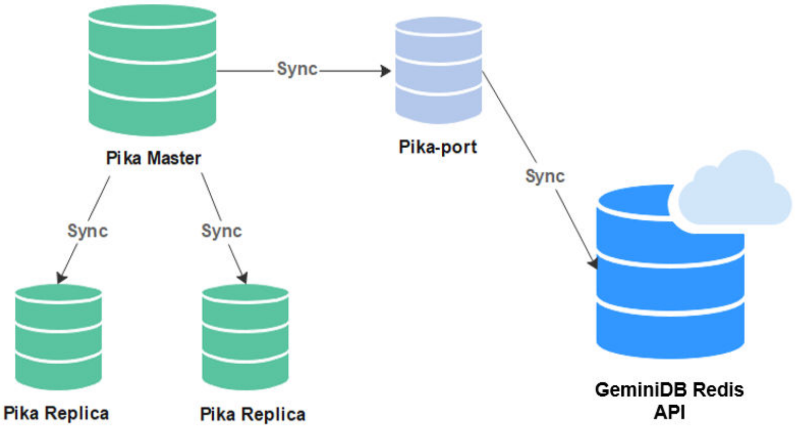
It supports cold and hot data separation. Hot data can be read from the cache directly, improving read efficiency. RocksDB has been customized to allow the storage capacity to be scaled in seconds. A proxy is used to ensure that upper-layer applications are not affected by underlying sharding or scaling.

This section describes how to migrate data from Pika to GeminiDB Redis API.

Migration Principles

The pika-port tool is used and acts as a slave node of Pika and data is migrated in master/slave replication mode. The master Pika node compares pika-port with its own binlog offset to determine whether to perform full migration or incremental migration. If full migration is required, the master Pika node sends the full data snapshot to pika-port, and pika-port sends the parsed snapshot data to GeminiDB Redis API. After the full migration is complete, incremental migration starts. pika-port parses the incremental data and sends the data to GeminiDB Redis API in the form of Redis commands.

Figure 4-9 Migration Principles



Precautions

- pika-migrate and pika-port act as the slave node of the source Pika and reads only full and incremental data without damaging your data.
- The master/slave synchronization process between the source DB and pika-migrate and pika-port is added, which may affect the performance of the source DB.
- Full and incremental migration can be performed without service interruption. Services are interrupted for a short period of time when services are switched over to GeminiDB Redis API.

Migration Performance Reference

- Environment: Pika (single node) and pika-port are deployed on an ECS with 8 vCPUs and 32 GB memory on Huawei Cloud. The target DB is a three-node GeminiDB Redis instance with 8 vCPUs and 16 GB memory.
- Preset data: Use the memtier_benchmark tool to preset 200 GB of data.
- Migration performance: about 50,000 QPS.

4.9 From SSDB to GeminiDB Redis API

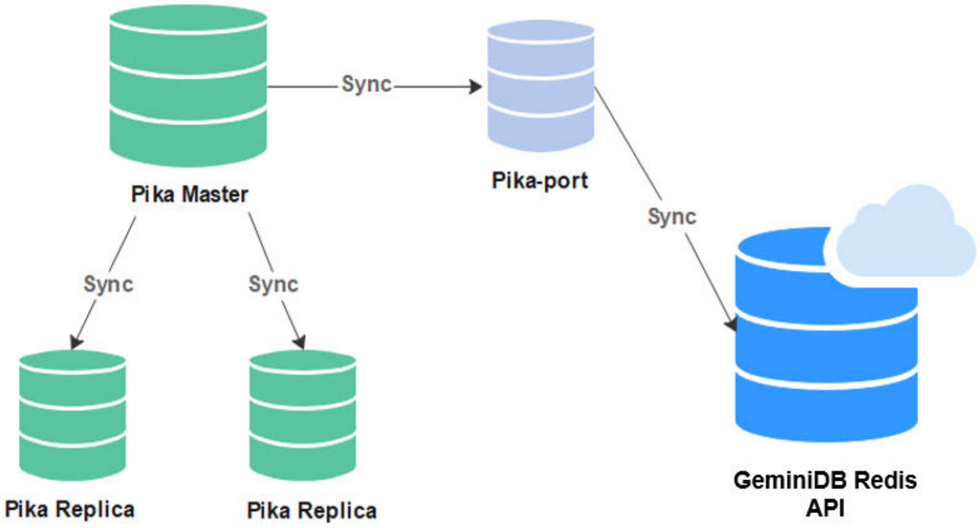
SSDB is a high-performance NoSQL database written in C/C++. It is compatible with Redis APIs and supports multiple data structures, including key-value pairs, hashmap, sorted set, and list. SSDB is a persistent KV storage system and uses leveldb as the underlying storage engine. Its services directly interact with LevelDB. Operations such as compaction have direct impact on service read and write. GeminiDB Redis API is a cloud-native NoSQL database with decoupled compute and storage and full compatibility with Redis. To ensure data security and reliability, it provides multi-copy, strict consistency based on a shared storage pool. RocksDB is used as the storage engine. Compared with leveldb, RocksDB greatly improves performance, solves the problem that leveldb proactively restricts write, and implements cold and hot separation, reducing the impact of operations at the storage layer on performance.

This section describes how to migrate data from SSDB to GeminiDB Redis API.

Migration Principles

ssdb-port acts as a slave node (replica) of the master node of the source SSDB database and migrates data through master/slave replication. Then, it parses and converts the obtained data into the format supported by Redis, and sends the data to the Redis instance specified in the configuration file. The following figure shows the migration process. After the full synchronization is complete, the new data in SSDB is also synchronized to the Redis instance.

Figure 4-10 Migration diagram



Precautions

- As the slave node of the SSDB master node, ssdb-port reads only full and incremental data without damaging your data.
- The performance of the source SSDB is affected for running ssdb-port.

- Full migration and incremental migration can be performed without service interruption. After all data is migrated, services need to be stopped for a short period of time.

Prerequisites

Create an ECS in the VPC where the GeminiDB Redis instance is located and deploy the migration tool ssdb-port to ensure that the source SSDB instance can communicate with the target GeminiDB Redis instance.

Procedure

To migrate data from SSDB to GeminiDB Redis API, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console and contact customer service.

Migration Performance Reference

- Environment: The source SSDB and ssdb-port are deployed on an ECS with 4 vCPUs and 16 GB memory. The destination is a three-node instance with 8 vCPUs and 16 GB memory.
- Preset data: Use the memtier_benchmark tool to preset 100 GB of data.
- Migration performance: about 3000 QPS.

4.10 From LevelDB to GeminiDB Redis API

LevelDB is an open-source, persistent, and single-node KV database engine. It provides high random write performance and sequential read/write performance, and applies to write intensive scenarios. LevelDB does not provide the C/S network structure and must be deployed on the same server as your services. Compared with RocksDB developed based on LevelDB, LevelDB has many disadvantages. For example, it cannot make the most out of multi-core servers, and does not support TB-level data storage, and cannot read data from HDFS.

GeminiDB Redis API uses RocksDB as the storage engine. It is compatible with the Redis protocol and provides various data types to meet LevelDB requirements. In addition, RocksDB has been customized to allow storage to be scaled in seconds, making it easy to migrate LevelDB workloads to the Redis ecosystem. You do not need to migrate data during scaling.

This section describes how to migrate data from LevelDB to GeminiDB Redis API.

Migration Principles

- Use the self-developed migration tool leveldb-port to deploy LevelDB on the same server as your services, prepare the configuration file, and start the migration task to automatically complete full and incremental migration.
- The full migration process is efficient. It takes a snapshot of the LevelDB data, scans the entire database, packs the data into a format that can be identified by GeminiDB Redis API, and then sends the data to GeminiDB Redis API.
- During incremental migration, the WAL file of LevelDB and the LevelDB operations are parsed, and the keys in the WAL file are sharded and sent by multiple threads.

Precautions

- The migration tool needs to be deployed on the source DB, which consumes certain performance. You can modify the configuration file to control the performance.
- During the migration, the source data file of LevelDB is read-only. There is no risk of data damage.
- Services do not need to be stopped during the migration.
- If a fault occurs during the migration, clear the GeminiDB Redis instance and restart the migration.

Procedure

To migrate data from LevelDB to GeminiDB Redis API, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console and contact customer service.

Migration Performance Reference

- Environment: The source LevelDB and leveldb-port are deployed on a Huawei Cloud ECS with 4 vCPUs and 16 GB memory. The target DB is a three-node GeminiDB Redis instance with 2 vCPUs and 8 GB memory.
- Full migration: 10 GB data is preconfigured, and the migration speed is about 8 MB/s.
- Incremental migration: Set the value to 1 KB and the migration speed to 7,000 QPS.

4.11 From Kvrocks to GeminiDB Redis API

RocksDB is a persistent key-value store, single-node DB engine developed by Facebook based on LevelDB. It has powerful sequential read/write and random write performance. Compared with LevelDB, RocksDB has many optimizations. Its performance is greatly improved and the problem that LevelDB proactively restricts write operations is solved. As a DB engine, RocksDB does not provide the C/S network structure. It must be deployed on the same server as your services.

GeminiDB Redis API uses RocksDB as the storage engine and is compatible with the Redis protocol, meeting the usage requirements of RocksDB. In addition, RocksDB has been customized to allow storage to be scaled in seconds, making it easy to migrate RocksDB workloads to the Redis ecosystem. You do not need to migrate data during scaling.

This section describes how to migrate data from RocksDB to GeminiDB Redis API.

Migration Principles

- Use the self-developed migration tool rocksdb-port to deploy RocksDB on the same server as your services, prepare the configuration file, and start the migration task to automatically complete full and incremental migration.
- The full migration process is efficient. It takes a snapshot of the RocksDB data, scans the entire database, packs the data into a format that can be identified by GeminiDB Redis API, and then sends the data to GeminiDB Redis API.

- During incremental migration, the WAL file of RocksDB and the RocksDB operations are parsed, and the keys in the WAL file are sharded and sent by multiple threads.

Precautions

- The migration tool needs to be deployed on the source DB, which consumes certain performance. You can modify the configuration file to control the performance.
- During the migration, the source data file of RocksDB is read-only. There is no risk of data damage.
- Services do not need to be stopped during the migration.
- If a fault occurs during the migration, clear the GeminiDB Redis instance and restart the migration.

Procedure

To migrate data from RocksDB to GeminiDB Redis API, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console and contact customer service.

4.12 Migrating AWS Elastic Cache for Redis Databases To GeminiDB Redis

Migration Principles

After backing up and exporting the RDB file in AWS ElasticCache for Redis, you can use the Redis-Shake to restore data to GeminiDB Redis.

Precautions

- AWS does not support the **psync/sync** command and data cannot be incrementally migrated.
- Before the migration, ensure that the network between the ECS where Redis-shake is deployed and the destination GeminiDB Redis is normal.
- Ensure that the security group configuration on the source and target ends is enabled.

Procedure

Step 1 Deploy the required migration tool.

1. Obtain the **redis-shake package**.

   NOTE

   Download the Redis-Shake release package and decompress it.

2. Modify the **Redis-Shake.conf** configuration file and configuring the following items:

log.level = info #Default log level. A printed INFO log contains migration progress information, based on which you can judge whether the migration is complete.

source.rdb.input = /xx/xx.rdb # Absolute path of the source RDB file.

target.address = <host>:8635 # Address for logging in to the destination instance.

target.password_raw = ***** # Password for logging in to the destination.

target.version = 5.0 # Version of the destination Redis instance.

target.type = standalone # Type of the destination instance.

target.db = 0 #Data is migrated to the specified database of the destination GeminiDB Redis. The default value is **db0**.

big_key_threshold = 1 #Setting the big key threshold

3. Specify whether data of the destination is overwritten.

   key_exists = none

NOTE

If there are duplicate keys on the source and destination, specify whether data of the destination is overwritten. The options are as follows:

- **rewrite** indicates that the source overwrites the destination.
- **none** indicates that the migration process exists once duplicate keys are detected.
- **ignore** indicates that keys in the source are retained and keys in the destination are ignored. This value does not take effect in rump mode.

none is recommended. There will be no duplicate data because the source is an RDB file. If the migration process exits unexpectedly, contact customer service.

Step 2 Migrate data.

Migration starting command:

./redis-shake.linux -conf=redis-shake.conf -type=restore

NOTE

Use the restore mode because the source is an RDB file.

Stop the migration process after the migration is complete.

Step 3 Verify data.

Data is obtained from the RDB file. Therefore, you need to check the GeminiDB Redis data at the destination end from the service perspective.

----End

5 FAQs

5.1 Most Asked Questions

Product Consulting

- What Are the Differences Between GeminiDB Redis API, Open-Source Redis, and Other Open-Source Redis Cloud Services?
- How Is the Performance of GeminiDB Redis API Compared with Open-Source Redis?
- What Redis Versions and Commands Are Compatible with GeminiDB Redis API? Whether Applications Need to Be Changed for Client Connection?
- Can Data Be Migrated from a Self-Built Redis Instance to a GeminiDB Redis Instance? What Are the Precautions?
- Are Total Memory and Total Capacity of a GeminiDB Redis Instance the Same? What Is the Relationship Between Memory and Capacity?
- How Do I Select Proper Node Specifications and Node Quantity When Purchasing a GeminiDB Redis Instance?
- How Does GeminiDB Redis API Persist Data? Will Data Be Lost?
- What Is the Memory Eviction Policy of GeminiDB Redis API?
- Does GeminiDB Redis API Support Modules Such as a Bloom Filter?

Database Connection

- How Do I Connect to a GeminiDB Redis Instance?
- What Can I Do with IP Addresses of GeminiDB Redis Nodes?
- How Does Load Balancing Work in GeminiDB Redis API?
- Can I Change the VPC of a GeminiDB Redis Instance?
- How Do I Access a GeminiDB Redis Instance from a Private Network?
- Do I Need to Enable Private Network Access Control for a Load Balancer After Setting a Security Group?

Database Usage

Backup and Restoration

5.2 About GeminiDB Redis API

5.2.1 What Are the Differences Between GeminiDB Redis API, Open-Source Redis, and Other Open-Source Redis Cloud Services?

Redis, an open-source in-memory data structure store, is used as a cache broker. GeminiDB Redis API, an enhanced version of open-source Redis, is an elastic KV database compatible with the Redis protocol, supports much larger capacity than memory, and delivers ultimate performance. Hot data is stored in memory, and full data is stored in a high-performance storage pool. GeminiDB Redis API features:

- Low stable latency

  The average single-point read/write latency is shorter than 1 ms, and the P99 latency is shorter than 2 ms. By adopting a multi-thread architecture, GeminiDB Redis API allows for flexible QPS adjustment ranging from 10,000 to 10,000,000.

- High cost-effectiveness

  30% lower comprehensive costs: Because no standby node is required and GeminiDB Redis API offers an ultra-high data compression ratio of 4:1, it is cheaper to scale out storage capacity.

- Higher O&M efficiency

  2 GB to 100 TB more capacity can be added to storage devices without any impact on services. Point-in-Time Recovery (PITR) restores databases up to a specific moment in time.

- More enhanced features for enterprises

  An expiration time can be specified for individual fields in a hash. A Bloom filter can be used. Data can be imported extremely fast. Memory acceleration can be enabled.

For details about the comparison between GeminiDB Redis and open-source self-built KV databases, see .

5.2.2 How Is the Performance of GeminiDB Redis API Compared with Open-Source Redis?

By adopting a multi-thread architecture, GeminiDB Redis API allows for flexible adjustment of QPS ranging from 10,000 to 10,000,000 based on the CPU quantity. Low average single-point read/write latency (< 1 ms) and p99 latency (< 2 ms) on a single instance of GeminiDB Redis API are similar to those of open-source Redis..

5.2.3 What Redis Versions and Commands Are Compatible with GeminiDB Redis API? Whether Applications Need to Be Changed for Client Connection?

GeminiDB Redis API is fully compatible with Redis 6.2 (including 6.2.*x*). Data can be migrated from Redis 6.2 and earlier (such as 5.0, 4.0, and 2.8) to GeminiDB Redis API without the need of changing applications. Any Redis client can access GeminiDB Redis API.

5.2.4 Can Data Be Migrated from a Self-Built Redis Instance to a GeminiDB Redis Instance? What Are the Precautions?

Yes. Before migration, ensure that:

- Version: If the version of the source database is 6.2 or earlier (including 6.2.*x*), data can be directly migrated. If the version of the source database is later than 6.2, you need to evaluate the migration project and then migrate data to GeminiDB Redis 6.2. You can submit a service ticket for consultation.
- Specifications: Configure proper specifications based on QPS and data volumes of the source instance.

5.2.5 What Is the Availability of a GeminiDB Redis Instance?

The formula for calculating the instance availability is as follows:

DB instance availability = (1 – Failure duration/Total service duration) × 100%

The failure duration refers to the total duration of faults that occur during the running of a DB instance after you buy the instance. The total service duration refers to the total running time of the DB instance.

5.2.6 Are Total Memory and Total Capacity of a GeminiDB Redis Instance the Same? What Is the Relationship Between Memory and Capacity?

No. In an open-source Redis instance, all data is stored in memory, and the total capacity is the amount of memory that can be utilized. In a GeminiDB Redis instance, all data is stored in a high-performance shared storage pool, and hot data is stored in memory. Generally, you only need to pay attention to the total capacity and usage of the instance. The CPU usage increases as QPS increases. In this case, you need to scale up the specifications.

5.2.7 How Do I Select Proper Node Specifications and Node Quantity When Purchasing a GeminiDB Redis Instance?

When purchasing a GeminiDB Redis instance, pay attention to QPS and data volume. You can select **Fast configure** or **Standard configure** for **Instance Creation Method**.

- **Fast configure**: If 16 GB of storage space is used for a cluster, you can select **16 GB** in the **Instance Specifications** area. If QPS does not meet requirements, select higher specifications.
- **Standard configure**: Select specifications of compute and storage resources separately. The node specifications and number of nodes determine instance QPS, and the total instance capacity determines the maximum storage space. After selecting the node specifications, number of nodes, and total instance capacity, you can view the QPS and number of connections of the selected instance next to **Specification Preview**.

5.2.8 How Does GeminiDB Redis API Persist Data? Will Data Be Lost?

Open-Source Redis persists data periodically, so there is a high probability that data loss occurs in abnormal scenarios. GeminiDB Redis API data is updated to a storage pool in real time, improving data security. Similar to other NoSQL databases, backend processes on the GeminiDB Redis API instance write write-ahead logs (WALs) into OS buffers. The buffers immediately return and then are updated to the storage pool. Therefore, a small amount of data may be lost in the case of an unexpected power failure. GeminiDB Redis API ensures data is not lost during routine O&M, such as changing specifications, upgrading versions, and adding nodes.

5.2.9 What Is the Memory Eviction Policy of GeminiDB Redis API?

If keys of an open-source Redis instance are evicted from the memory, the key values cannot be read later. By default, GeminiDB Redis API supports a noeviction policy, that is, user keys are not evicted. All data is stored in a storage pool. Hot data evicted from the memory can be read from the storage pool. The data is reloaded to the memory after being accessed, and user keys are not deleted. Therefore, GeminiDB Redis API users do not need to set or modify the **maxmemory-policy** parameter. If unnecessary data is stored, users need to add an expiration time to avoid dramatical increase in data volumes.

5.2.10 Does GeminiDB Redis API Support Modules Such as a Bloom Filter?

Yes. In addition, you can set an expiration time for individual fields in a hash shard. Shards can be scanned in parallel. Data can be imported extremely fast using FastLoad.

5.3 Billing

5.3.1 What Are the Differences Between Yearly/Monthly and Pay-per-use Billing Mode?

Yearly/Monthly is a prepaid billing mode in which resources are billed based on the service duration. This cost-effective mode is ideal when the duration of resource usage is predictable. It is recommended for long-term users.

Pay-per-use is a post payment mode, so you can start or stop an instance at any time. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.

5.3.2 Can I Switch Between Yearly/Monthly and Pay-per-Use Payments?

You can change the billing mode from yearly/monthly to pay-per-use or vice versa.

- For details about how to change the billing mode from yearly/monthly to a pay-per-use, see **Changing the Billing Mode from Yearly/Monthly to Pay-per-Use**.
- For details about how to change the billing mode from pay-per-use to yearly/monthly, see **Changing the Billing Mode from Pay-per-Use to Yearly/Monthly**.

5.4 Database Usage

5.4.1 Why Is the Key Not Returned Using Scan Match?

Symptom

As shown in the following figure, the value of key is **test** and exists in the database. However, no data is returned using this scan match command.

```
139.9.177.148: 8635> scan 1 match tes*
1) "21"
```

```
2) (empty list or set)
139.9.177.148: 8635> get test
"abc"
139.9.177.148:8635>scan 0 match tes*
1) "21"
2) (empty list or set)
139.9.177.148: 8635>
```

Possible Causes

The MATCH command is used to iterate elements that only match a specified pattern. Pattern matching is performed after the command obtain elements from the data set and before the elements are returned to the client. If all the extracted elements do not match the pattern, no element is returned.

Solution

If multiple scans are performed, the iteration is complete when the returned cursor is 0. The cursor returned from the last scan is used for the next scan.

5.4.2 How Do I Process Existing Data Shards After Migrating Workloads to GeminiDB Redis API?

GeminiDB Redis API uses decoupled compute and storage and allows adding data shards dynamically, making scaling smooth. After an GeminiDB Redis instance is connected, data sharding is not required on the service side.

5.4.3 Does GeminiDB Redis API Support Fuzzy Query Using the Keys Command?

Yes.

An OOM error or high latency may occur when you query data using the KEYS command. You can use the KEYS command for service tests, but not for production. You can use SCAN and MATCH commands for fuzzy match.

5.4.4 Does the GeminiDB Redis API Support Multiple Databases?

GeminiDB Redis API allows you to create multiple databases in an instance since March 2022. Instances created before March 2022 do not support this function and cannot be upgraded to support it.

This feature has the following constraints:

- The number of databases ranges from 0 to 999.
- The SWAPDB command is not supported.
- The result of the **dbsize** command is not updated in real time. The result does not decrease to 0 immediately after **flushdb** is executed, and will change to 0 after a while.
- Executing SELECT and FLUSHDB commands in LUA scripts is not supported.
- Executing SELECT and FLUSHDB commands in transactions is not supported.

- The MOVE command is not supported.

5.4.5 Why the Values Returned by Scan Operations Are Different Between GeminiDB Redis API and Open-Source Redis 5.0?

GeminiDB Redis API may return values in a different sequence from open-source Redis, but they both comply with open-source document description requirements. This is because open-source Redis does not specify the sorting rules for:

- Returned values of SCAN/HSCAN/SSCAN operations
- Returned values of ZSCAN operations ZSET when its elements have the same score

5.4.6 Why Are Error Messages Returned by Some Invalid Commands Different Between GeminiDB Redis API and Open-Source Redis 5.0?

GeminiDB Redis API checks command syntax and checks for keys each time it executes a command. However, open-source Redis has no specific rules and returns the results for invalid commands in random. Therefore, error messages returned by some invalid commands may be different.

5.4.7 How Do I Resolve the Error "CROSSSLOT Keys in request don't hash to the same slot"?

Scenarios

When multi-key commands are executed in a GeminiDB Redis instance, the error "CROSSSLOT Keys in request don't hash to the same slot" may be reported.

Error Cause

Commands involving multiple keys were executed across slots in a GeminiDB Redis cluster instance. For example, EVAL and BRPOPLPUSH were executed across slots.

Solution

- Change key names and use hash tags to ensure that the involved keys are in the same slot. Avoid data skew when you use hash tags. For more information, see **Which Commands Require Hash Tags in GeminiDB Redis Cluster Instances?**.
- When hash tags cannot be used, change the instance type to primary/standby. For details, see **Compatible APIs and Versions**.

5.4.8 How Many Commands Can Be Contained in a GeminiDB Redis Transaction?

It is recommended that a transaction contain a maximum of 100 commands. Exercise caution when using commands with time complexity of O(N).

5.4.9 Which Commands Require Hash Tags in GeminiDB Redis Cluster Instances?

In a GeminiDB Redis cluster instance, if you need to run the following commands that manage multiple keys, use hash tags to when designing key names:

MSETNX, BLPOP, BRPOP, BRPOPLPUSH, RPOPLPUSH, SDIFF, SDIFFSDIFFSTORE, SINTER, SINTERSTORE, SMOVE, SUNION, SUNIONSUNIONSTORE, ZINTERSTORE, ZUNIONSTORE, XREAD, XREADGROUP, PFCOUNT, PFMERGE, GEORADIUS, GEORADIUS_RO, GEORADIUSBYMEMBER, GEORADIUSBYMEMBER_RO, GEOSEARCHSTORE, BITOP, RENAME, RENAMENX, and SORT.

5.4.10 What Do I Do If the Error "ERR Unknown Command Sentinel" Is Displayed?

Scenarios

When **SENTINEL** commands are executed on a GeminiDB Redis instance, the error message "ERR unknown command sentinel" may be displayed.

Error Cause

If the value of **CompatibleMode** of a GeminiDB Redis cluster instance is not **3** or the value of **CompatibleMode** of a primary/standby GeminiDB Redis instance is not **2**, executing **SENTINEL** commands is not allowed.

Solution

Step 1 **Log in to the GeminiDB console.**

Step 2 In the service list, choose **Databases** > **GeminiDB**.

Step 3 On the **Instances** page, click the instance whose specifications you want to change. The **Basic Information** page is displayed.

Step 4 In the navigation pane on the left, choose **Parameters**.

Step 5 Change the value of **CompatibleMode** and click **Save**.

- For a cluster DB instance, change the value of **CompatibleMode** to **3**.
- For a primary/standby DB instance, change the value of **CompatibleMode** to **2**.

----End

5.4.11 How Long Does It Take to Add GeminiDB Redis Nodes at the Same Time? What Are the Impacts on Services?

GeminiDB Redis nodes can be added at the same time, which can be completed within 5 minutes. Shared storage is used. After nodes are added, data does not need to be migrated, but slots are rebalanced. A retry mechanism is needed to avoid service interruptions due to a few seconds of jitter or latency.

5.4.12 What Are the Differences Between Online and Offline Specification Changes of GeminiDB Redis Nodes? How Long Will the Changes Take? What Are the Impacts on Services?

- Online change: Nodes are changed in rolling mode. The change duration is positively related to the number of nodes. Each node takes about 5 to 10 minutes. In addition, both primary/standby and cluster instances contain three internal management nodes, which are changed at the same time. For example, a GeminiDB Redis instance consists of six nodes, including three worker nodes and three internal management nodes. The online change takes about 30 to 60 minutes. During specification changes of a single node, services are affected due to a few seconds of jitter. Therefore, a reconnection mechanism is required. You are advised to change the node specifications during off-peak hours and keep the CPU and memory usage at a low level. This prevents exceptions such as heavy load on other nodes and process startup failures.
- Offline change: Specifications of all nodes are changed concurrently. During the change, services are interrupted for about 10 to 20 minutes. Offline change is applicable when services are stopped or no service is accessed. Exercise caution when performing this operation.

For your online production services, you are advised to perform the change online. For details, see **Changing the CPU and Memory Specifications of an Instance**.

5.4.13 Can I Download Backups of a GeminiDB Redis Instance to a Local PC and Restore Data Offline?

Backups of a GeminiDB Redis instance differ from RDB files of an open-source Redis instance and cannot be used by users. Therefore, the backups cannot be downloaded to a local PC. If instance data is corrupted, you can restore backup data to a new instance. For details, see **Overview** and **Restoring Data to a New Instance**.

5.4.14 What Is the Data Backup Mechanism of GeminiDB Redis API? What Are the Impacts on Services?

To back up data of a GeminiDB Redis instance, snapshots need be created in seconds only for the storage layer, which does not affect compute nodes. Therefore, services are not affected as well. When backup data is uploaded, a small amount of CPU and bandwidth resources are consumed, which may cause slight jitter.

GeminiDB Redis instances support automated and manual backup. For details, see **Overview**.

5.4.15 Why Does the CPU Usage Remain High Despite Low Service Access Volume on a GeminiDB Redis Preferential Instance with 1 CPU and 2 Nodes?

GeminiDB Redis API collects metrics and reports monitoring data. The CPU usage of your nodes is high because of its small specifications. A GeminiDB Redis

Preferential instance with one CPU and two nodes is recommended in the test environment. A GeminiDB Redis instance with one CPU (standard) or two or more CPUs is recommended in the production environment.

5.4.16 Why Does the Number of Keys Decrease and Then Become Normal on the Monitoring Panel on the GUI of GeminiDB Redis API?

The number of keys is scanned and counted asynchronously by a GeminiDB Redis server to ensure final consistency. When an instance process is restarted (due to node restart, instance fault, specification change, or version upgrade), the keys are counted again. In this case, the number of keys displayed decreases temporarily and becomes accurate gradually.

5.4.17 Why Is CPU Usage of GeminiDB Redis Nodes Occasionally High?

There are many possible reasons, such as sudden spike in service traffic, large key operations, network jitter, data backup and garbage recycle tasks on a server. If the CPU usage is occasionally high, just ignore it. If there are other service reasons (excluding high QPS), you can submit a service ticket.

5.4.18 When Does a GeminiDB Redis Instance Become Read-Only?

To ensure that the GeminiDB Redis instance can still run properly when the storage space is about to be used up, the database is set to read-only, and data cannot be modified. If this happens, you can scale up the storage to restore the database status to read/write.

Table 5-1 Setting an instance status to read-only

Storage Capacity	Description
< 600 GB	● When the storage usage reaches 97%, the instance is read-only. ● When the storage usage decreases to 85%, the read-only status is automatically disabled for the instance.
≥ 600 GB	● When the remaining storage space is less than 18 GB, the instance is read-only. ● When the remaining storage space is greater than or equal to 90 GB, the read-only status is automatically disabled for the instance.

5.5 Database Connection

5.5.1 How Do I Connect to a GeminiDB Redis Instance?

You can connect to a GeminiDB Influx instance using a private network, public network, load balancer IP address, or program code. For details, see **Connection Modes**.

5.5.2 What Can I Do with IP Addresses of GeminiDB Redis Nodes?

GeminiDB Redis API provides multiple IP addresses for you to access a cluster and achieve load balancing and disaster recovery.

You can use multiple IP addresses in any of the following ways.

1. Use the connection pool on the service side implement load balancing and fault detection.
2. Contact customer service to configure the Elastic Load Balance (ELB) service and provide a unique IP address.
3. Configure domain names for multiple proxy IP addresses. For details about how to connect to an instance through a private domain name, see **Connecting to an Instance Using a Load Balancer Address (Recommended)**.

5.5.3 How Does Load Balancing Work in GeminiDB Redis API?

GeminiDB Redis API uses dedicated load balancers with scaled specifications and a maximum bandwidth of 10 Gbit/s. For details, see .

5.5.4 How Can I Create and Connect to an ECS?

1. To create an ECS, see *Elastic Cloud Server User Guide*.
   - The ECS to be created must be in the same VPC and security group with the GeminiDB Redis instance to which it connects.
   - Configure the security group rules to allow the ECS to access to the instance.
2. To connect to an ECS, see "Logging in to an ECS" *Getting Started with Elastic Cloud Server User Guide*.

5.5.5 Can I Change the VPC of a GeminiDB Redis Instance?

After a GeminiDB Redis instance is created, the VPC where the instance resides cannot be changed.

However, you can change a VPC by restoring the full backup of your instance to the VPC you want to use. For details, see **Restoring Data to a New Instance**.

5.5.6 How Do I Access a GeminiDB Redis Instance from a Private Network?

You can access a GeminiDB Redis instance through a load balancer or a directly-connected node.

- Access through a load balancer (recommended): The load balancer is associated with a high-availability backend cluster, using an internal IP address that is accessible only to clients. Periodical health checks are performed on backend nodes to prevent single points of failure (SPOFs).
- Access through a directly-connected node: An agent installed on a GeminiDB Redis node enables you to connect to any node. Then you can access the entire cluster. To prevent SPOFs, this access mode is only recommended in test scenarios.

For details about how to connect to a GeminiDB Redis instance over a private network, see **Connecting to GeminiDB Redis Instances over a Private Network**.

5.5.7 Do I Need to Enable Private Network Access Control for a Load Balancer After Setting a Security Group?

You can access a GeminiDB Redis instance through a node or load balancer. Therefore, you need to configure both a security group and private network access control for a load balancer to ensure instance security.

- Security groups take effect only for nodes. It is a collection of access control rules for ECSs and GeminiDB Redis instances that have the same security requirements and are mutually trusted in a VPC. For details, see **Configuring Security Group Rules for Nodes**.
- Security groups cannot take effect for load balancers. If access control is disabled, all IP addresses that can access the VPC of the GeminiDB Redis instance also can access the instance using a load balancer IP address. Therefore, you need to configure access control properly. For details, see **Enabling or Disabling Private Network Access for a Load Balancer**.

5.6 Backup and Restoration

5.6.1 How Long Can a GeminiDB Redis Instance Backup Be Saved?

Automated backup data is kept based on the backup retention period you specified. There is no limit for the manual backup retention period. You can delete manual backups as needed.

For more backup information, see **Managing Automated Backups** and **Managing Manual Backups**.

5.7 Memory Acceleration

5.7.1 Will All Data Be Cached to GeminiDB Redis Instances After Memory Acceleration Is Enabled and MySQL Database Data Is Updated?

No. You need to specify conversion rules of the MySQL database tablespaces, table names, and fields of GeminiDB Redis instances on the GUI. After the configuration is complete, data that meets the rules is automatically synchronized to a GeminiDB Redis instance.

5.7.2 If Memory Acceleration Is Enabled, GeminiDB Redis Instance Data Increases Continuously. Do I Need to Scale Out the Capacity? How Do I Manage Cached Data?

By default, each piece of data of GeminiDB Redis instances will expire in 30 days. You can adjust the expiration time. If the data volume keeps increasing, you need to scale out storage capacity of GeminiDB Redis instances in a timely manner.

5.7.3 Is Memory Acceleration Recommended When Customers' Service Data Can Be Synchronized Between MySQL and Redis? In Which Scenarios Can Memory Acceleration Be enabled?

If customers' service data can be synchronized between MySQL and Redis, you are advised to migrate cache data to GeminiDB Redis instances. Memory acceleration is recommended for new services to simplify development.

5.7.4 How Long Is the Latency of Synchronization from RDS for MySQL to GeminiDB Redis API? What Factors Affect the Latency?

Data can be synchronized in real time. The latency may be affected by the following factors and needs to be measured:

- Physical distance between RDS for MySQL and GeminiDB Redis instances. It is recommended that the instances be in the same region.
- You are advised to set the CPU specifications of RDS for MySQL to GeminiDB Redis instances to the same value.

5.7.5 Will the Source MySQL Database Be Affected After Memory Acceleration Is Enabled?

Memory acceleration works based on MySQL binlogs, which has little impact on the source MySQL database.

5.7.6 GeminiDB Redis Instances with Memory Acceleration Enabled Needs to Process a Large Number of Binlogs in a Short Period of Time. Will a Large Number of Resources Be Occupied and Online Services Be Affected?

If a large number of DDL operations are performed on the source MySQL database, a large number of GeminiDB Redis resources are consumed. You can query OPS (**dbcache_ops_per_sec**) after memory acceleration is enabled. You are advised to configure basic resource alarms. For details, see **Configuring Alarm Rules**.

5.8 Instance Freezing, Release, Deletion, and Unsubscription

Why Are My GeminiDB Redis Instances Released?

If your subscriptions have expired but not been renewed, or you are in arrears due to insufficient balance, your instances enter a grace period. If you do not renew the subscriptions or top up your account after the grace period expires, your instances will enter a retention period and become unavailable. If you still do not renew them or top up your account after the retention period ends, your instances will be released and your data stored will be deleted.

Why Are My GeminiDB Redis Instances Frozen?

Your instances may be frozen for a variety of reasons. The most common reason is that you are in arrears.

Can I Still Back Up Data If My Instances Are Frozen?

No. If your GeminiDB Redis instances are frozen because your account is in arrears, go to top up your account to unfreeze your instances and then back up instance data.

How Do I Unfreeze My Instances?

If your GeminiDB Redis instances are frozen because your account is in arrears, you can unfreeze them by renewing them or topping up your account. The frozen GeminiDB Redis instances can be renewed, released, or deleted. Yearly/Monthly instances that have expired cannot be unsubscribed from, while those that have not expired can be unsubscribed from.

What Impacts Does Instance Freezing, Unfreezing or Release Have on My Services?

- After an instance is frozen:
  - It cannot be accessed, and your services will be interrupted. For example, if a GeminiDB Redis instance is frozen, it cannot be connected.
  - No changes can be performed on it if it is a yearly/monthly instance.
  - It can be unsubscribed from or deleted manually.
- After it is unfrozen, you can connect to it again.
- Releasing an instance means deleting it. Before the deletion, GeminiDB Redis API determines whether to **move the instance to the recycle bin** based on the recycling policy you specified.

How Do I Renew My Instances?

After a yearly/monthly GeminiDB Redis instance expires, you can renew it on the **Renewals** page. For details, see **Renewal Management**.

Can My Instances Be Recovered After They Are Released or Unsubscribed From?

If your instance is moved to the recycle bin after being deleted, you can recover it from the recycle bin by referring to **Recycling an Instance**. If the recycling policy is not enabled, you cannot recover it.

When you unsubscribe from an instance, confirm the instance information carefully. If you have unsubscribed from an instance by mistake, purchase a new one.

How Do I Delete a GeminiDB Redis Instance?

- To delete a pay-per-use instance, see **Deleting a Pay-per-Use Instance**.
- To delete a yearly/monthly instance, see **Unsubscribing from a Yearly/ Monthly Instance**.