GeminiDB Redis

User Guide

Issue	01
Date	2025-09-29

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

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 KV databases (such as Redis and Pika databases):

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

Dimension	Item	Open-Source 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 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 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 databases are deployed in hybrid mode, and bandwidth is strictly limited. Flow control is easily triggered for databases with low specifications.	**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. Node addition does not involve data migration. Services are only affected for seconds.
	HA scenarios such as node breakdowns and primary/secondary switchovers	**Long switchover time: RTO > 30s**	**Second-level jitters, RTO < 10s**

Dimension	Item	Open-Source 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.
Cost	Utilization cost	**The in-memory storage is expensive.**	**30% cost reduction with the same specifications** Users can purchase additional compute resources and storage resources independently to eliminate the resource waste associated with coupled storage and compute.

Dimension	Item	Open-Source KV Database	GeminiDB
	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.
Advanced features	Autoscaling	Not supported	Supported
	Hash field expiration	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.**

Dimension	Item	Open-Source KV Database	GeminiDB
	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 absorb large bursts of traffic without interrupting services.

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

Live streams generate large amounts of hot data. Most of the data comes from popular live channels. To reduce costs for customers, GeminiDB Redis instances can store data from these popular live channels in memory and other data in shared disks.

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 instances can store data separately in memory and shared disks, 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 API and Versions

This section describes the compatible API and versions supported by GeminiDB Redis API.

Table 1-2 Compatible API and versions

Compatible API	Instance Type	Version
Redis	• Proxy cluster In a sharded cluster, a proxy cluster GeminiDB Redis instance is connected through proxies to a standalone Redis instance, Redis Sentinel, and Redis Cluster. The proxy cluster instance has strong horizontal scaling capabilities and can handle millions of QPS and dozens of terabytes of data. • Primary/Standby A primary/standby instance is compatible with a standalone Redis node and Redis Sentinel.	7.0, 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 instances facilitate hot and cold data exchanges while offering storage that significantly exceeds the memory limit. 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. **Table 1-7** lists the node memory capacity.

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

Instance Type	Storage (GB)	Node Flavor	Node CPUs	Node Memory (GB)	Nodes	QPS	Max. Connections	Databases	ACL Accounts
Cluster	4	geminidb.redis.medium.2	1	2	2	20,000	20,000	256	200

Instance Type	Storage (GB)	Node Flavor	Node CPUs	Node Memory (GB)	Nodes	QPS	Max. Connections	Databases	ACL Accounts
	8	geminidb.redis.medium.4	1	4	2	20,000	20,000	256	200
	16	geminidb.redis.large.4	2	8	2	40,000	20,000	256	200
	24	geminidb.redis.large.4	2	8	3	60,000	30,000	256	200
	32	geminidb.redis.large.4	2	8	4	80,000	40,000	256	200
	48	geminidb.redis.xlarge.4	4	16	3	120,000	30,000	1,000	200
	64	geminidb.redis.xlarge.4	4	16	4	160,000	40,000	1,000	200
	96	geminidb.redis.2xlarge.4	8	32	3	240,000	30,000	1,000	200
	128	geminidb.redis.2xlarge.4	8	32	4	320,000	40,000	1,000	200
	192	geminidb.redis.2xlarge.4	8	32	6	480,000	60,000	1,000	200
	256	geminidb.redis.2xlarge.4	8	32	8	640,000	80,000	1,000	200
	384	geminidb.redis.2xlarge.4	8	32	10	800,000	100,000	1,000	200
	512	geminidb.redis.4xlarge.4	16	64	6	960,000	60,000	1,000	200

Instance Type	Storage (GB)	Node Flavor	Node CPUs	Node Memory (GB)	Nodes	QPS	Max. Connections	Databases	ACL Accounts
	768	geminidb.redis.4xlarge.4	16	64	9	1,440,000	90,000	1,000	200
	1024	geminidb.redis.4xlarge.4	16	64	12	1,920,000	120,000	1,000	200
	2048	geminidb.redis.4xlarge.4	16	64	22	3,520,000	220,000	1,000	200
	4096	geminidb.redis.8xlarge.4	32	128	24	7,680,000	240,000	1,000	200
	8192	geminidb.redis.8xlarge.4	32	128	36	11,520,000	360,000	1,000	200

Table 1-4 Primary/Standby GeminiDB Redis instance specifications (fast configuration)

Instance Type	Storage (GB)	Node Flavor	Node CPUs	Node Memory (GB)	Nodes	Shards	QPS	Max. Connections	Databases	ACL Accounts
Primary/Standby	16	geminidb.redis.medium.4	1	4	2	1	10,000	10,000	1,000	200
	24	geminidb.redis.large.4	2	8	2	1	20,000	10,000	1,000	200
	32	geminidb.redis.large.4	2	8	2	1	20,000	10,000	1,000	200

Instance Type	Storage (GB)	Node Flavor	Node CPUs	Node Memory (GB)	Nodes	Shards	QPS	Max. Connections	Databases	ACL Accounts
	48	geminidb.redis.xlarge.4	4	16	2	1	40,000	20,000	1,000	200
	64	geminidb.redis.xlarge.4	4	16	2	1	40,000	20,000	1,000	200
	96	geminidb.redis.2xlarge.4	8	32	2	1	80,000	20,000	1,000	200

Table 1-5 Specifications of a capacity-oriented GeminiDB Redis instance with cloud native storage

Flavor	Number of vCPUs	Maximum Connections per Node	DB Instances	ACL Accounts
geminidb.redis-geminifs.large.4	2 vCPUs	10,000	1,000	200
geminidb.redis-geminifs.xlarge.4	4 vCPUs	10,000	1,000	200
geminidb.redis-geminifs.2xlarge.4	8 vCPUs	10,000	1,000	200
geminidb.redis-geminifs.4xlarge.4	16 vCPUs	10,000	1,000	200
geminidb.redis-geminifs.8xlarge.4	32 vCPUs	10,000	1,000	200

Table 1-6 Specifications of a standard GeminiDB Redis instance with cloud native storage

Shard Specifications	Memory (GB)	Maximum Connections per Node	Databases	ACL Accounts
4 GB	4	10,000	1,000	200
8 GB	8	10,000	1,000	200
16 GB	16	10,000	1,000	200
32 GB	32	10,000	1,000	200
64 GB	64	10,000	1,000	200
128 GB	128	10,000	1,000	200

Table 1-7 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

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.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

1.5 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 1-8 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.

Status	Description
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.

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	In a sharded cluster, a proxy cluster GeminiDB Redis instance is connected through proxies to a standalone Redis node, Redis Sentinel, and Redis Cluster. The proxy cluster instance has strong horizontal scaling capabilities and can handle millions of QPS and dozens of terabytes of data.	**Buying and Connecting to a Cluster Instance**
Primary/Standby	A primary/standby instance is compatible with a standalone Redis node and Redis Sentinel. This instance type is used when hashtags are unavailable.	**Buying and Connecting to a Primary/Standby 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 **Connecting to a GeminiDB Redis Instance**.

2.2 Buying and Connecting to a Cluster Instance

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

In a sharded cluster, a proxy cluster GeminiDB Redis instance is connected through proxies to a standalone Redis node, Redis Sentinel, and Redis Cluster. The proxy cluster instance has strong horizontal scaling capabilities and can handle millions of QPS and dozens of terabytes of data.

Each tenant can create a maximum of 50 GeminiDB Redis instances by default. To request a higher quota, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

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

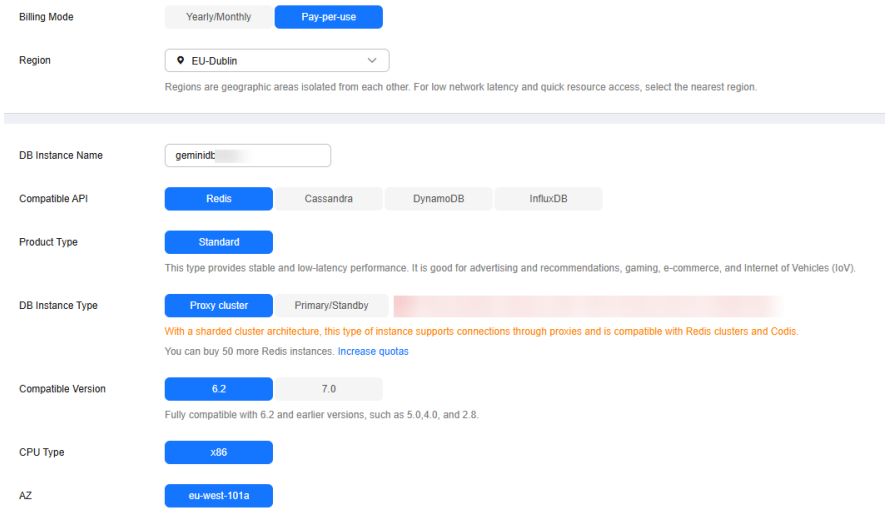
  For details about other connection methods, see **Instance Connection and Management**.

Step 1: Buying an Instance

1. Log in to the **GeminiDB console**.
2. On the **Instances** page, click **Buy DB Instance**.
3. 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.

Figure 2-1 Billing mode and basic information

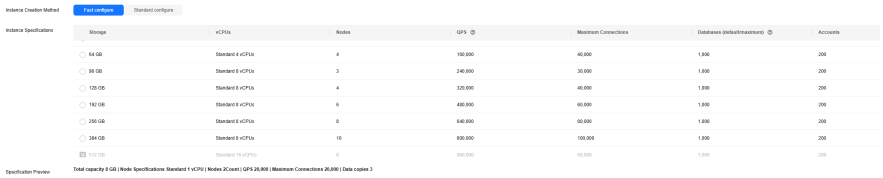


Parameter	Example Value	Description
Billing mode description	Pay-per-use	Billing mode of an instance • **Yearly/Monthly**: A prepaid billing mode in which you pay for resources before using it. Bills are settled based on the subscription period. The longer the subscription term, the bigger the discount. This mode is a good option for long-term stable services. • **Pay-per-use** is a postpaid mode. You are billed based on how long you have actually used GeminiDB. Pricing is listed on a per-hour basis, and bills are calculated down to the second. This mode allows you to adjust resource usage easily. You neither need to prepare for resources in advance, nor end up with excessive or insufficient preset resources.
Region	Select EU-Dublin.	Region where a tenant is located **NOTE** To reduce network latency, select a region nearest from which you will access the instance. Instances deployed in different regions cannot communicate with each other over a private network. After you buy an instance, you cannot change its region.

Parameter	Example Value	Description
DB Instance Name	User-defined	The instance name: • Can be the same as an existing instance name. • Can contain 4 to 64 characters and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_).
Compatible API	Redis	GeminiDB is compatible with mainstream NoSQL databases, including Redis, DynamoDB, Cassandra, and InfluxDB. You can select GeminiDB APIs by following **How Do I Select an API?**
Storage Type	Classic	• **Classic**: classic architecture with decoupled storage and compute • **Cloud native**: more flexible, new-gen version with support for more AZs
Product Type	Standard	Stable and low-latency performance is provided for common scenarios such as advertising and recommendation, gaming, e-commerce, and Internet of Vehicles (IoV).
DB Instance Type	Proxy cluster	**Proxy cluster**: In a sharded cluster, a proxy cluster GeminiDB Redis instance is connected through proxies to a standalone Redis instance, Redis Sentinel, and Redis Cluster. The proxy cluster instance has strong horizontal scaling capabilities and can handle millions of QPS and dozens of terabytes of data. **NOTE** To create a Redis Cluster instance, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service to grant required permissions.
Compatible Version	6.2	7.0, 6.2 (including 6.2.*X*), 5.0, and earlier versions
CPU Type	x86	x86 CPUs use the Complex Instruction Set Computing (CISC) instruction set. Each instruction can be used to execute low-level hardware operations. Executing these instructions is complex and time-consuming.

Parameter	Example Value	Description
AZ	AZ 1, AZ 2, and AZ 3	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 2-2 Specifications and storage



Parameter	Example Value	Description
Instance Creation Method	**Fast configure**	Two options are available: • **Fast configure** Provides you with recommended specifications. You can select one of them based on service requirements, without the need to specify the specifications, node quantity, and storage space. • **Standard configure** Provides a standard process to configure instance specifications, including specifying the specifications, node quantity, and storage space. Currently, a maximum of 36 nodes are supported. To add more, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Parameter	Example Value	Description
Instance Specifications	2U8GB	Higher CPU specifications provide better performance. Select specifications as needed. For details, see **Instance Specifications**.

Parameter	Example Value	Description
VPC	default_vpc	Virtual private network where your instances are located. A VPC isolates networks for different services. You can select an existing VPC or create a VPC. **NOTE** • After a GeminiDB Redis instance is created, its VPC cannot be changed. • To connect a GeminiDB Redis instance to an ECS over a private network, ensure the GeminiDB Redis instance and the ECS are in the same VPC. If they are not, you can create a **VPC peering connection** between them.
Subnet	default_subnet	A subnet provides dedicated network resources that are logically isolated from other networks for security purposes.
Password	Skip	• **Skip**: You can set the database password after creating an instance. • **Configure**: You can set the database password when creating an instance. **NOTE** You cannot set a password after creating a Redis Cluster GeminiDB Redis instance.

Parameter	Example Value	Description
Password	Configured based on the password policy	If **Password** is set to **Configure**, you need to set the database password. • Must be 8 to 32 characters long. • Can contain at least two types of the following characters: 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.
Enterprise Project	default	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***.

Retain the default values for other parameters.

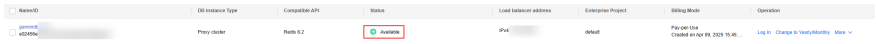
5. On the order confirmation page, check the instance information. If you need to modify the information, click **Previous**. If no modification is required, read and agree to the service agreement and click **Submit**.

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 instance is created, its status becomes **Available**.

Figure 2-3 Available instance



Step 2: Connecting to an Instance Through DAS

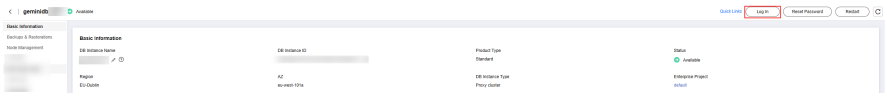
DAS enables you to manage DB instances from a web-based console, simplifying database management and improving efficiency. You can connect and manage instances through DAS. By default, you have the permission of remote login. DAS is secure and convenient for connecting to DB instances.

1. Log in to the **GeminiDB console**.
2. 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

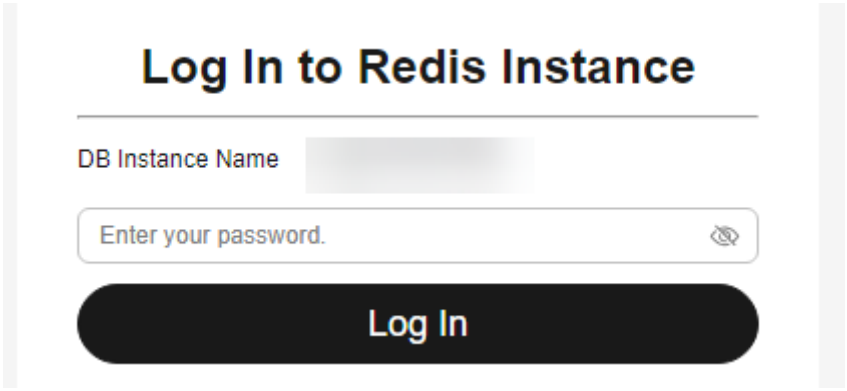
Alternatively, click the instance name to go to the **Basic Information** page. Click **Log In** in the upper right corner of the page.

Figure 2-5 Connecting to a GeminiDB Redis instance



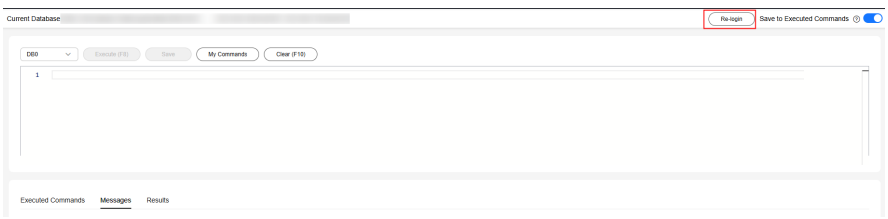
3. Enter a password for logging in to the instance.

Figure 2-6 Logging in to the GeminiDB Redis instance



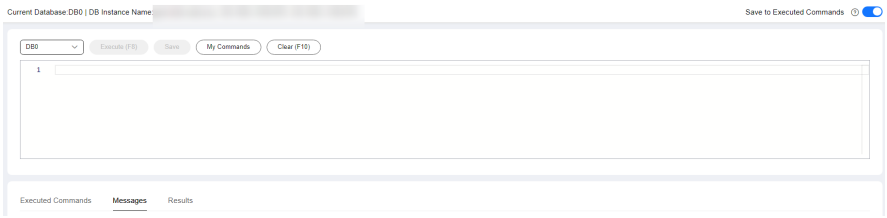
If you need to log in again after the password is reset, click **Re-login** in the upper right corner and use the new password.

Figure 2-7 Re-login



4. Manage relevant databases.

Figure 2-8 Instance homepage

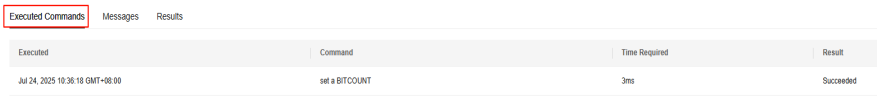


- Save commands to the execution record.

  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 2-9 Viewing executed commands

  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.

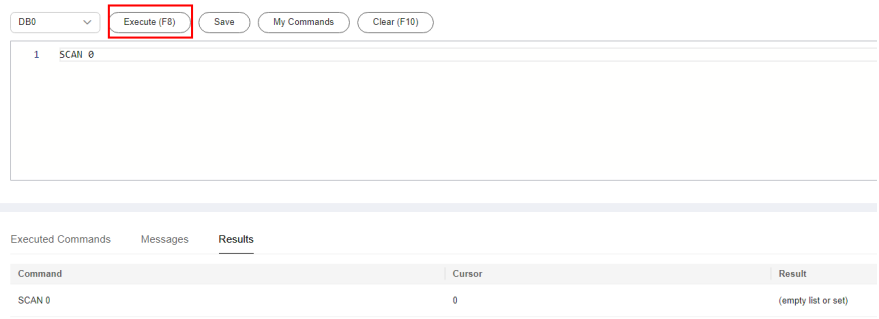
- Execute a command.

  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 2-10 Executing a command

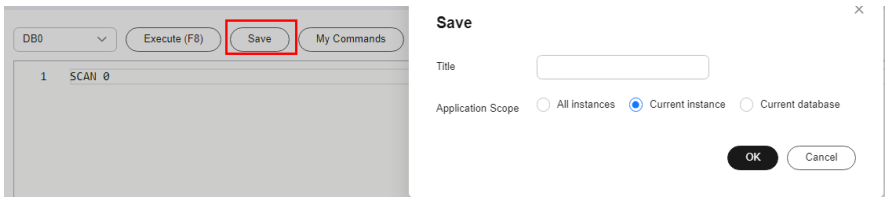
  

  After a command is executed, you can view the execution result on the **Results** page.

- 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 2-11 Saving a command

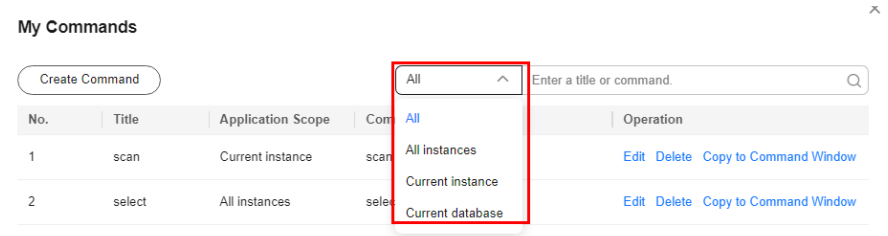


- View my commands.

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

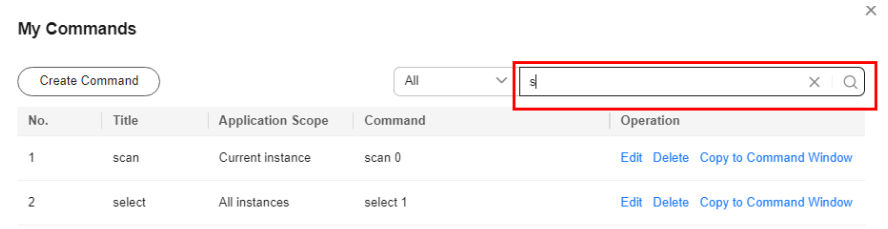
  You can set a filter to narrow the scope of commands. If you select **All**, all commands saved in the current account are displayed.

Figure 2-12 Filtering commands



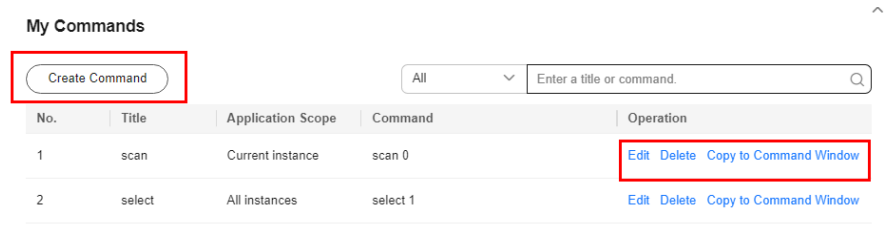
Alternatively, you can enter a command title or statement in the search box to search for the corresponding command.

Figure 2-13 Searching for a command



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

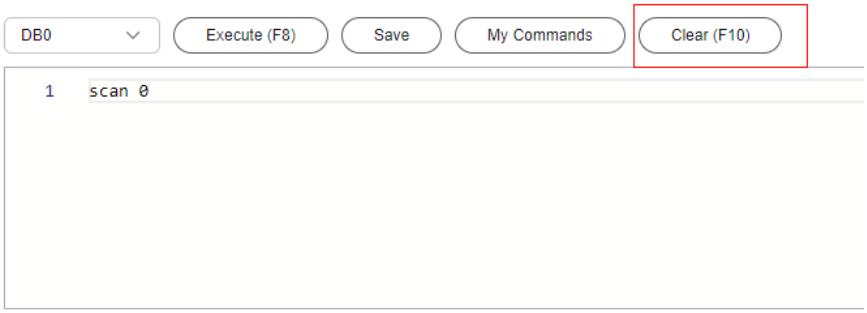
Figure 2-14 Managing a command



- Clear a command.

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

Figure 2-15 Clearing a command



FAQs

Question: What should I do if the DAS console cannot be redirected after I click **Log In** in the **Operation** column in the instance list or click **Log In** on the **Basic Information** page?

Solution: Set your browser to allow pop-ups and try again.

2.3 Buying and Connecting to a Primary/Standby Instance

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

A primary/standby instance is compatible with a standalone Redis node and Redis Sentinel. This instance type is used when hashtags are unavailable.

Each tenant can create a maximum of 50 GeminiDB Redis instances by default. To request a higher quota, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

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

  For details about other connection methods, see **Instance Connection and Management**.

Usage Notes

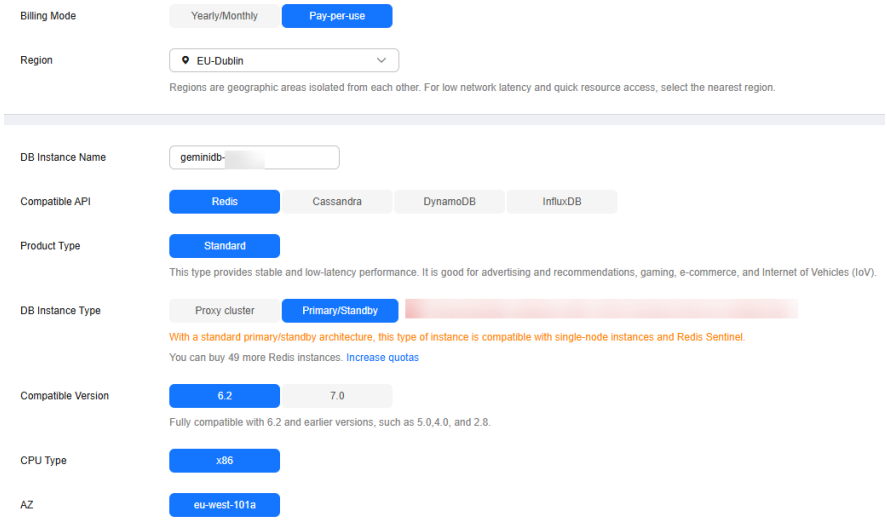
This function is now in OBT. To use it, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Step 1: Buying an Instance

1. Log in to the **GeminiDB console**.
2. On the **Instances** page, click **Buy DB Instance**.
3. 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 3-9** lists details about the parameters.

Figure 2-16 Billing mode and basic information

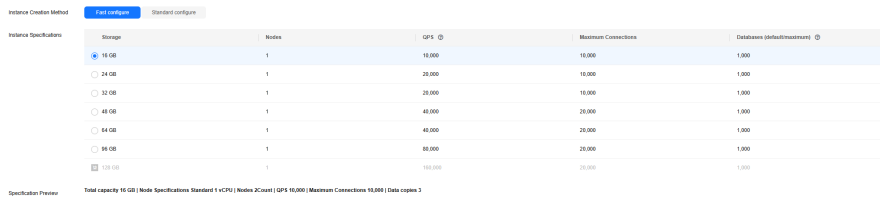


Parameter	Example Value	Description
Billing mode description	Pay-per-use	Billing mode of an instance • **Yearly/Monthly**: A prepaid billing mode in which you pay for resources before using it. Bills are settled based on the subscription period. The longer the subscription term, the bigger the discount. This mode is a good option for long-term stable services. • **Pay-per-use** is a postpaid mode. You are billed based on how long you have actually used GeminiDB. Pricing is listed on a per-hour basis, and bills are calculated down to the second. This mode allows you to adjust resource usage easily. You neither need to prepare for resources in advance, nor end up with excessive or insufficient preset resources.
Region	Select EU-Dublin.	Region where a tenant is located **NOTE** To reduce network latency, select a region nearest from which you will access the instance. Instances deployed in different regions cannot communicate with each other over a private network. After you buy an instance, you cannot change its region.

Parameter	Example Value	Description
DB Instance Name	User-defined	The instance name: • Can be the same as an existing instance name. • Can contain 4 to 64 characters and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_).
Compatible API	Redis	GeminiDB is compatible with mainstream NoSQL databases, including Redis, DynamoDB, Cassandra, and InfluxDB. You can select GeminiDB APIs by following **How Do I Select an API?**
Product Type	Standard	Stable and low-latency performance is provided for common scenarios such as advertising and recommendation, gaming, e-commerce, and Internet of Vehicles (IoV).
DB Instance Type	Primary/Standby	**Primary/Standby** A primary/standby instance is compatible with a standalone Redis node and Redis Sentinel. This instance type is used when hash tags are unavailable.
Compatible Version	6.2	7.0, 6.2 (including 6.2.*X*), 5.0, and earlier versions
CPU Type	x86	x86 CPUs use the Complex Instruction Set Computing (CISC) instruction set. Each instruction can be used to execute low-level hardware operations. Executing these instructions is complex and time-consuming.

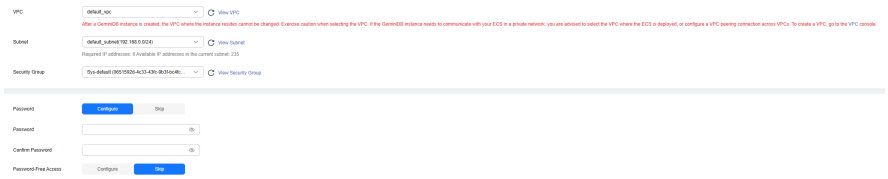
Parameter	Example Value	Description
AZ	AZ 1, AZ 2, and AZ 3	Availability zone where the instance is created. An AZ is a part of a region with its own independent power supplies and networks. AZs are physically isolated but can communicate with each other over a private network. If there are multiple AZs, you need to select primary and standby AZs. Instances can be deployed in a single AZ or three AZs. ● If low network latency is required, deploy your instance in one AZ. ● To meet disaster recovery requirements, select three AZs and specify primary and standby AZs. – Primary AZ: AZ where a primary node is located – Standby AZ: AZ where a standby node is located

Figure 2-17 Specifications and storage



Parameter	Example Value	Description
Instance Creation Method	Fast configure	Two options are available: • **Fast configure** Provides you with recommended specifications. You can select one of them based on service requirements, without the need to specify the specifications, node quantity, and storage space. • **Standard configure** Provides a standard process to configure instance specifications, including specifying the specifications, node quantity, and storage space.
Instance Specifications	2U8GB	Higher CPU specifications provide better performance. Select specifications as needed. For details, see **Instance Specifications**.

Figure 2-18 Network and database configurations



Parameter	Example Value	Description
VPC	default_vpc	Virtual private network where your instances are located. A VPC isolates networks for different services. You can select an existing VPC or create a VPC. **NOTE** • After a GeminiDB Redis instance is created, its VPC cannot be changed. • To connect a GeminiDB Redis instance to an ECS over a private network, ensure the GeminiDB Redis instance and the ECS are in the same VPC. If they are not, you can create a **VPC peering connection** between them.
Subnet	default_subnet	A subnet provides dedicated network resources that are logically isolated from other networks for security purposes.
Password	Skip	• **Skip**: You can set the database password after creating an instance. • **Configure**: You can set the database password when creating an instance.
Password	Configured based on the password policy	If **Password** is set to **Configure**, you need to set the database password. • Must be 8 to 32 characters long. • Can contain at least two types of the following characters: 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.

Parameter	Example Value	Description
Enterprise project	default	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***.

Retain the default values for other parameters.

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 instance is created, its status becomes **Available**.

Figure 2-19 Available instance

Step 2: Connecting to an Instance Through DAS

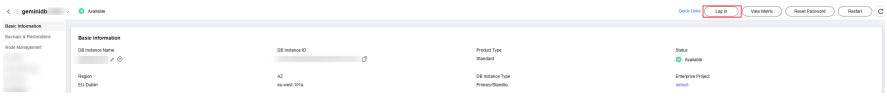
DAS enables you to manage DB instances from a web-based console, simplifying database management and improving efficiency. You can connect and manage instances through DAS. By default, you have permissions required for remote login. DAS is recommended for connecting to your instance.

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

   Figure 2-20 Logging in to a GeminiDB Redis instance

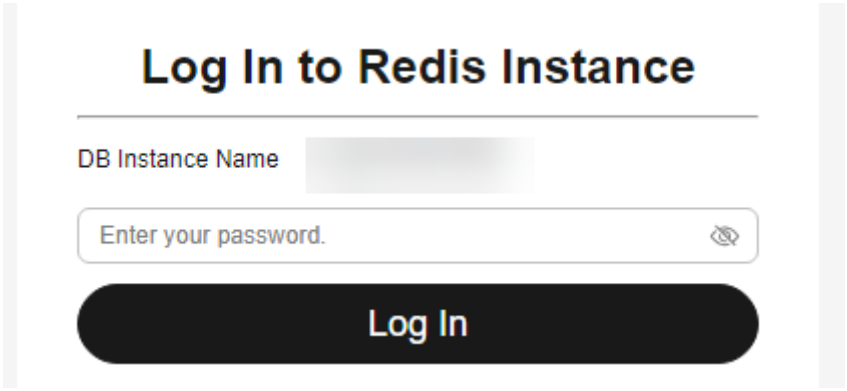
   Alternatively, click the instance name to go to the **Basic Information** page. Click **Log In** in the upper right corner of the page.

Figure 2-21 Logging in to a GeminiDB Redis instance



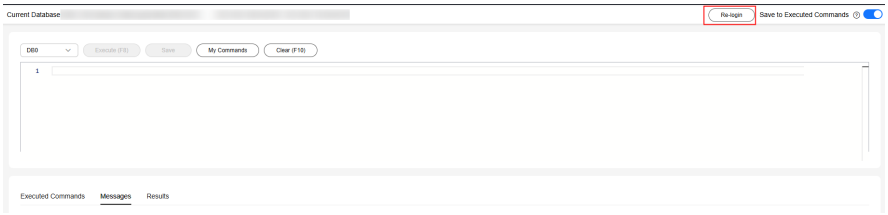
3. Enter a password for logging in to the instance.

Figure 2-22 Logging in to a GeminiDB Redis instance



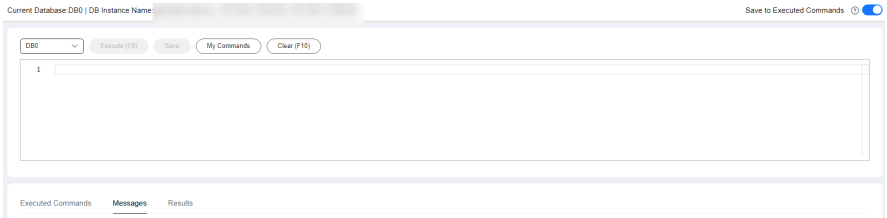
If you need to log in again after the password is reset, click **Re-login** in the upper right corner and use the new password.

Figure 2-23 Re-login



4. Manage relevant databases.

Figure 2-24 Instance homepage

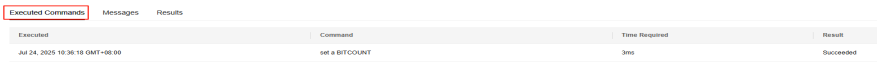


- Save commands to the execution record.

  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 2-25 Viewing executed commands



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.

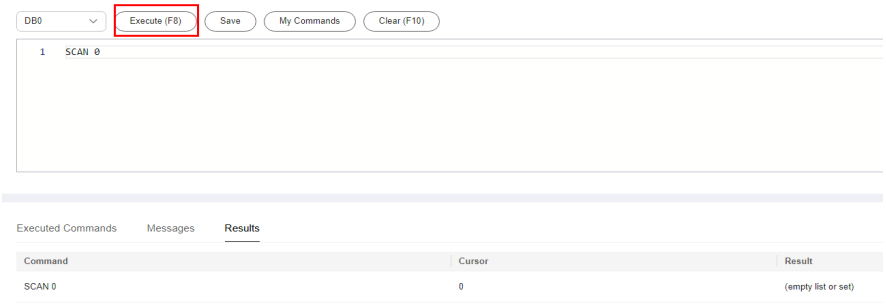
- Execute a command.

  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 2-26 Executing a command

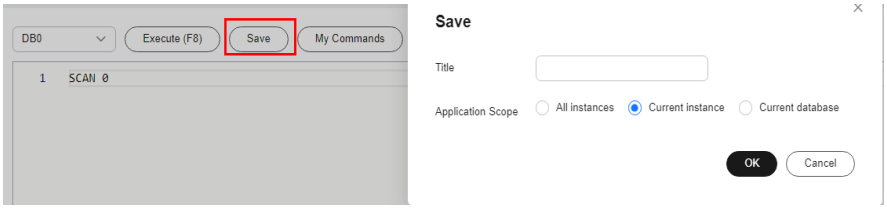


After a command is executed, you can view the execution result on the **Results** page.

- 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 2-27 Saving a command

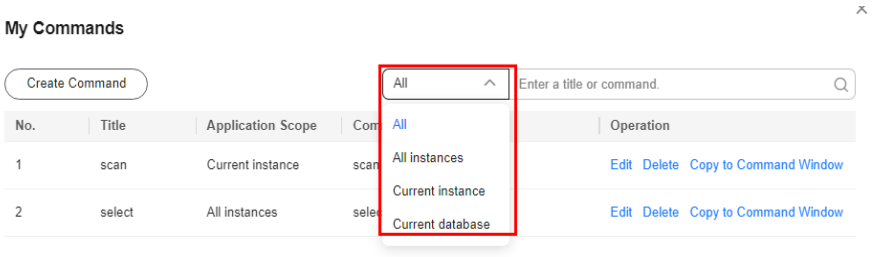


- View my commands.

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

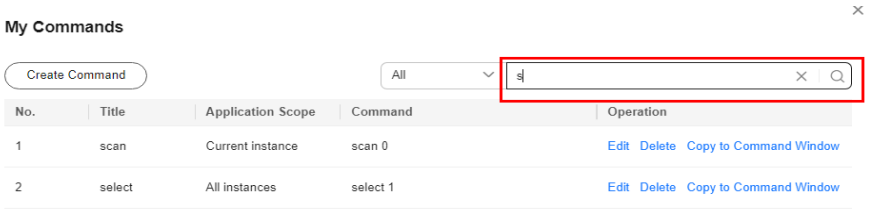
  You can set a filter to narrow the scope of commands. If you select **All**, all commands saved in the current account are displayed.

Figure 2-28 Filtering commands



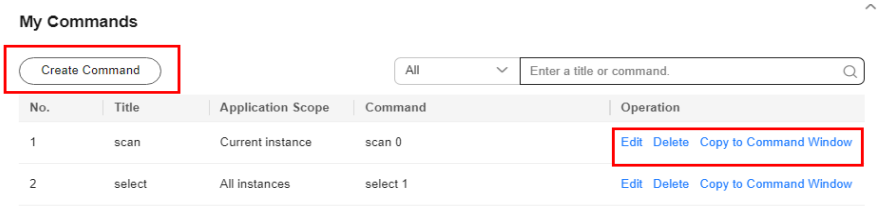
Alternatively, you can enter a command title or statement in the search box to search for the corresponding command.

Figure 2-29 Searching for a command



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

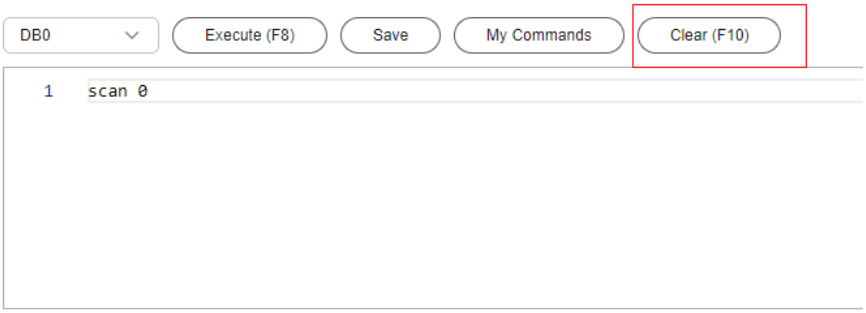
Figure 2-30 Managing a command



- Clear a command.

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

Figure 2-31 Clearing a command



FAQs

Question: What should I do if the DAS console cannot be redirected after I click **Log In** in the **Operation** column in the instance list or click **Log In** on the **Basic Information** page?

Solution: Set your browser to allow pop-ups and try again.

3 Working with GeminiDB Redis API

3.1 Using IAM to Grant Access to GeminiDB Redis API

3.1.1 Creating a User and Granting GeminiDB Redis API 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.

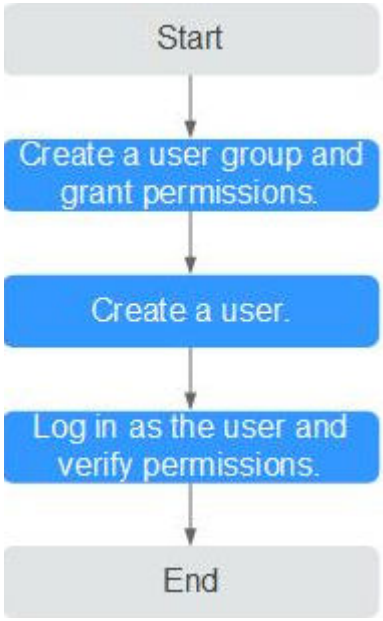
The following describes the procedure for granting permissions (see **Figure 3-1**).

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



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 as the created user by following **Logging In to Huawei Cloud as an IAM User**, select the authorized region, and verify 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 Custom Policies of GeminiDB Redis API

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.

Usage Notes

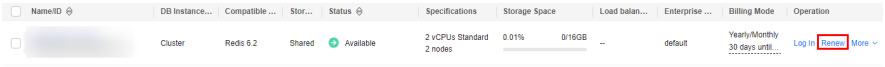
- Pay-per-use instances cannot be renewed.

Renewing a Single Yearly/Monthly Instance

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

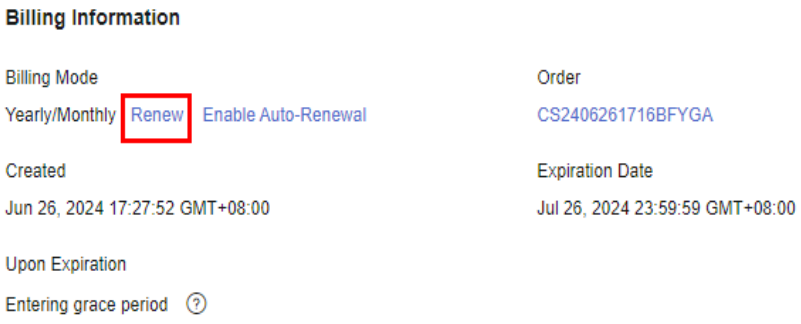
Step 2 On the **Instances** page, locate the instance that you want to renew and click **Renew** in the **Operation** column.

Figure 3-2 Renewing an instance



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



Step 3 On the displayed page, renew the instance.

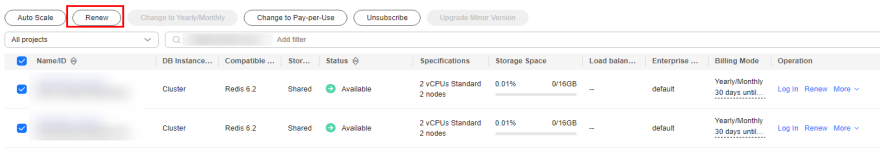
----End

Renewing Multiple Instances In Batches

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

Step 2 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



Step 3 In the displayed dialog box, click **Yes**.

----End

3.2.2 Changing a Pay-per-Use Instance to Yearly/Monthly

This section describes how to change a pay-per-use GeminiDB Redis instance to yearly/monthly. To use a pay-per-use instance for a long time, you can change it to yearly/monthly to reduce costs.

Usage Notes

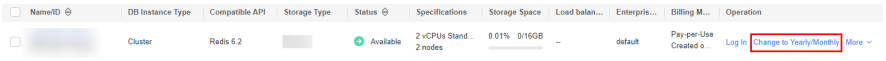
- Only when the status of a pay-per-use instance is **Available**, its billing mode can be changed to yearly/monthly.

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

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

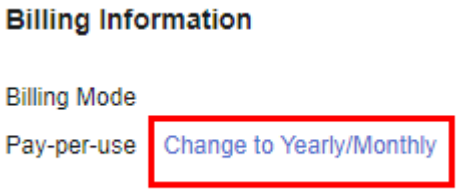
Step 2 On the **Instances** page, locate the target instance and click **Change to Yearly/Monthly** in the **Operation** column.

Figure 3-5 Change to Yearly/Monthly



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 Change to Yearly/Monthly



Step 3 On the displayed page, specify a subscription duration in month. The minimum duration is one month.

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

Step 4 Select a payment method and click **Pay**.

Step 5 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**.

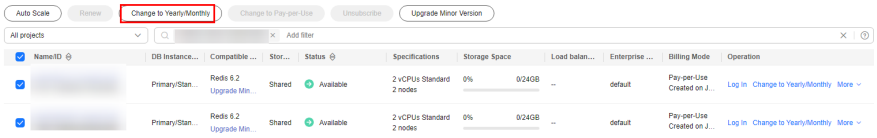
----End

Changing the Billing Mode of Pay-per-Use Instances to Yearly/Monthly In Batches

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

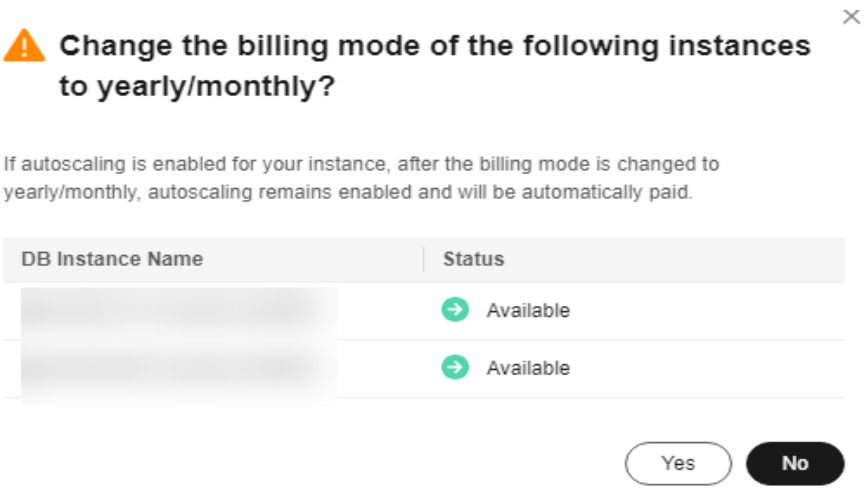
Step 2 On the **Instances** page, select the instances whose billing mode you want to change and click **Change to Yearly/Monthly** above the instance list.

Figure 3-7 Change to Yearly/Monthly



Step 3 In the displayed dialog box, click **Yes**.

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



Step 4 On the displayed page, specify a subscription 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

3.2.3 Changing a Yearly/Monthly Instance to Pay-per-Use

You can change a yearly/monthly GeminiDB Redis instance to pay-per-use if you intend to discontinue long-term use once it expires.

Usage Notes

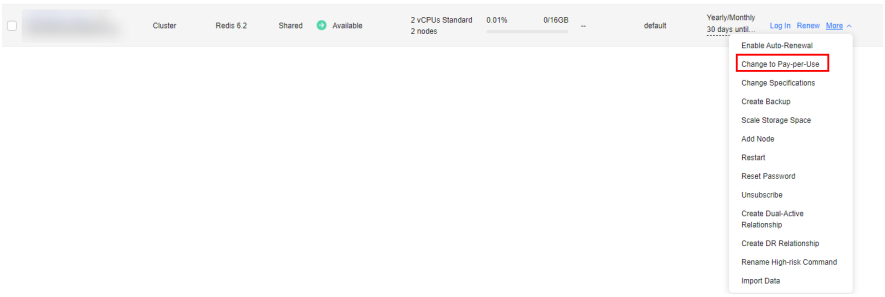
- The billing mode of a yearly/monthly instance can only be changed to pay-per-use when the instance is in the **Available** status.

Changing a Single Yearly/Monthly Instance to Pay-per-Use

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

Step 2 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 Change to Pay-per-Use



Step 3 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.

NOTE

After the billing mode is changed, auto-renewal will be disabled.

Step 4 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 5 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 6 In the displayed dialog box, click **Yes**.

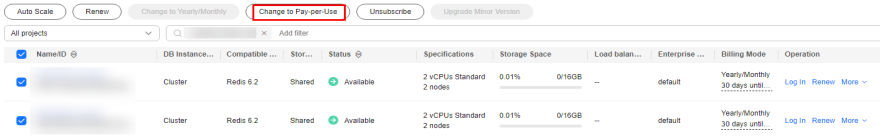
----End

Batch Changing Yearly/Monthly to Pay-per-Use

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

Step 2 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 Change to Pay-per-Use



Step 3 In the displayed dialog box, click **Yes**.

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.

NOTE

After the billing mode is changed, auto-renewal will be disabled.

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

3.2.4 Unsubscribing a Yearly/Monthly Instance

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

Usage Notes

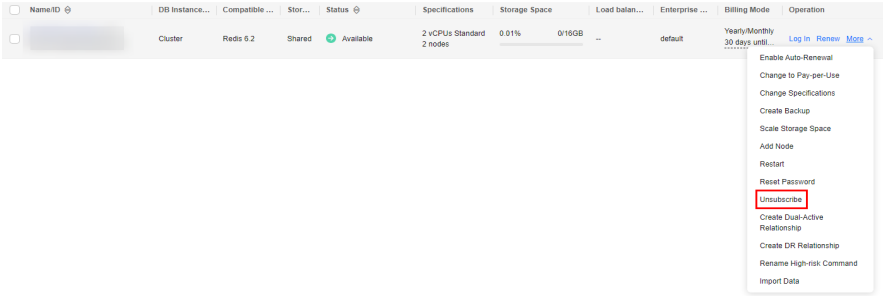
- Unsubscribed operations cannot be undone. Exercise caution when performing this operation. To retain data, create a manual backup before unsubscription. For details, see **Creating a Manual Backup**.
- After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.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

Unsubscribing a Single Yearly/Monthly Instance

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

Step 2 On the **Instances** page, locate the instance you want to unsubscribe and choose **More** > **Unsubscribe** in the **Operation** column.

Figure 3-11 Unsubscribe



Step 3 In the displayed dialog box, click **Yes**.

Step 4 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

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

NOTE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. To retain data, back it up before submitting the unsubscription request.

Step 6 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.

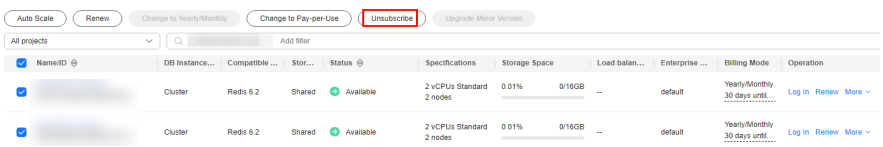
----End

Batch Unsubscribing from Yearly/Monthly Instances

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

Step 2 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 Unsubscribe



Step 3 In the displayed dialog box, click **Yes**.

Step 4 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

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

NOTE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. To retain data, back it up before submitting the unsubscription request.

Step 6 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 a GeminiDB Redis Instance

3.3.1 Buying a GeminiDB Redis Cluster Instance

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

In a sharded cluster, a proxy cluster GeminiDB Redis instance is connected through proxies to a standalone Redis instance, Redis Sentinel, and Redis Cluster. The proxy cluster instance has strong horizontal scaling capabilities and can handle millions of QPS and dozens of terabytes of data.

Each tenant can create a maximum of 50 GeminiDB Redis instances by default. To request a higher quota, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Prerequisites

- You have created a Huawei Cloud account.

Procedure

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

Step 2 On the **Instances** page, click **Buy DB Instance**.

Step 3 On the displayed page, specify instance specifications and click **Next**.

Figure 3-13 Billing mode and basic information

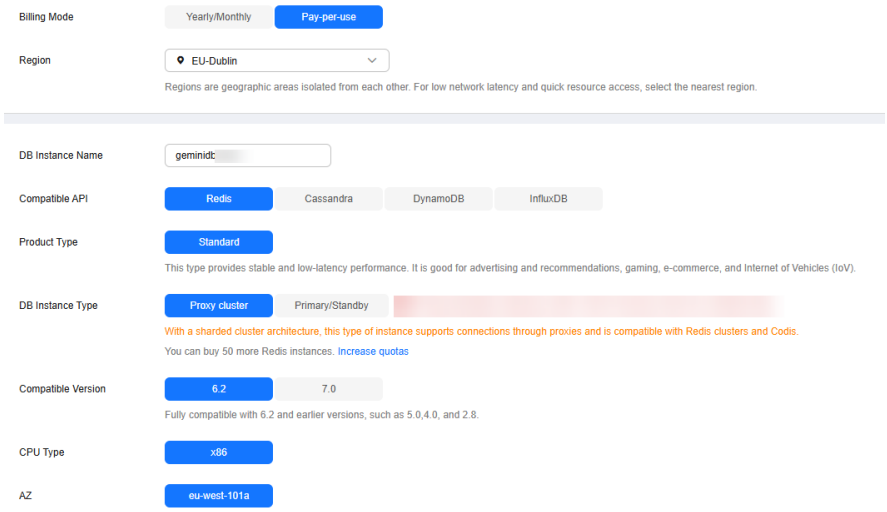


Table 3-1 Billing mode description

Parameter	Description
Billing Mode	Select **Yearly/Monthly** or **Pay-per-use**. • **Yearly/Monthly** – Specify **Required Duration**. The system deducts fees from your account 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. For details, see **Changing a Yearly/Monthly Instance 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 a Yearly/Monthly Instance**. ▪ Yearly/Monthly instances with cloud native storage are now in OBT. To use such an instance, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service. • Pay-per-use – If you select this billing mode, you are billed based on how much time the instance is in use. – To use an instance for a long time, change its billing mode to yearly/monthly to reduce costs. For details, see **Changing a Pay-per-Use Instance to Yearly/Monthly**.

Table 3-2 Basic information

Parameter	Description
Region	Region where a tenant is located **NOTE** To reduce network latency, select a region nearest from which you will access the instance. 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. • Can contain 4 to 64 characters and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_). You can change the name of an instance after it is created. For details, see **Modifying a GeminiDB Redis Instance Name**.
Compatible API	**Redis** GeminiDB is compatible with mainstream NoSQL databases, including Redis, DynamoDB, Cassandra, and InfluxDB. You can select GeminiDB APIs by following **How Do I Select an API?**
Storage Type	• **Classic**: classic architecture with decoupled storage and compute • **Cloud native**: more flexible, new-gen version with support for more AZs **NOTE** – Cloud native storage supports only proxy cluster instances. – Classic and cloud native are different deployment modes. Cloud native supports more AZs. If both classic and cloud native are supported, you can select any of them.
Product Type	**Standard**: Stable and low-latency performance is provided for common scenarios such as advertising and recommendation, gaming, e-commerce, and Internet of Vehicles (IoV).
DB Instance Type	**Proxy cluster**: In a sharded cluster, a proxy cluster GeminiDB Redis instance is connected through proxies to a standalone Redis instance, Redis Sentinel, and Redis Cluster. The proxy cluster instance has strong horizontal scaling capabilities and can handle millions of QPS and dozens of terabytes of data. **NOTE** To create a Redis Cluster instance, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service to grant required permissions.
Compatible Version	7.0, 6.2 (including 6.2.*X*), 5.0, and earlier versions

Parameter	Description
CPU Type	x86 x86 CPUs use the Complex Instruction Set Computing (CISC) instruction set. Each instruction can be used to execute low-level hardware operations. Executing these instructions is complex and time-consuming.
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

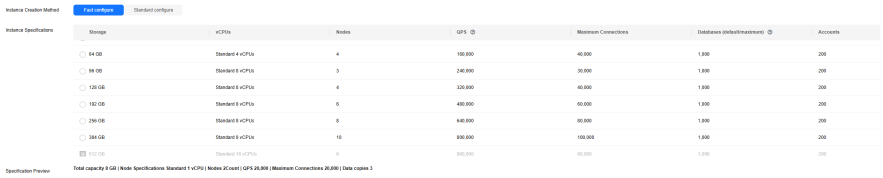


Table 3-3 Specifications and storage

Parameter	Description
Instance Creation Method	Two options are available: • **Fast configure** Recommended specifications, node quantity, and storage space **NOTE** – Instance specifications with the memory of 8 GB and 16 GB are available only in single AZs. The console shows available specifications. – The QPS is only for reference. **Standard configure** Instance flavor, node specifications, node quantity, and storage space that can be specified
Instance Specifications	You need to specify instance specifications after selecting **Fast configure** for **Instance Creation Method**. Higher CPU specifications provide better performance. Select specifications as needed. For details, see **Instance Specifications**.

Parameter	Description
Specification Type	You need to select a specification type after selecting **Standard configure** for **Instance Creation Method**. • **Standard**: The default and recommended CPU-to-memory ratio is 1:4, which balances low latency demands with high concurrency requirements. • **Enhanced**: The CPU-to-memory ratio is 1:8, which boosts the access hit rate while reducing latency.
Node Specifications	You need to select node specifications after selecting **Standard configure** for **Instance Creation Method** and **Classic** for **Storage Type**. For details, see **Instance Specifications**.
Nodes	You need to specify the node quantity after selecting **Standard configure** for **Instance Creation Method** and **Classic** for **Storage Type**. Number of required nodes. After an instance is created, you can add nodes. Currently, a maximum of 36 nodes are supported. To add more, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.
Total Storage Space	You need to specify the storage space after selecting **Standard configure** for **Instance Creation Method**. Higher CPU specifications provide better performance. Select specifications as needed. For details, see **Instance Specifications**.
Specification Preview	After you select instance specifications, the system automatically shows details of the total capacity, node specifications, number of nodes, QPS benchmark, total number of connections, and number of data copies. This helps keep track of the selected instance specifications.

Table 3-4 Network

Parameter	Description
VPC	Virtual private network where your instances are located. A VPC isolates networks for different services. You can select an existing VPC or create a VPC. 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. • To connect a GeminiDB Redis instance to an ECS over a private network, ensure the GeminiDB Redis instance and the ECS are 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** An IPv6 subnet cannot be associated with your instance. Select an IPv4 subnet.
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.

Table 3-5 Database configuration

Parameter	Description
Password	• **Skip**: You can set the database password after creating an instance. • **Configure**: You can set the database password when creating an instance.
Password	Password of database administrator **rwuser**: • Must be 8 to 32 characters long. • Can contain at least two types of the following characters: 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.

Parameter	Description
Confirm Password	Enter the database password again.

Table 3-6 Enterprise project

Parameter	Description
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 3-7 Advanced settings

Parameter	Description
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 **Encrypting Data over SSL for a GeminiDB Redis Instance**.

Parameter	Description
Tags	This setting is optional. Adding tags helps you better identify and manage your instances. A maximum of 20 tags can be added for each instance. A tag consists of a tag key and a tag value. • A tag key is mandatory if the instance will be tagged. Each tag key is unique for each instance. It can contain 1 to 128 characters, cannot start with **_sys_**, and cannot start or end with a space. Only letters, digits, spaces, and the following special characters are allowed: -_@.:/+= • A tag value is optional if the instance will be tagged. The value can be empty. The value can contain a maximum of 255 characters. Only letters, digits, spaces, and the following special characters are allowed: _.-:+=@/ 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 **Tag Management**.

Table 3-8 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 option, the instance is automatically renewed based on the subscription duration.

Step 4 On the displayed page, confirm instance details.

- Yearly/Monthly
  - 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 5 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.

The default database port of the instance is **6379** and cannot be changed.

----End

3.3.2 Buying a Primary/Standby GeminiDB Redis Instance

This section describes how to buy a primary/standby Redis instance on the GeminiDB console.

Each tenant can create a maximum of 50 GeminiDB Redis instances by default. To request a higher quota, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Usage Notes

This function is now in OBT. To use it, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Procedure

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

Step 2 On the **Instances** page, click .

Step 3 On the displayed page, select instance specifications and click **Next**.

Figure 3-15 Billing mode and basic information

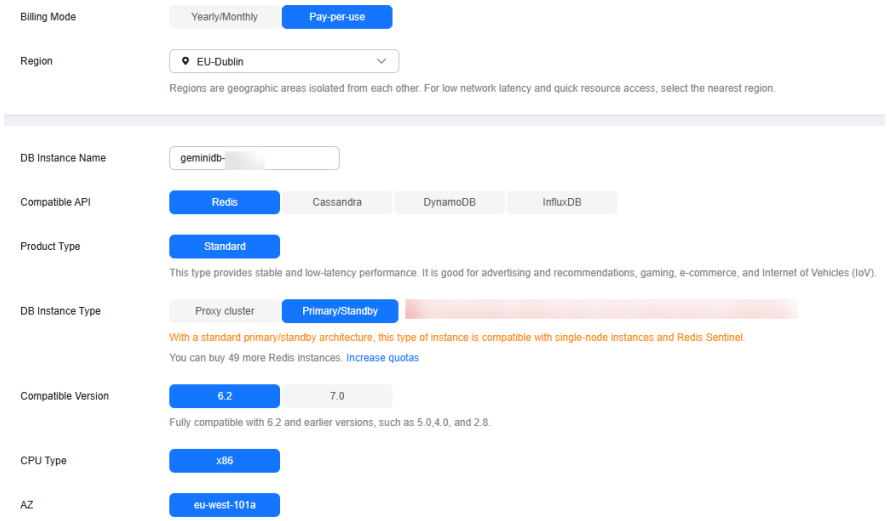


Table 3-9 Billing mode description

Parameter	Description
Billing Mode	Select **Yearly/Monthly** or **Pay-per-use**. • **Yearly/Monthly** – Specify **Required Duration**. The system deducts fees from your account 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. For details, see **Changing a Yearly/Monthly Instance 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 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. – To use an instance for a long time, change its billing mode to yearly/monthly to reduce costs. For details, see **Changing a Pay-per-Use Instance to Yearly/Monthly**.

Table 3-10 Basic information

Parameter	Description
Region	Region where a tenant is located **NOTE** To reduce network latency, select a region nearest from which you will access the instance. 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. • Can contain 4 to 64 characters and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_). You can change the name of an instance after it is created. For details, see **Modifying a GeminiDB Redis Instance Name**.
Compatible API	**Redis** **NOTE** GeminiDB is compatible with mainstream NoSQL databases, including Redis, DynamoDB, Cassandra, and InfluxDB. You can select GeminiDB APIs by following **How Do I Select an API?**

Parameter	Description
Storage Type	**Classic**: classic architecture with decoupled storage and compute
Product Type	**Standard**: Stable and low-latency performance is provided for common scenarios such as advertising and recommendation, gaming, e-commerce, and Internet of Vehicles (IoV).
DB Instance Type	**Primary/Standby** **NOTE** A primary/standby instance is compatible with a standalone Redis node and Redis Sentinel. This instance type is used when hashtags are unavailable.
Compatible Version	7.0, 6.2 (including 6.2.*X*), 5.0, and earlier versions
CPU Type	**x86** **NOTE** x86 CPUs use the Complex Instruction Set Computing (CISC) instruction set. Each instruction can be used to execute low-level hardware operations. Executing these instructions is complex and time-consuming.
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. If there are multiple AZs, you need to select primary and standby AZs. Instances can be deployed in a single AZ or three AZs. ● If low network latency is required, deploy your instance in one AZ. ● To meet disaster recovery requirements, select three AZs and specify primary and standby AZs. – Primary AZ: AZ where a primary node is located – Standby AZ: AZ where a standby node is located

Figure 3-16 Specifications and storage

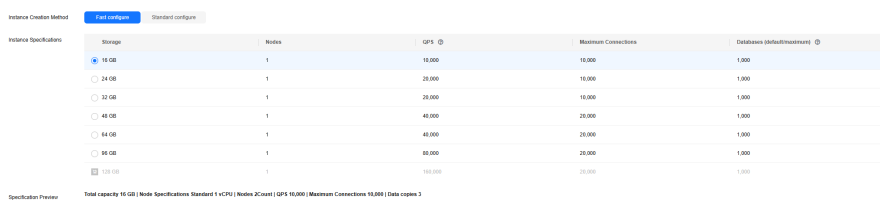


Table 3-11 Specifications and storage

Parameter	Description
Instance Creation Method	Two options are available: • **Fast configure** Provides you with recommended specifications. You can select one of them based on service requirements, without the need to specify the specifications, node quantity, and storage space. **NOTE** – The QPS is only for reference. • **Standard configure** Provides a standard process to configure instance specifications, including specifying the specifications, node quantity, and storage space.
Instance Specifications	You need to specify instance specifications after selecting **Fast configure** for **Instance Creation Method**. Higher CPU specifications provide better performance. Select specifications as needed. For details, see **Instance Specifications**.
Specification Type	You need to select a specification type after selecting **Standard configure** for **Instance Creation Method**. • **Standard**: The default and recommended CPU-to-memory ratio is 1:4, which balances low latency demands with high concurrency requirements.
Node Specifications	You need to select node specifications after selecting **Standard configure** for **Instance Creation Method**. For details, see **Instance Specifications**.
Nodes	The default value is **2**. One is the primary node and the other is standby. If there is a fault, the primary and standby nodes can automatically switch over.
Total Storage Space	You need to specify the storage space after selecting **Standard configure** for **Instance Creation Method**. Higher CPU specifications provide better performance. Select specifications as needed. For details, see **Instance Specifications**.
Specification Preview	After you select instance specifications, the system automatically shows details of the total capacity, node specifications, number of nodes, QPS benchmark, total number of connections, and number of data copies. This helps keep track of the selected instance specifications.

Figure 3-17 Network and database configurations

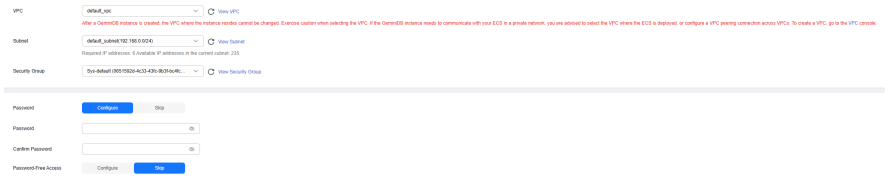


Table 3-12 Network

Parameter	Description
VPC	Virtual private network where your instances are located. A VPC isolates networks for different services. You can select an existing VPC or create a VPC. If there are no VPCs available, the system allocates resources to you by default. **NOTE** • After an instance is created, the VPC where the instance is deployed cannot be changed.
Subnet	A subnet where your instance is created. The subnet provides dedicated and isolated networks, improving network security. **NOTE** An IPv6 subnet cannot be associated with your instance. Select an IPv4 subnet.
Security Group	A security group controls access between 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.

Table 3-13 Database configuration

Parameter	Description
Password	• **Skip**: You can set the database password after creating an instance. • **Configure**: You can set the database password when creating an instance.

Parameter	Description
Password	Password of database administrator **rwuser**: • Must be 8 to 32 characters long. • Can contain at least two types of the following characters: 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**.

Table 3-14 Advanced settings

Parameter	Description
Static Data Encryption	Data is encrypted when stored. • **Disable** • **Enable** Static data encryption improves security but slightly affects database I/O performance. An agency will be created after disk encryption is enabled.
Key Name	You can select an existing key or create a key. The key cannot be disabled, deleted, or frozen when used, or the database becomes unavailable.

Parameter	Description
Tags	The setting is optional. Adding tags helps you better identify and manage your instances. A maximum of 20 tags can be added for each instance. A tag consists of a tag key and a tag value. • A tag key is mandatory if the instance will be tagged. Each tag key is unique for each instance. It can contain 1 to 128 characters, cannot start with **_sys_**, and cannot start or end with a space. Only letters, digits, spaces, and the following special characters are allowed: -_@.:/+= • A tag value is optional if the instance will be tagged. The value can be empty. The value can contain a maximum of 255 characters. Only letters, digits, spaces, and the following special characters are allowed: _.-:+=@/ 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 **Tag Management**.

Step 4 On the displayed page, confirm instance details.

- Yearly/Monthly
  - 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 5 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.

----End

3.4 Instance Connection and Management

3.4.1 Connecting to a GeminiDB Redis Instance

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 3-18 shows the process of connecting to a GeminiDB Redis instance.

Figure 3-18 Connection Methods

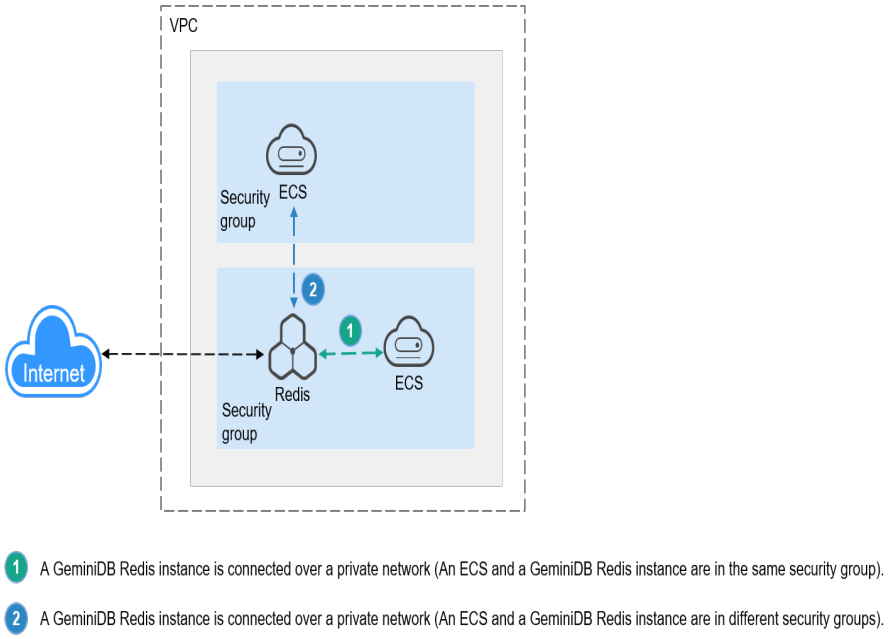


Table 3-15 Connection methods

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

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 the GeminiDB Redis instance by following **Setting Security Group Rules for a GeminiDB Redis Instance**. – 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. **NOTE** Redis Cluster GeminiDB Redis instances cannot be accessed over a public network.

3.4.2 Connecting to a GeminiDB Redis Instance on the DAS Console

DAS makes DB instance management secure and efficient from a web-based console. By default, you have permissions required for remote login. DAS is recommended for connecting to your instance.

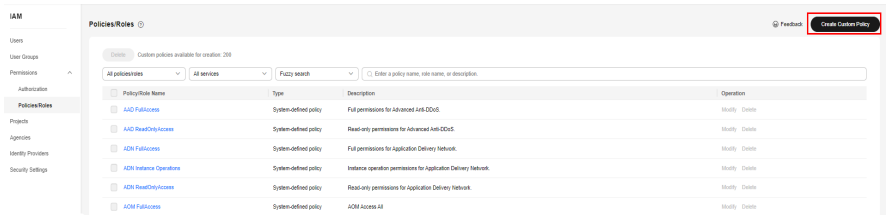
Configuring the Required Permissions

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 3-19 Creating a custom policy



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

Figure 3-20 Configuring a custom policy

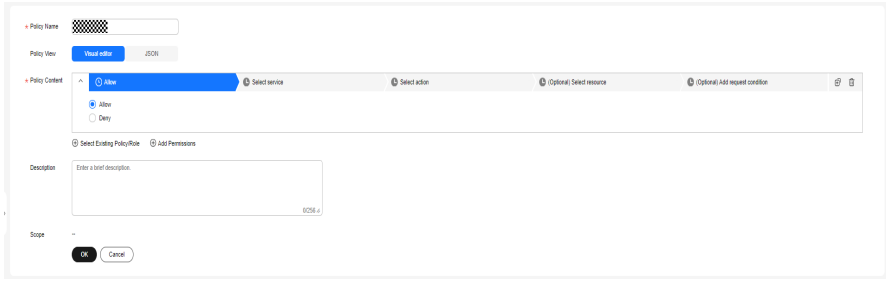


Table 3-16 Custom policy description

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

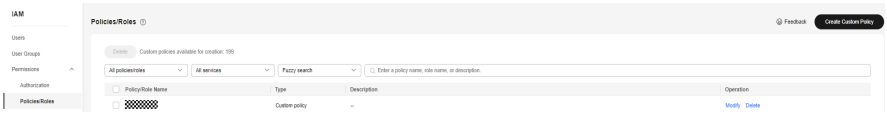
Parameter	Description

Policy Content	Configure the following policy content:
	{
	"Version": "1.1",
	"Statement": [
	{
	"Action": [
	"das:*:*",
	"nosql:instance:list"
	],
	"Effect": "Allow"
	}
	]
	}
	Alternatively, click **Select Existing Policy/Role**, select **DAS FullAccess** as a template, and retain only the DB type information. In this example, retain only **nosql:instance:list**.

Description	Enter a policy description.
Scope	Retain the default settings (project-level service).

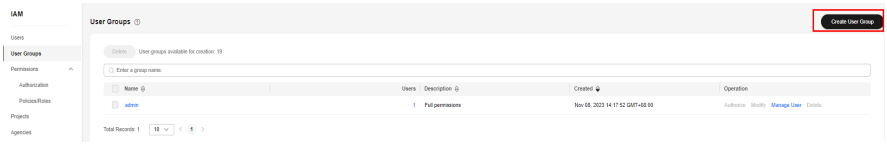
3. Click **OK**. You can then view the created custom policy on the **Permissions** page.

Figure 3-21 Viewing the created policy



4. Create a user group.

Figure 3-22 Creating a user group



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

Figure 3-23 Authorizing the user group using the created custom policy

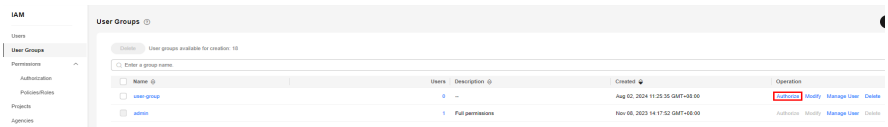
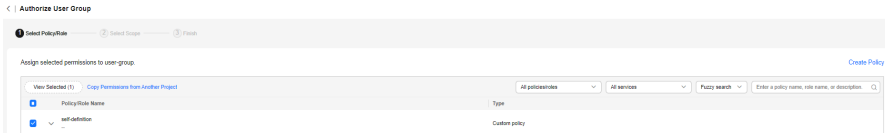
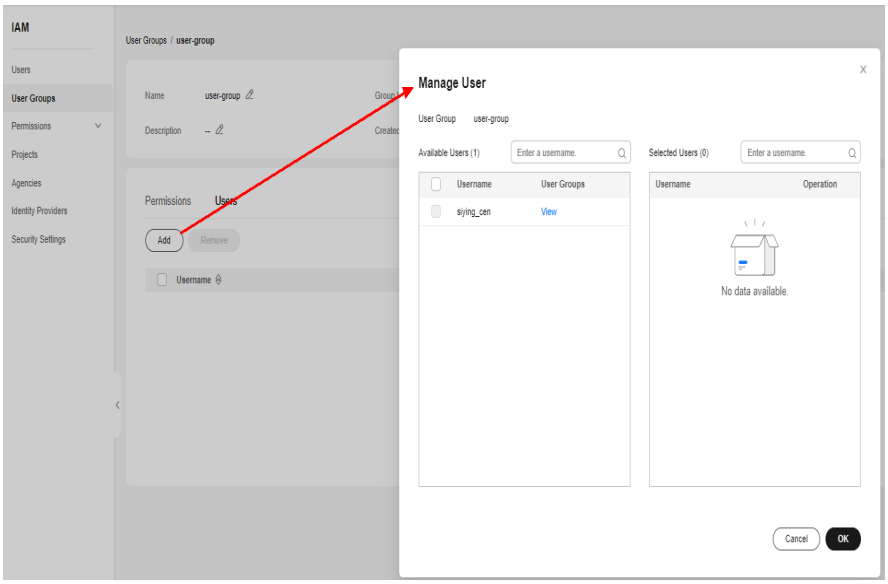


Figure 3-24 Selecting the created custom policy



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

Figure 3-25 Adding users



Prerequisites

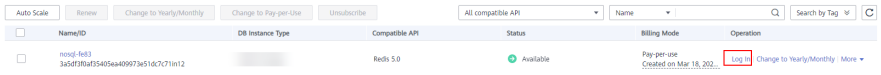
There is an available GeminiDB Redis instance.

Procedure

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

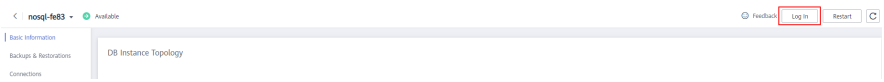
Step 2 In the instance list, locate the target instance and click **Log In** in the **Operation** column.

Figure 3-26 Logging in to a GeminiDB Redis instance



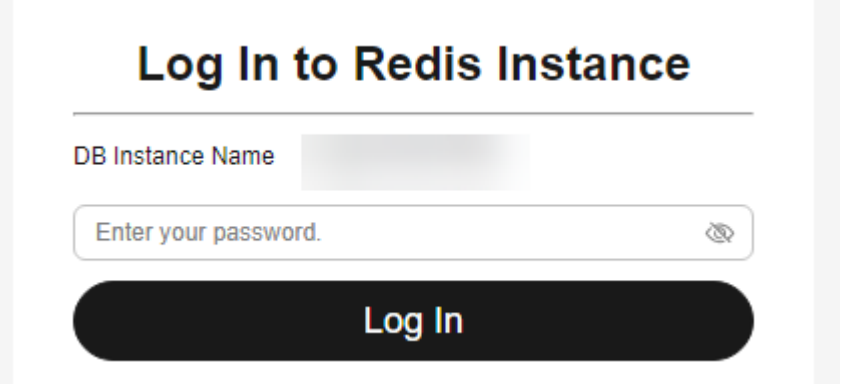
Alternatively, click the instance name to go to the **Basic Information** page. Click **Log In** in the upper right corner of the page.

Figure 3-27 Logging in to a GeminiDB Redis instance



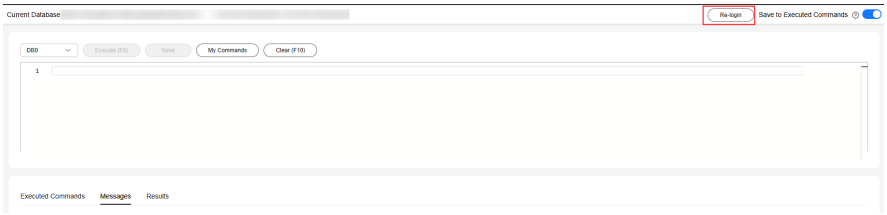
Step 3 Enter the password for logging in to the instance.

Figure 3-28 Login page



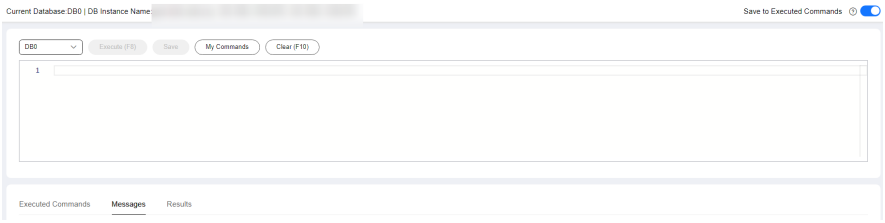
If you need to log in again after the password is reset, click **Re-login** in the upper right corner and use the new password.

Figure 3-29 Re-login



Step 4 Manage relevant databases.

Figure 3-30 Instance homepage

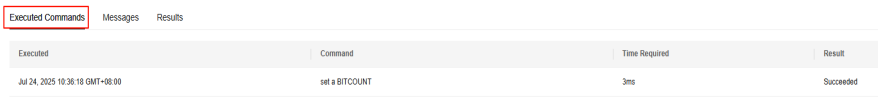


- Save commands to the execution record.

  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 3-31 Executed commands

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.

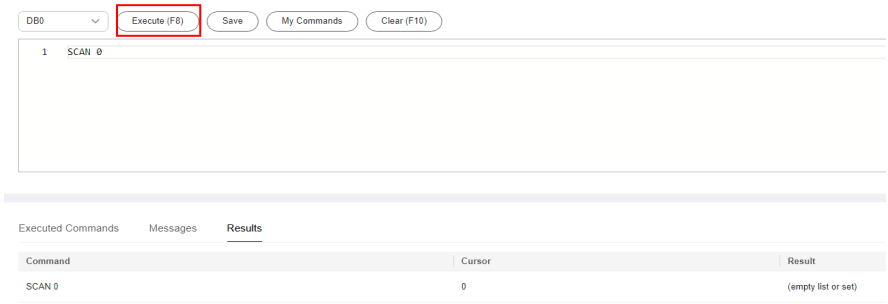
- Execute a command.

  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 3-32 Executing a command

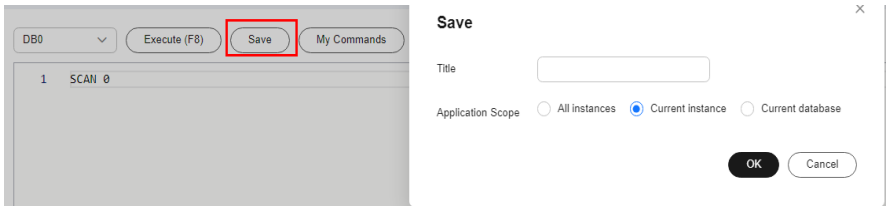


After a command is executed, you can view the execution result on the **Results** page.

- 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 3-33 Save

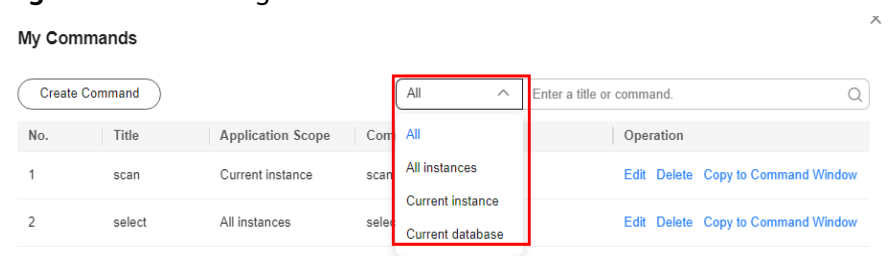


- View my commands.

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

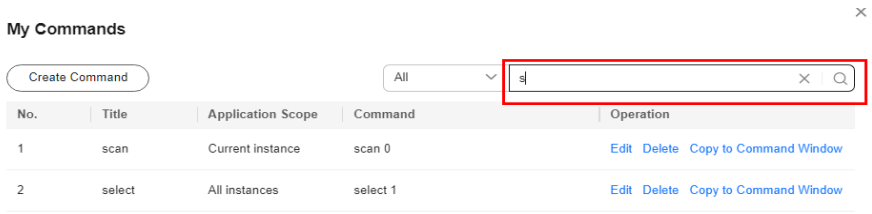
  Set a filter to narrow the scope of commands. If you select **All**, all commands saved in the current account are displayed.

Figure 3-34 Filtering commands



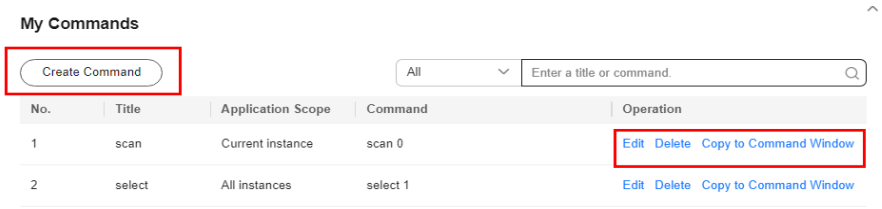
Alternatively, enter a command title or statement in the search box to search for the corresponding command.

Figure 3-35 Searching for a command



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

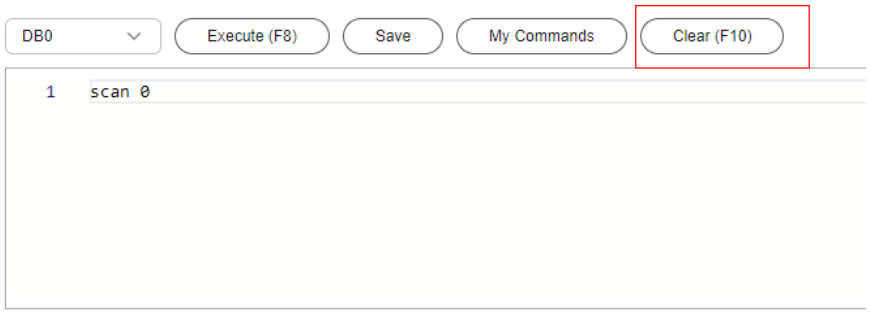
Figure 3-36 Managing a command



- Clear a command.

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

  Figure 3-37 Clearing a command

----End

FAQs

Question: What should I do if the DAS console cannot be redirected after I click **Log In** in the **Operation** column in the instance list or click **Log In** on the **Basic Information** page?

Solution: Set your browser to allow pop-ups and try again.

3.4.3 Connecting to a GeminiDB Redis Instance 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 SPOFs.

Usage Notes

- The target instance must be in the same VPC and subnet as the ECS.
- The instance security group must allow access from the ECS.

  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 **Setting Security Group Rules for 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*.
- 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 package.

tar -xzf redis-6.2.0.tar.gz

Step 4 Open the **src** directory and connect to the DB instance.

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 6379 -a <*DB_PWD*>

Table 3-17 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.
<DB_PORT>	Access port corresponding to the load balancer IP address of the instance. The procedure is as follows: Click the instance name to go to the **Basic Information** page. In the **Connection Information** area, you can see the instance 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.

Usage Notes

- The target instance must be in the same VPC and subnet as the ECS.
- The instance security group must allow access from the ECS.

  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 **Setting Security Group Rules for 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*.
- Download the **Redis client installation package**.

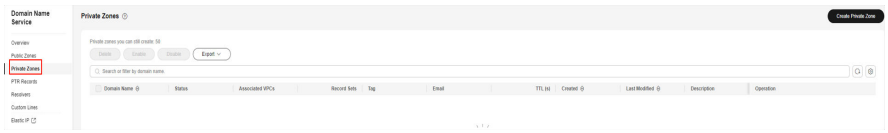
Configuring a Private Domain Name of the GeminiDB Redis Instance

Creating a Private Domain Name

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

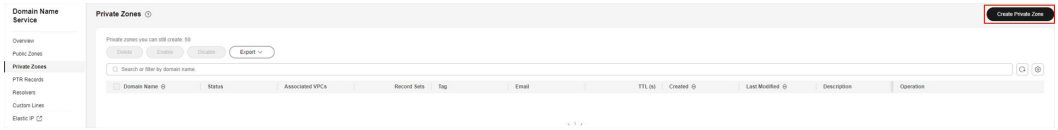
Step 2 On the displayed page, click **Private Zones**.

Figure 3-38 Private zones



Step 3 Click **Create Private Zone**.

Figure 3-39 Creating a private zone



Step 4 Set parameters as prompted.

Figure 3-40 Create Private Zone

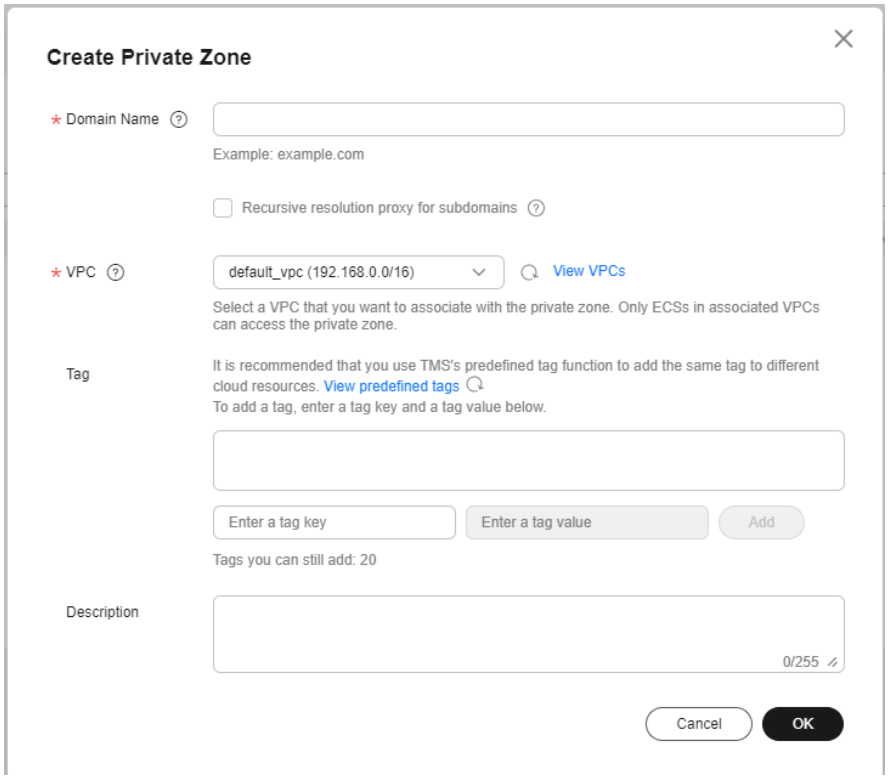


Table 3-18 Parameter description

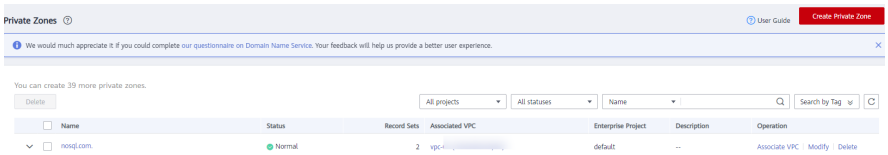
Parameter	Description	Example Value
Domain Name	Domain name of a private zone You can enter a top-level domain that complies with the domain naming rules. For details about the domain name format, see **Domain Name Formats and Structure**.	example.com
VPC	The VPC associated with the private domain name must be the same as the VPC where the GeminiDB Redis instance is located. Otherwise, the private domain name cannot be resolved.	-

Parameter	Description	Example Value
Tag	(Optional) Identifier of a resource. Each tag contains a key and a value. You can add a maximum of 20 tags to a domain name. Key and value naming rules: **Key**: • Cannot be left blank. • Must be unique for each resource. • Can contain a maximum of 128 characters. • Can contain letters, digits, spaces, and special characters _.:=+-@ but cannot start or end with a space or start with _**sys**_. **Value**: • Can contain a maximum of 255 characters. • Can contain letters, digits, spaces, and the following special characters: _.:/=+-@	example_key1 example_value1
Description	(Optional) Description of the zone, which cannot exceed 255 characters	This is a zone example.

Step 5 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 3-41 Private domain name status



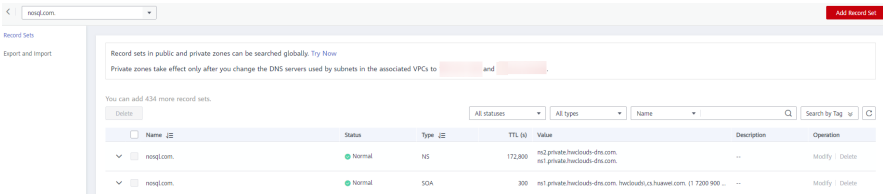
----End

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.

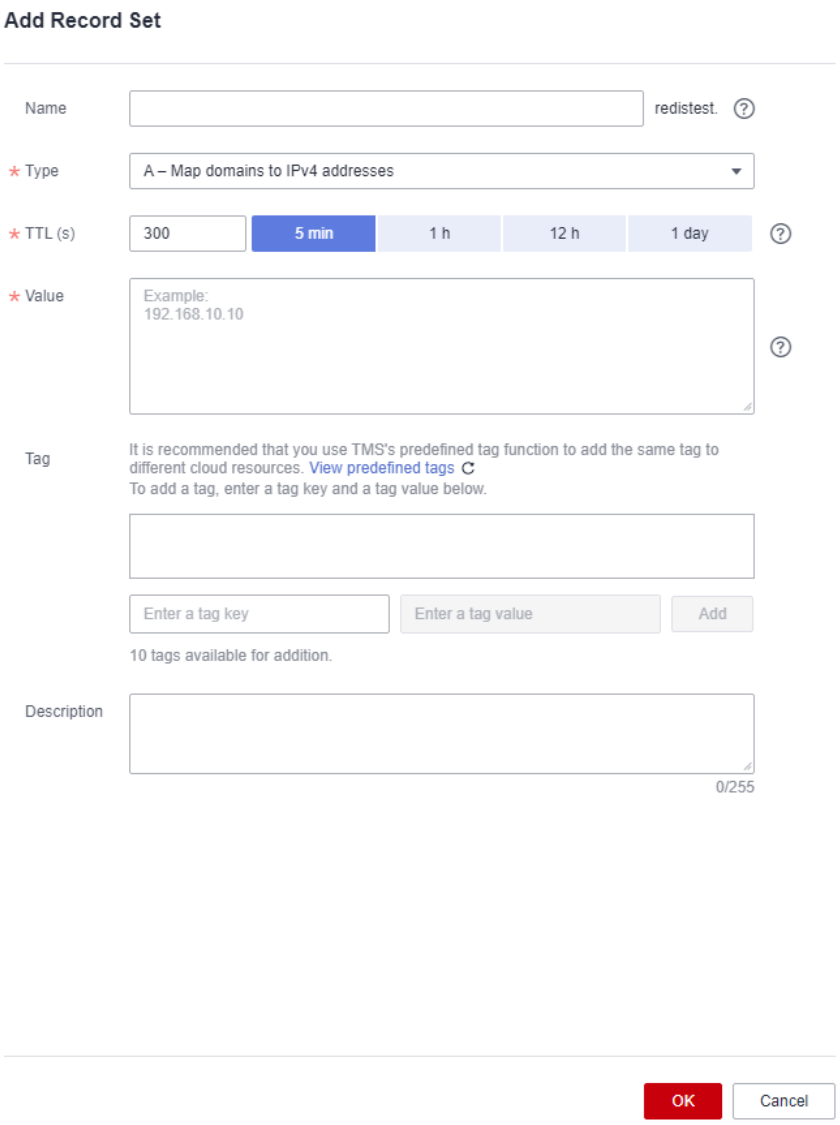
Figure 3-42 Adding a record set



Step 2 In the displayed **Add Record Set** dialog box, set parameters as prompted.

Value: Enter the load balancer IP address.

Figure 3-43 Adding a record set



For details about how to configure parameters, see **Adding Record Sets for a Private Zone**.

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

Logging In to an ECS and Connecting an Instance to the Redis Client

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 package.

tar -xzf redis-6.2.0.tar.gz

Step 4 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 6379 -a *<DB_PWD>*

Table 3-19 Parameter description

Parameter	Description
<DB_Domain_Name>	Private domain name of the instance to be connected. The private domain name is the one created in **Configuring a Private Domain Name of the GeminiDB Redis Instance**.
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the port number, perform the following steps: Click the instance name to go to the **Basic Information** page. In the **Connection Information** area, you can see the instance port.

Parameter	Description
<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.

```
Domain_Name:port>
```

----End

3.4.3.3 Connecting to an Instance Using a Private IP Address

You can connect an ECS to a GeminiDB Redis instance using a private IP address.

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.

Usage Notes

- The target instance must be in the same VPC and subnet as the ECS.
- The instance security group must allow access from the ECS. For details, see **Setting Security Group Rules for a GeminiDB Redis Instance**.
- To connect to a DB instance over a non-SSL connection, SSL must be disabled. For details about how to disable SSL, see **Encrypting Data over SSL for 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*.

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 --no-check-certificate https://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 Open the **src** directory and connect to the DB instance.

```
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 6379 -a <DB_PWD>
```

Table 3-20 Parameter description

Parameter	Description
<DB_HOST>	Private IP address of an instance to be connected. To obtain this IP address, go to the **Instances** page and click the target instance name. In the navigation pane, choose **Node Management**. You can see the private IP address in the **Node Information** area 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 port number, perform the following steps: Click the instance name to go to the **Basic Information** page. In the **Connection Information** area, you can see the instance 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 a GeminiDB Redis Instance 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 a GeminiDB Redis Instance**

over SSL. This section describes how to connect to a GeminiDB Redis instance over a non-SSL connection.

Usage Notes

- To connect to a DB instance over a non-SSL connection, SSL must be disabled. For details about how to disable SSL, see **Encrypting Data over SSL for a GeminiDB Redis Instance**.
- 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 an EIP to a GeminiDB Redis Instance Node** and **Setting Security Group Rules for a GeminiDB Redis Instance**.

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

```
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 6379 -a <DB_PWD>
```


Table 3-21 Parameter description

Parameter	Description
<DB_HOST>	EIP bound to the instance to be connected. To obtain the EIP, go to the **Instances** page and click the target instance name. In the navigation pane, choose **Node Management**. You can see the EIP 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 an EIP to a GeminiDB Redis Instance 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: Click the instance name to go to the **Basic Information** page. In the **Connection Information** area, you can see the instance 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 an EIP to a**

GeminiDB Redis Instance Node and **Setting Security Group Rules for a GeminiDB Redis Instance**.

NOTE

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

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

Configuring a Public Domain Name of the GeminiDB Redis Instance

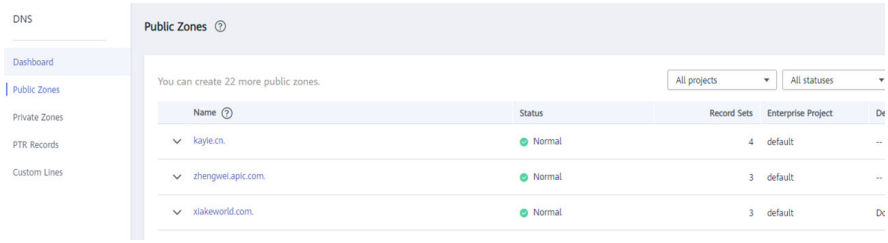
Domain Name Not Created on Huawei Cloud

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

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

Step 2 In the navigation pane, choose **Public Zones**.

Figure 3-44 Public zones



Step 3 In the upper right corner of the page, click **Create Public Zone**.

Step 4 Set the parameters as prompted.

Figure 3-45 Creating a public zone

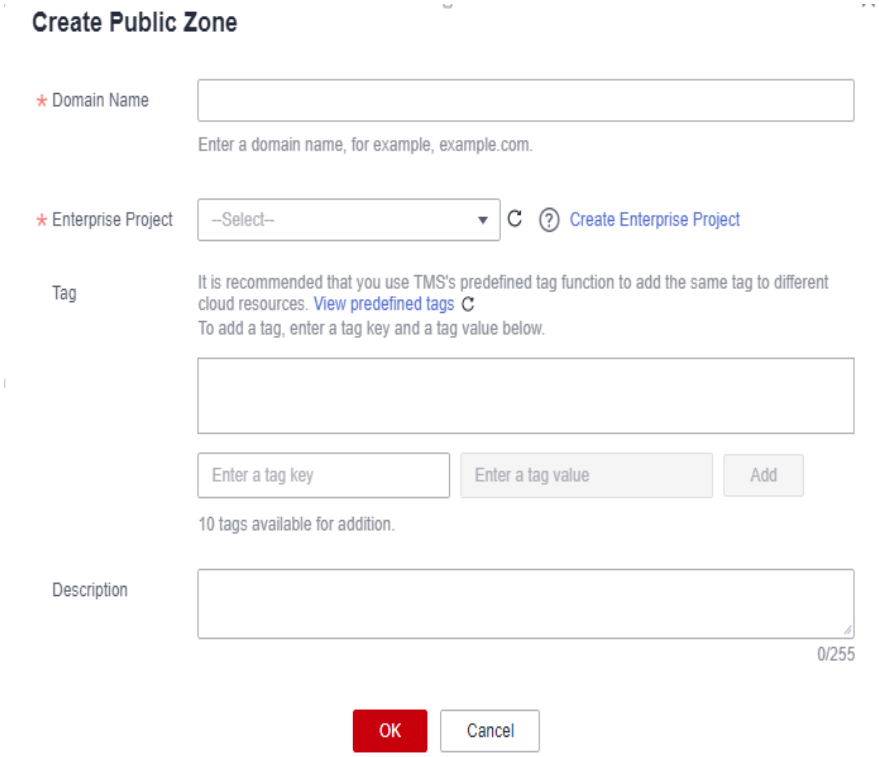


Table 3-22 Public zone parameters

Parameter	Description	Example Value
Domain Name	Domain name you have registered. It 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

Parameter	Description	Example Value
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 notes: ● If you do not manage domain names by enterprise project, select **default**. ● If you manage domain names by enterprise project, select an existing enterprise project.	default
Tag	(Optional) Identifier of a resource. Each tag contains a key and a value. You can add a maximum of 20 tags to a domain name. Key and value naming rules: **Key**: ● Cannot be left blank. ● Must be unique for each resource. ● Can contain a maximum of 128 characters. ● Can contain letters, digits, spaces, and special characters _.:=+-@ but cannot start or end with a space or start with **_sys_**. **Value**: ● Can contain a maximum of 255 characters. ● Can contain letters, digits, spaces, and special characters _.:=+-@ but cannot start or end with a space.	example_key1 example_value1
Description	(Optional) Description of the zone, which cannot exceed 255 characters	This is a zone example.

Step 5 Click **OK**.

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

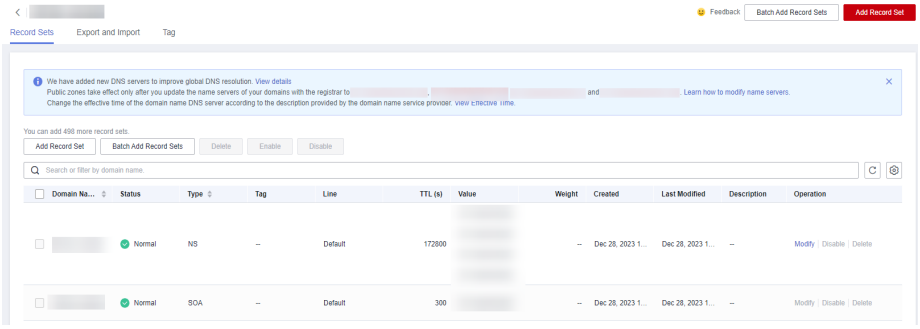
----End

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.

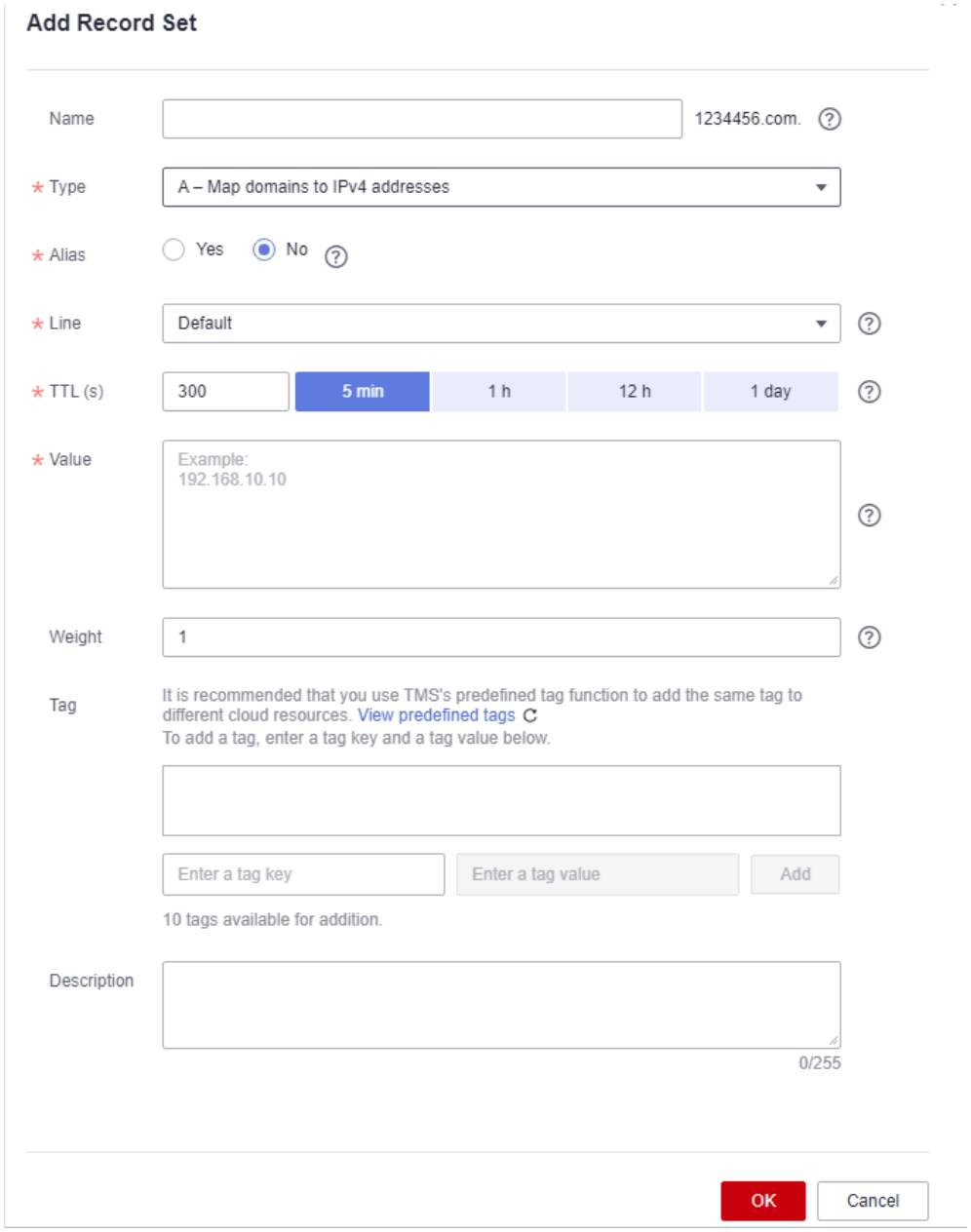
Step 1 Click the name of the public domain name you created. On the displayed page, click **Add Record Set** in the upper right corner.

Figure 3-46 Adding a record set



Step 2 In the displayed **Add Record Set** dialog box, set parameters as prompted.

Figure 3-47 Adding a record set



For details about how to configure parameters, see **Adding Record Sets for a Public Zone**.

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

Logging In to an ECS and Connecting an Instance to the Redis Client

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 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_Domain_Name*> **-p** <*DB_PORT*> **-a** <*DB_PWD*>

Example:

./redis-cli -h redis.com -p 6379 -a *<DB_PWD>*

Table 3-23 Parameter description

Parameter	Description
<DB_Domain_Name>	Public domain name of the instance to be connected. The public domain name is the one created in **Configuring a Public Domain Name of the GeminiDB Redis Instance**.
<DB_PORT>	Port for accessing the target instance. Configure this parameter based on service requirements. To obtain the port number, perform the following steps: Click the instance name to go to the **Basic Information** page. In the **Connection Information** area, you can see the instance 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.

```
Domain_Name:port>
```

----End

3.4.5 Connection Information Management

3.4.5.1 Setting Security Group Rules for a GeminiDB Redis Instance

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.

Usage Notes

- 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 3-24**.

Table 3-24 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**.

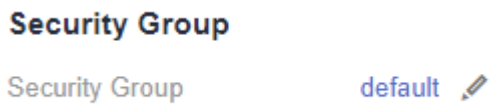
Procedure

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

Step 2 On the **Instances** page, locate the instance that you want to configure security group rules for and click its name.

Step 3 Configure security group rules.

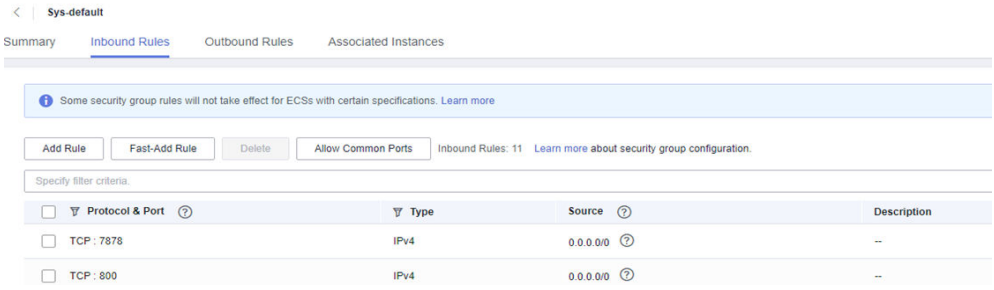
Figure 3-48 Security group



Step 4 Add Inbound Rule

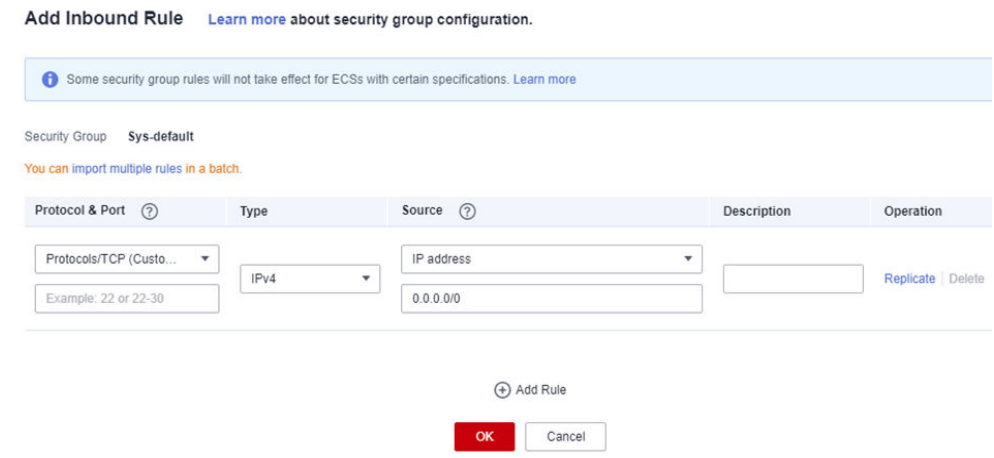
1. Click the **Inbound Rules** tab.

Figure 3-49 Inbound rules



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

Figure 3-50 Adding a rule



3. Add a security group rule as prompted.

Table 3-25 Inbound rule settings

Parameter	Description	Example Value
Protocol & Port	– Network protocol. Currently, GeminiDB Redis instances can be accessed only over TCP. – **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 5 Click **OK**.

----End

3.4.5.2 Viewing the IP Address and Port Number of a GeminiDB Redis Instance

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

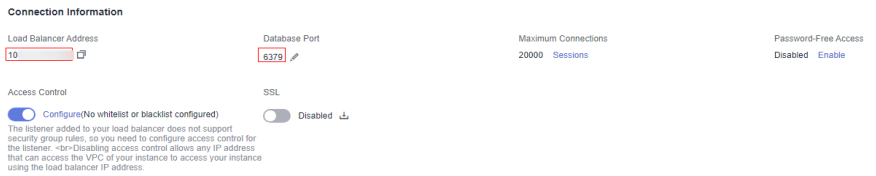
Viewing the Load Balancer IP Address and Port

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

Step 2 On the **Instances** page, locate the instance whose IP address and port you want to view and click its name.

Step 3 In the **Connection Information** area, view the load balancer IP address and corresponding port.

Figure 3-51 Viewing the load balancer IP address and port



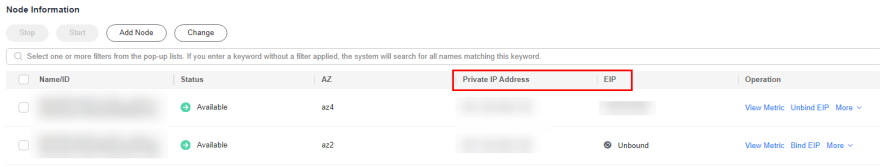
----End

Viewing the Private IP Address or EIP

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

Step 2 On the **Instances** page, locate the instance whose node IP addresses you want to view and click its name.

Figure 3-52 Obtaining IP addresses



----End

Viewing the Port for Accessing Each Instance Node

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

Step 2 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 3-53 Obtaining the port number

----End

3.4.5.3 Binding an EIP to a GeminiDB Redis Instance Node

Scenarios

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

Usage Notes

- 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 On the **Instances** page, click the target instance. The **Basic Information** page is displayed.

Step 3 In the navigation pane, choose **Nodes**. In the **Node Information** area, browse to the target node and click **Bind EIP** in the **Operation** column.

Step 4 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.

Step 5 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 If a node has been bound to an EIP, click the target instance on the **Instances** page. The **Basic Information** page is displayed.

Step 3 In the navigation pane, choose **Nodes**. In the **Node Information** area, browse to the target node and click **Unbind EIP** in the **Operation** column.

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

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

----End

3.4.5.4 Encrypting Data over SSL for a GeminiDB Redis Instance

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.

Usage Notes

- 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 On the **Instances** page, click the target instance.

Step 3 In the **Connection Information** area, click  to enable SSL.

Figure 3-54 Enabling SSL

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

----End

Disabling SSL

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

Step 2 On the **Instances** page, click the target instance.

Step 3 In the **Connection Information** area, click  to disable SSL.

Figure 3-55 Disabling SSL

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

----End

3.4.5.5 Connecting a GeminiDB Redis Instance over 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.

Usage Notes

- The target instance and ECS must be in the same VPC and subnet.
- The instance security group must allow access from the ECS. For details, see **Setting Security Group Rules for a GeminiDB Redis Instance**.
- 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 Open the **src** directory and connect to the DB instance.

- 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>
  ```

  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 6379 -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>
  ```

  Example:

  ```
  ./redis-cli -h 192.168.0.208 -p 6379 -a <DB_PWD> --tls --cacert ./cacert.crt
  ```

Table 3-26 Parameter Description

Parameter	Description
<DB_HOST>	Private IP address of an instance to be connected. To obtain this IP address, go to the **Instances** page and click the target instance name. In the navigation pane, choose **Nodes**. You can see the private IP address in the **Node Information** area 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: Click the instance name to go to the **Basic Information** page. In the **Connection Information** area, you can see the instance port.
<DB_PWD>	Administrator password set when you a GeminiDB Redis instance
<CACERT_PATH>	SSL certificate path

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

```
IP:port>
```

----End

3.4.5.6 Changing the Security Group of a GeminiDB Redis Instance

Scenarios

You can change the security group of a GeminiDB Redis instance.

Precautions

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

Procedure

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

Step 2 On the **Instances** page, locate the instance whose security group you want to change and click its name.

Step 3 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 4 View the modification result.

----End

3.4.5.7 Configuring Private Network Access to a GeminiDB Redis Instance

Scenarios

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

Usage Notes

- 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**.

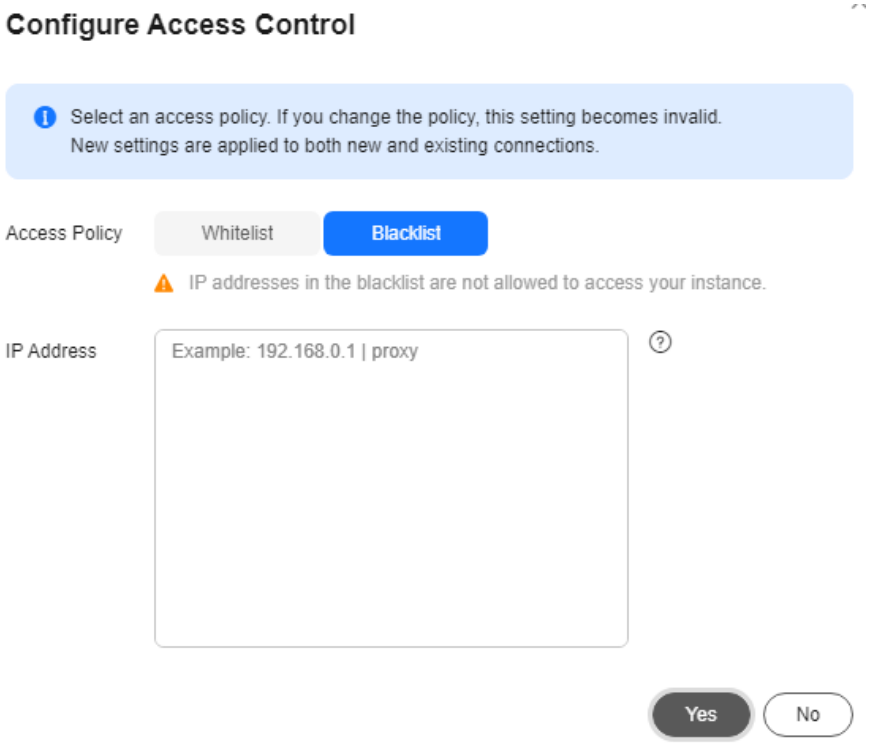
Step 2 On the **Instances** page, click the target instance.

Step 3 In the **Connection Information** area, click next to **Access Control**.

Figure 3-56 Enabling private network access for a load balancer

Step 4 Select **Blacklist** or **Whitelist** and specify IP addresses in that list.

Figure 3-57 Configuring access control



- Blacklist: The blacklist and whitelist 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 On the **Instances** page, click the target instance.

Step 3 In the **Connection Information** area, click  next to **Access Control**. In the displayed dialog box, click **Yes**.

Step 4 Check the load balancer address cannot take effect.

----End

3.5 Data Migration

3.5.1 Migration Solution

This section describes how to migrate services to a GeminiDB Redis instance. If you have any questions about the migration, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console to get technical support.

Usage Notes

Cloud services such as Alibaba Cloud Tair (Redis® OSS-Compatible) and TencentDB for Redis cannot use Huawei Cloud DRS for data migration due to the following factors:

- Some self-developed Redis-like databases are not compatible with the PSync protocol.
- Architecture restrictions: For many cloud vendors, the proxy component is added between users and Redis. PSync is not supported due to the proxy.
- Security restrictions: In native Redis, fork() is used over PSync, which causes memory expansion, user request delay increase, and even out of memory.
- Business strategy: A large number of users use RedisShake to migrate services from the cloud or change the cloud, so PSync is shielded.

Migration Tool

- Data Replication Service (DRS) is used for full and incremental data migration while ensuring data security.
- **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 3-27 Migration scenarios

No.	Source	Destination	Migration Solution
1	Alibaba Cloud Tair (Redis® OSS-Compatible)	GeminiDB Redis	**Migration from Tair (Redis OSS-Compatible) to GeminiDB Redis API**
2	Redis	GeminiDB Redis	**Migrating Data from Redis to GeminiDB Redis API Using RedisShake**
3	RDB file	GeminiDB Redis	**(Recommended) Importing Data to Restore RDB Files to a GeminiDB Redis Instance**
4	Kvrocks	GeminiDB Redis	**Migrating Data from Kvrocks to GeminiDB Redis API**
5	Pika	GeminiDB Redis	**Migrating Data from Pika to GeminiDB Redis API**
6	SSDB	GeminiDB Redis	**Migrating Data from SSDB to GeminiDB Redis API**
7	LevelDB	GeminiDB Redis	**Migrating Data from LevelDB to GeminiDB Redis API**
8	RocksDB	GeminiDB Redis	**Migrating Data from RocksDB to GeminiDB Redis API**
9	AWS ElastiCache for Redis	GeminiDB Redis	**Migrating Data from Amazon ElastiCache for Redis to GeminiDB Redis API**

3.5.2 Migration from Tair (Redis OSS-Compatible) to GeminiDB Redis API

This section describes how to migrate data from Tair (Redis OSS-compatible) to GeminiDB Redis API.

Migration Principles

- You can use DTS (a migration tool of Alibaba Cloud) to migrate data from Tair (Redis OSS-compatible) to GeminiDB Redis API, without worrying about restrictions on the SYNC and PSYNC commands.

Usage Notes

- 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 memory of a GeminiDB Redis instance must be greater than or equal to that of a Tair (Redis OSS-Compatible) instance.
- Ensure that the security groups of the source and destination databases are configured correctly.
- Some types of Tair (Redis OSS-Compatible) instances do not support online full+incremental migration, for example, hybrid-storage instances. You need to scan all source data before migration. For details, see **Fully Scanning Data on and Migrating It from an Open-Source Redis Instance to a GeminiDB Redis Instance**.

Preparations

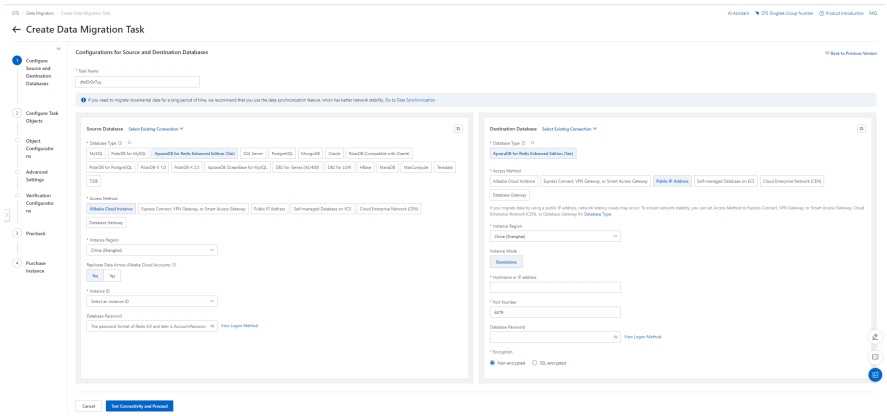
- Migrating data using a public IP address
  - Purchase a Huawei Cloud EIP in advance. The bandwidth must be greater than the source database traffic.
  - Bind the EIP to a Huawei Cloud GeminiDB Redis instance node.
  - When configuring DTS, ensure that the destination database is connected through a public IP address.
- Migrating data via a direct connection
  - Purchase Alibaba Cloud Elastic Compute Service (ECS) in advance and ensure that it can connect to Huawei Cloud GeminiDB Redis API.
  - Forward traffic received by a local port to the destination, so that data can be migrated from Tair (Redis OSS-Compatible) to GeminiDB Redis API.

    ssh -g -L (*Forwarding port***):(***IP address of a load balancer associated with the GeminiDB Redis instance***):(***GeminiDB Redis instance port***) -N -f root@ (***Local ECS IP address***)**
  - When configuring DTS, ensure that the destination is an ECS-hosted database.

Creating a Data Synchronization Task on DTS

Step 1 Select **ApsaraDB for Redis Enhanced Edition (Tair)** as the source. To use an EIP for migration, select **Public IP Address** for the destination database and enter the EIP for **Hostname or IP address**. To use a direct connection, select **Self-managed Database on ECS** for the destination database, enter the ECS IP address for **Hostname or IP address**, and enter a forwarding port for **Port Number**. Enter the database password and click **Test Connectivity and Proceed**. You will proceed with the next step if the test is successful. If there is an exception, check whether the entire migration flow is normal and whether the whitelist configuration is correct.

Figure 3-58 Source and destination database configurations



Step 2 Select **Full Data Migration** or **Full Data Migration + Incremental Data Migration**. Select **Precheck and Report Errors** and select the database to be migrated.

CAUTION

You can also select multiple databases. If only one database can be migrated, select **0**.

Figure 3-59 Database to be migrated

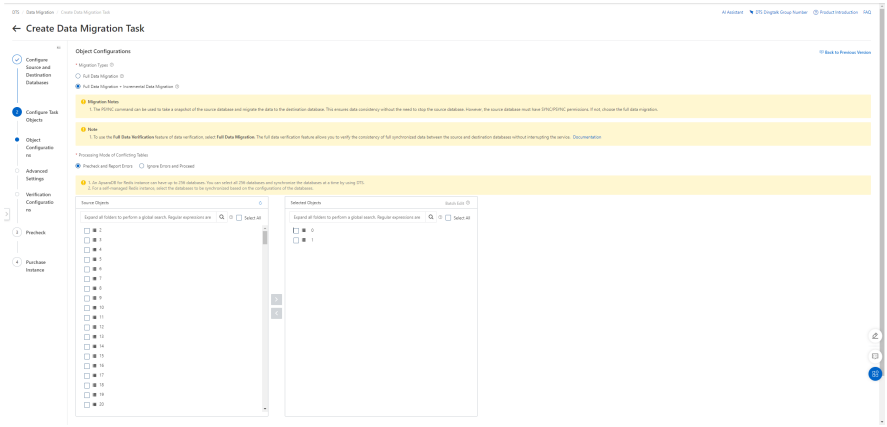
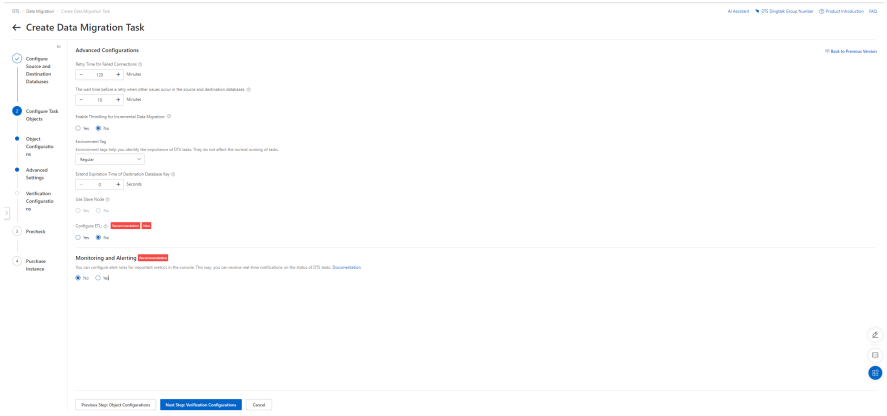
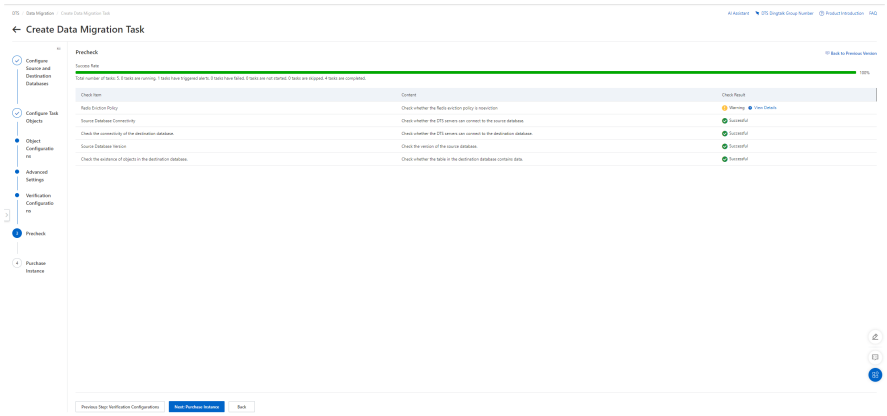


Figure 3-60 Advanced settings



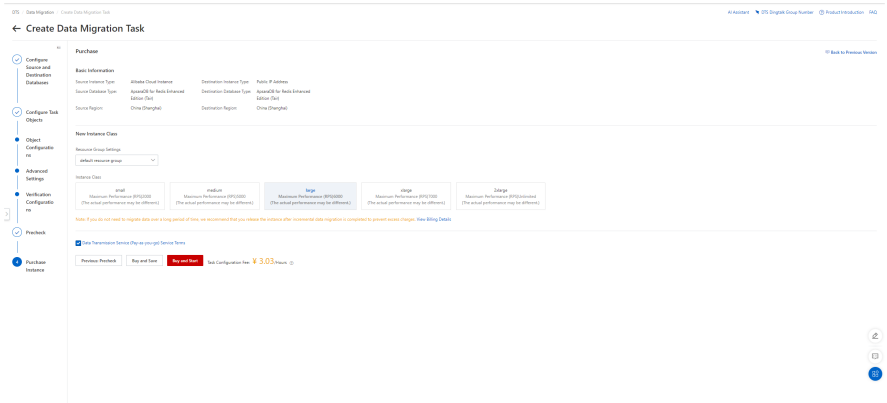
Step 3 After the precheck is complete, click **Next: Purchase Instance**.

Figure 3-61 Precheck



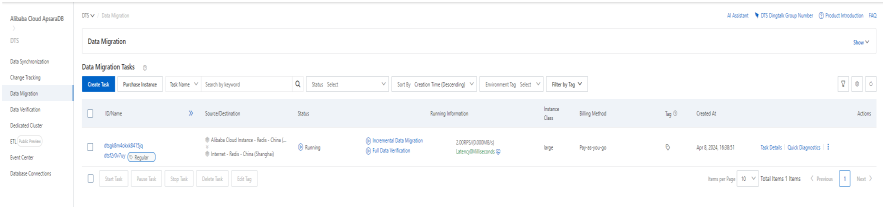
Step 4 Select the bandwidth for the migration and click **Buy and Start**.

Figure 3-62 Bandwidth configuration



Step 5 If **Full Data Migration + Incremental Data Migration** is selected, the migration task will not automatically stop. If there is no delay (0 ms), the full synchronization is complete.

Figure 3-63 Task status

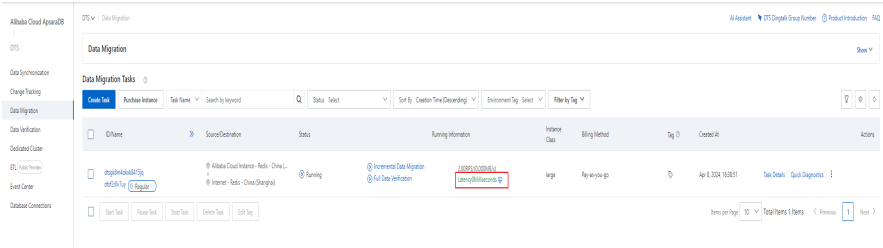


----End

Stopping Data Synchronization on DTS

Step 1 After the migration, stop the data synchronization task.

Figure 3-64 Stopping the data migration task



----End

Verifying Data Consistency After Migration

After the migration is complete, you can check data consistency.

NOTE

- Data has been migrated, or incremental migration has started.
- Redis-full-check must be deployed on the ECS, and the ECS is connected to the source and destination databases.
- During incremental migration, data may be inconsistent due to network latency between the source and destination databases. You are advised to stop writing data to the source and then verify data consistency.
- When Redis is used, an expiration time is usually set for keys. During migration, setting a key expiration time affects data consistency. Data may be inconsistent due to inconsistent expiration time.
- During migration, DTS writes temporary probing keys to Redis on the destination database. Non-service data may be detected during data verification, which is normal.

Procedure

Step 1 Log in to the ECS and ensure it is connected to the source and destination 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 3-28 Parameter description

Parameter	Description	Example Value
-s	Source database address and port	-s 10.0.0.1:6379
-p	Source database password	-
-t	Destination database address and port	-t 10.0.0.2:6379
-a	Destination database password	-
-m	Verification mode: 1. All key-value pairs 2. Value length only 3. Key integrity only 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 a file saving the verification result. The default value is **result.db**.	-d result.db

Step 4 Check 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 there are abnormal keys.


```
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

3.5.3 Migrating Data from Redis to GeminiDB Redis API Using Redis-Shake

You can use DRS or Redis-Shake to migrate data from Redis to GeminiDB Redis API. Redis-Shake is taken as an example in this section.

How Data Is Migrated

This section describes how to use Redis-Shake to migrate data from open-source Redis to GeminiDB Redis API. 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.

Usage Notes

- 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 migrate data from open-source Redis to GeminiDB Redis API, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Migrating Data from a Single-Node or Primary/Standby Redis 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 a host where an open-source Redis instance is deployed

   source.password_raw = *** #** Password for logging in to a source instance

   source.type = standalone # Source instance type

   target.address = <host>:6379 # Destination instance IP address

   target.password_raw = *** #** Password for logging in to a destination instance

   target.version = 5.0 # Version of the destination Redis instance

   target.type = standalone # Destination instance type

   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 exits unexpectedly, you can choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Step 2 Migrate data.

Run the following command to start migration:

./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 a Single-Node or Primary/Standby Redis Instance to a GeminiDB Redis Instance**.

./redis-full-check -s SOURCE_IP:SOURCE_PORT -p SOURCE_PWD -t TARGET_IP:6379 -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 a 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 source hosts

source.type = cluster # Cluster type of the source.

For other steps, see **Migrating Data from a Single-Node or Primary/Standby Redis 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 a Single-Node or Primary/Standby Redis 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 a host where an open-source Redis instance is deployed

   source.password_raw = *** #** Password for logging in to a source instance

   source.type = standalone # Source instance type

   target.address = <host>:6379 # Destination instance IP address

   target.password_raw = *** #** Password for logging in to a destination instance

   target.version = 5.0 # Version of the destination Redis instance

   target.type = standalone # Destination instance type

   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 exits unexpectedly, you can choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Step 2 Migrate data.

Run the following command to start migration:

./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:6379 -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

Verifying Data Consistency After Migration

After the migration is complete, you can check data consistency.

NOTE

- Data has been migrated from Redis, or incremental migration has started.
- Redis-full-check must be deployed on the ECS, and the ECS is connected to the source and destination databases.
- During incremental migration, data may be inconsistent due to network latency between the source and destination databases. You are advised to stop writing data to the source and then verify data consistency.
- When Redis is used, an expiration time is usually set for keys. During migration, setting a key expiration time affects data consistency. Data may be inconsistent due to inconsistent expiration time.
- During migration, DTS writes temporary probing keys to Redis on the destination database. Non-service data may be detected during data verification, which is normal.

Procedure

Step 1 Log in to the ECS and ensure it is connected 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 3-29 Parameter description

Parameter	Description	Example Value
-s	Source Redis database address and port number	-s 10.0.0.1:6379
-p	Password of the source Redis database	-
-t	Destination GeminiDB Redis database address and port number	-t 10.0.0.2:6379
-a	Password of the destination GeminiDB Redis database	-

Parameter	Description	Example Value
-m	Verification mode: 1. All key-value pairs 2. Value length only 3. Key integrity only 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 there are abnormal keys.

```
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

3.5.4 (Recommended) Importing Data to Restore RDB Files to a GeminiDB Redis Instance

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.

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**.

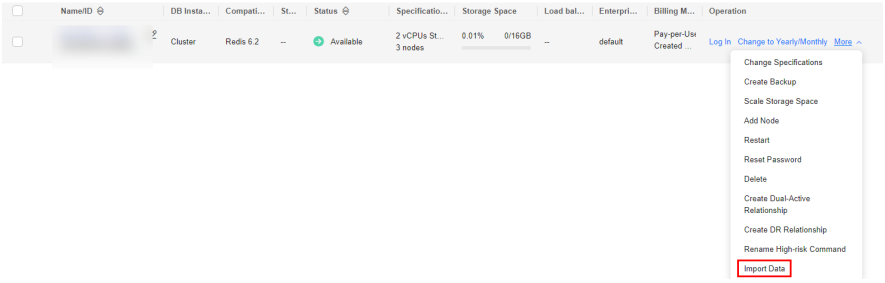
----End

Importing Backups

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

Step 2 On the **Instances** page, locate the target instance and choose **More** > **Import Data** in the **Operation** column.

Figure 3-65 Importing data



Step 3 On the **Data Import** page, specify **OBS Bucket** to which a backup have been uploaded.

Step 4 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 5 Click **Create Now**.

Step 6 Confirm the data import and click **OK**.

CAUTION

Importing data will overwrite data of the current database.

----End

3.5.5 Migrating Data 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 completely compatible with Redis. You can migrate data from Redis instances to GeminiDB Redis instances without refactoring. In addition to adapting to Kvrocks, GeminiDB Redis API also improves management capability and 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.

Usage Notes

- 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 console and contact the customer service.

Verifying Data Consistency After Migration

After the migration is complete, you can check data consistency.

NOTE

- Data has been migrated from Redis, or incremental migration has started.
- Redis-full-check must be deployed on the ECS, and the ECS is connected to the source and destination databases.
- During incremental migration, data may be inconsistent due to network latency between the source and destination databases. You are advised to stop writing data to the source and then verify data consistency.
- When Redis is used, an expiration time is usually set for keys. During migration, setting a key expiration time affects data consistency. Data may be inconsistent due to inconsistent expiration time.
- During migration, DTS writes temporary probing keys to Redis on the destination database. Non-service data may be detected during data verification, which is normal.

Procedure

Step 1 Log in to the ECS and ensure it is connected 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 3-30 Parameter description

Parameter	Description	Example Value
-s	Source Redis database address and port number	-s 10.0.0.1:6379
-p	Password of the source Redis database	-
-t	Destination GeminiDB Redis database address and port number	-t 10.0.0.2:6379
-a	Password of the destination GeminiDB Redis database	-

Parameter	Description	Example Value
-m	Verification mode: 1. All key-value pairs 2. Value length only 3. Key integrity only 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 there are abnormal keys.

```
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

3.5.6 Migrating Data 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

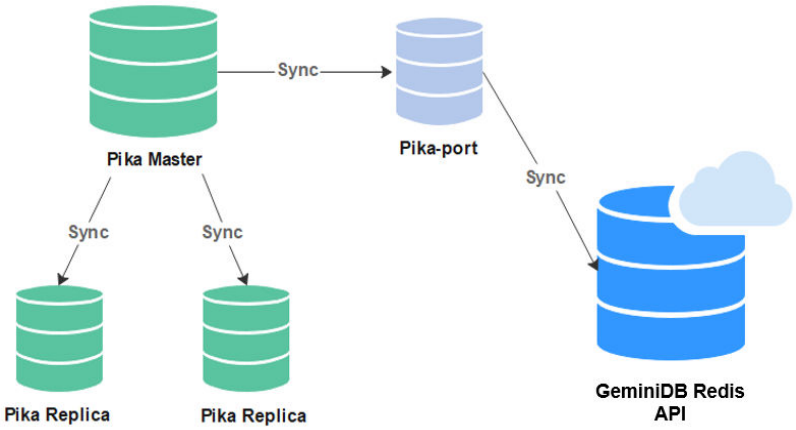
directly, improving read efficiency. RocksDB has been customized to allow the storage capacity to be scaled in seconds. A proxy is used to ensure that upper-layer applications are not affected by underlying sharding or scaling.

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 3-66 Migration Principles



Usage Notes

- 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.

3.5.7 Migrating Data from SSDB to GeminiDB Redis API

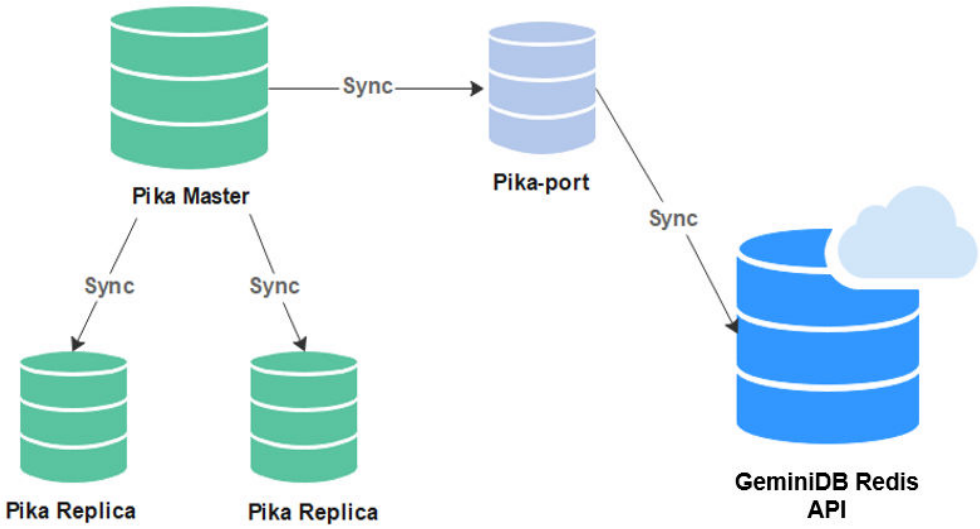
SSDB is a high-performance NoSQL database written in C/C++. It is compatible with Redis APIs and supports multiple data structures, including key-value pairs, hashmap, sorted set, and list. SSDB is a persistent KV storage system and uses leveldb as the underlying storage engine. Its services directly 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 3-67 Migration diagram



Usage Notes

- 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 console and contact the customer service.

Verifying Data Consistency After Migration

After the migration is complete, you can check data consistency.

NOTE

- Data has been migrated from Redis, or incremental migration has started.
- Redis-full-check must be deployed on the ECS, and the ECS is connected to the source and destination databases.
- During incremental migration, data may be inconsistent due to network latency between the source and destination databases. You are advised to stop writing data to the source and then verify data consistency.
- When Redis is used, an expiration time is usually set for keys. During migration, setting a key expiration time affects data consistency. Data may be inconsistent due to inconsistent expiration time.
- During migration, DTS writes temporary probing keys to Redis on the destination database. Non-service data may be detected during data verification, which is normal.

Procedure

Step 1 Log in to the ECS and ensure it is connected 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 3-31 Parameter description

Parameter	Description	Example Value
-s	Source Redis database address and port number	-s 10.0.0.1:6379
-p	Password of the source Redis database	-
-t	Destination GeminiDB Redis database address and port number	-t 10.0.0.2:6379

Parameter	Description	Example Value
-a	Password of the destination GeminiDB Redis database	-
-m	Verification mode: 1. All key-value pairs 2. Value length only 3. Key integrity only 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 there are abnormal keys.

```
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

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.

3.5.8 Migrating Data 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 utilize computing performance of multi-core servers, does not support terabytes of 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.

Usage Notes

- 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 console and contact the 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.

3.5.9 Migrating Data from RocksDB 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.

Usage Notes

- 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 console and contact the customer service.

3.5.10 Migrating Data from Amazon ElastiCache for Redis to GeminiDB Redis API

Migration Principles

After backing up and exporting an RDB file in an Amazon ElastiCache for Redis database, you can use Redis-Shake to restore data to a GeminiDB Redis instance.

Usage Notes

- 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 **Redis-Shake**.

   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 # Destination instance IP address

   target.password_raw = *** # Password for logging in to a destination instance

   target.version = 5.0 # Version of the destination Redis instance

   target.type = standalone # Destination instance type

   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 exits unexpectedly, you can choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Step 2 Migrate data.

Run the following command to start migration:

./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

Verifying Data Consistency After Migration

After the migration is complete, you can check data consistency.

> **NOTE**
>
> - Data has been migrated from Redis, or incremental migration has started.
> - Redis-full-check must be deployed on the ECS, and the ECS is connected to the source and destination databases.
> - During incremental migration, data may be inconsistent due to network latency between the source and destination databases. You are advised to stop writing data to the source and then verify data consistency.
> - When Redis is used, an expiration time is usually set for keys. During migration, setting a key expiration time affects data consistency. Data may be inconsistent due to inconsistent expiration time.
> - During migration, DTS writes temporary probing keys to Redis on the destination database. Non-service data may be detected during data verification, which is normal.

Procedure

Step 1 Log in to the ECS and ensure it is connected 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**

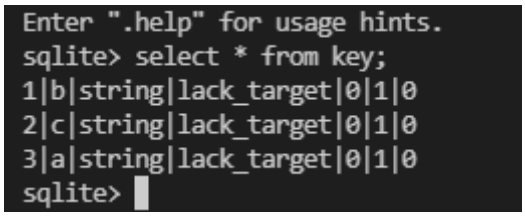
Table 3-32 Parameter description

Parameter	Description	Example Value
-s	Source Redis database address and port number	-s 10.0.0.1:6379
-p	Password of the source Redis database	-
-t	Destination GeminiDB Redis database address and port number	-t 10.0.0.2:6379
-a	Password of the destination GeminiDB Redis database	-
-m	Verification mode: 1. All key-value pairs 2. Value length only 3. Key integrity only 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 there are abnormal keys.



----End

3.5.11 Verifying Redis Data Consistency After Migration

After the migration is complete, you can check the consistency of Redis data.

Usage Notes

- The Redis migration has been completed, or the incremental migration has started.
- Redis-full-check must be deployed on the ECS, and the ECS is connected to the source and destination databases.
- 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 3-33 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.	-

Parameter	Description	Example Value
-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

3.6 Instance Lifecycle Management

3.6.1 Restarting a GeminiDB Redis Instance

Scenarios

You may need to restart an instance for routine maintenance.

Usage Notes

- Only instances in the **Available**, **Abnormal**, or **Checking restoration** statuses 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 On the **Instances** page, locate the target instance and click **Restart** or choose **More** > **Restart** in the **Operation** column.

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 3 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 4 In the displayed dialog box, click **Yes** or **Immediate**.

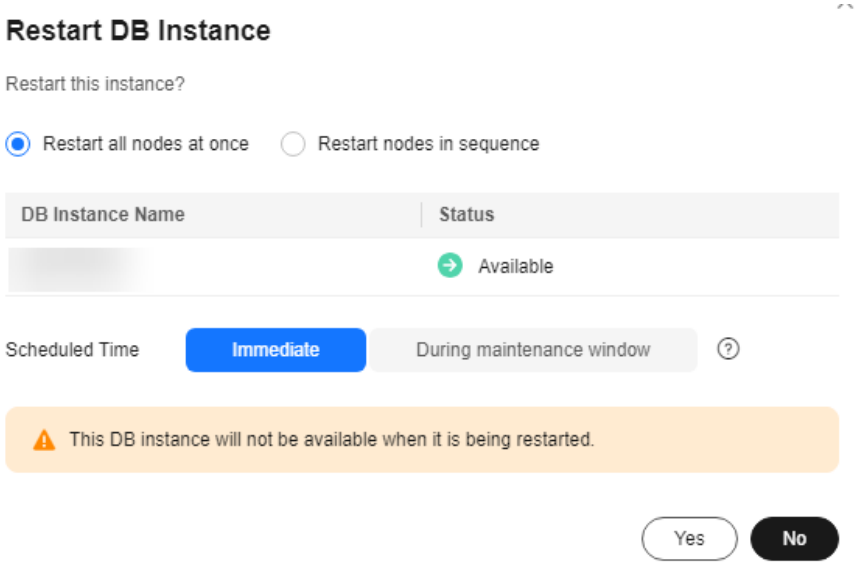
- Instance with classic storage

  For a GeminiDB Redis instance with classic storage, you can restart nodes one by one or all at once.

  NOTE

  Restarting nodes all at once will interrupt services for about 10 to 20 minutes and is suitable for maintenance operations during a temporary outage. Restarting nodes one by one will interrupt services for 3 to 5 seconds each time, which has less impact.

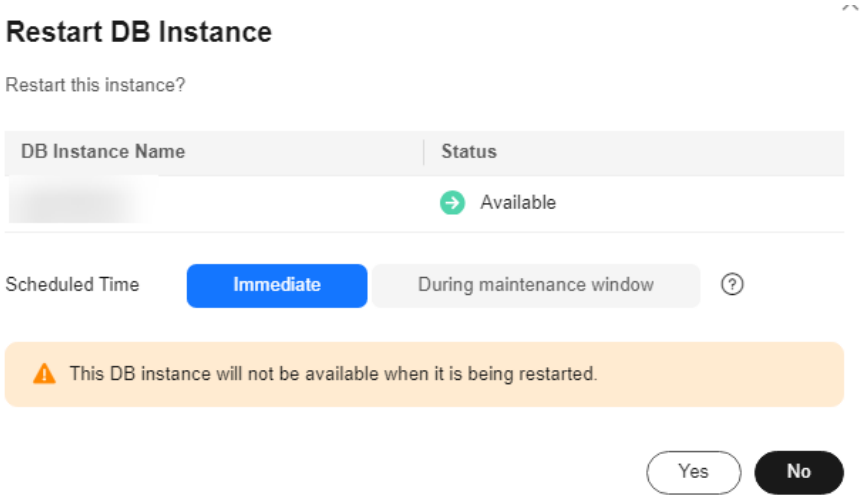
Figure 3-68 Restarting an instance



- Instance with cloud native storage

  For GeminiDB Redis instances with cloud native storage, click **Yes** or **Immediate**.

Figure 3-69 Restarting an instance



----End

3.6.2 Exporting Instance Information

Scenarios

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

Exporting All Instance Information

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

Step 2 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 3 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 instance, unsubscribe from it. For details, see **Unsubscribing 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 On the **Instances** page, locate the instance that you want to delete and in the **Operation** column choose **Delete** or **More** > **Delete**.

Step 3 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 4 In the displayed dialog box, click **Yes**.

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

----End

3.6.4 Recycling a GeminiDB Redis Instance

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

Usage Notes

- 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.
- Currently, you can put a maximum of 100 instances into the recycle bin.
- If you delete an instance of full storage, the deleted instance will not be moved to the recycle bin.
- After an instance is deleted, the most recent automated full backup (if no automated full backup is available one day ago, the latest one is retained) is retained and a full backup is performed. You can select any backup file to restore the instance data.

Modifying the Recycling Policy

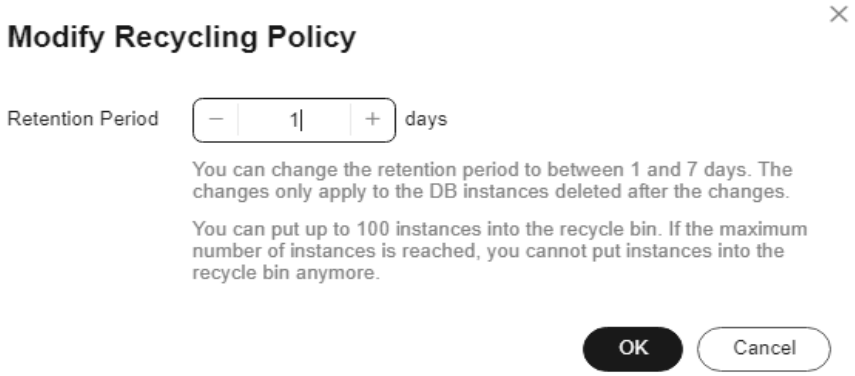
NOTE

You can modify the retention period, and the modifications are only applied to instances deleted after the retention period is modified, so exercise caution when performing this operation.

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

Step 2 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 3-70 Modifying the recycling policy



----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 On the **Recycling Bin** page, locate the target instance and click **Rebuild** in the **Operation** column.

Figure 3-71 Rebuilding an instance

Step 3 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 Modifying Instance Settings

3.7.1 Modifying a GeminiDB Redis Instance Name

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 On the **Instances** page, click  next to the target instance name and change it.

- 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 3 View the results on the **Instances** page.

----End

Method 2

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

Step 2 On the **Instances** page, click the target instance. The **Basic Information** page is displayed.

Step 3 In the **Instance Information** area on the **Basic Information** page, click  next to **DB Instance Name** and change the instance name.

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

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

----End

3.7.2 Changing the Administrator Password of a GeminiDB Redis Database

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

Step 3 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 4 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 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 3 In the **DB Information** area, click **Reset Password** in the **Administrator** field.

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

3.7.3 Changing vCPUs and Memory

Scenarios

You can change the vCPUs and memory of all nodes. You can change the vCPUs and memory of your instance as needed. If an instance is overloaded and compute resources need to be added urgently, you are advised to add compute nodes first.

Usage Notes

- 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.
- After specifications of a standard instance with cloud native storage are changed, the system automatically adjusts the storage to the number of shards multiplied by shard specifications (GB).

Procedure

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

Step 2 On the **Instances** page, locate the target instance and choose **More** > **Change Specifications** in the **Operation** column.

-

In the **DB Information** area on the **Basic Information** page, click **Change** under **Node Specifications**.

Step 3 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 10 to 20 minutes. Exercise caution when performing this operation. For your online production services, you are advised to perform the change online.

Step 4 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 5 View the change results.

In the **DB Information** area on the **Basic Information** page, you can see the new specifications.

----End

3.7.4 Adding Instance 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 Instance Nodes**.

Usage Notes

- 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, nodes can be added only for proxy cluster and Redis Cluster instances.

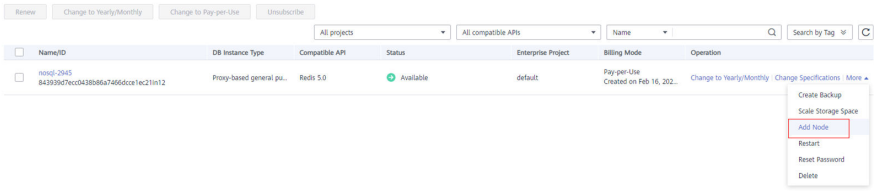
- Currently, a maximum of 36 nodes are supported. To add more, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Method 1

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

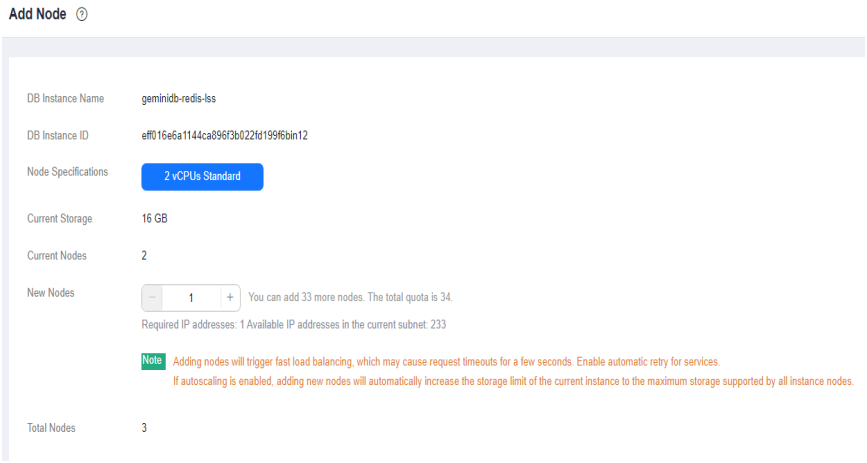
Step 2 On the **Instances** page, locate the instance which you want to add nodes for, click its name, and choose **More** > **Add Node** in the **Operation** column.

Figure 3-72 Adding nodes



Step 3 On the **Add Node** page, specify the number of nodes to be added and view the storage of the instance.

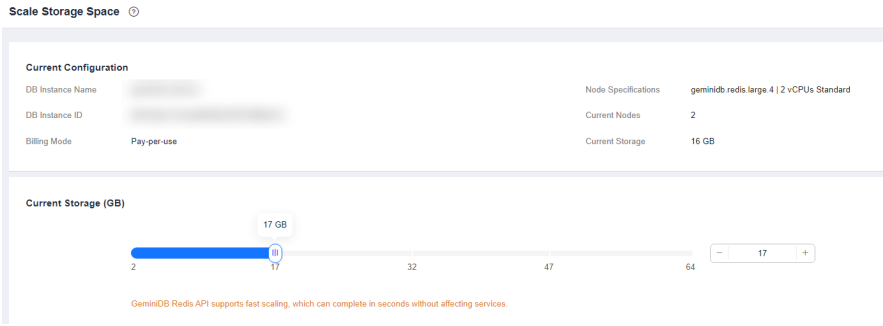
- If the storage capacity is sufficient, click **Next** and go to **Step 6**.
- If the storage capacity is insufficient, click **Next** and go to **Step 4**.



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

Step 4 On the **Scale Storage Space** page, select your target storage capacity and click **Next**.

Figure 3-73 Storage change



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

Step 6 On the displayed page, confirm the node configuration details.

- Yearly/Monthly
  - 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**.

----End

Method 2

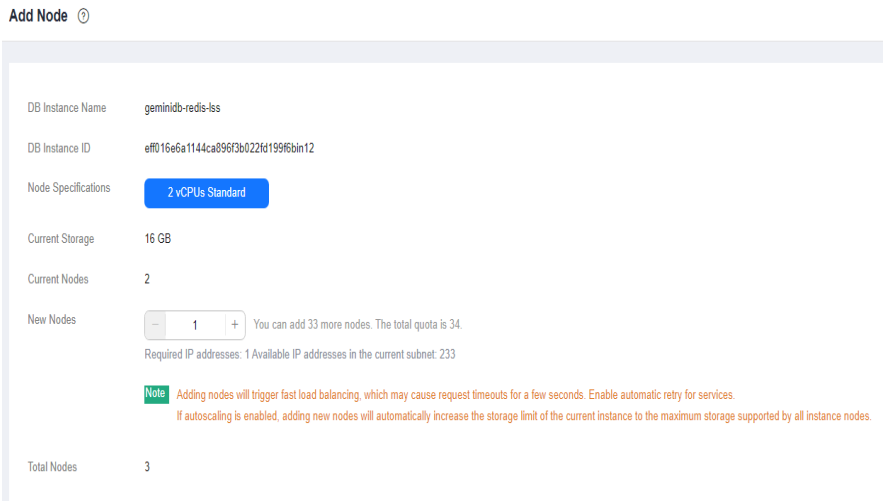
Step 1 Log in to the **GeminiDB console**.

Step 2 On the **Instances** page, locate the instance you want to add nodes for and click its name.

Step 3 In the navigation pane, choose **Node Management**.

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

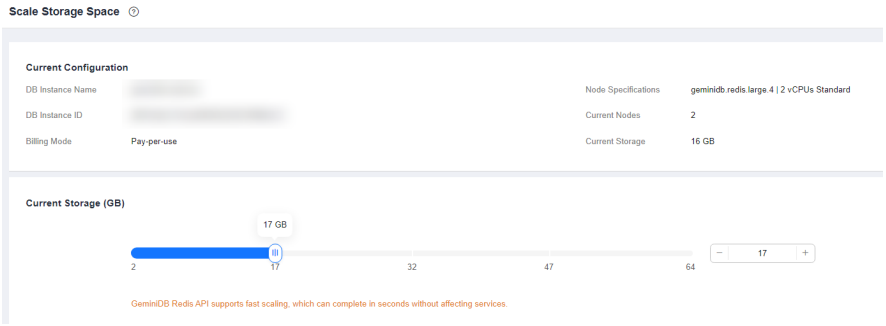
- If the storage is sufficient, click **Next** and go to **Step 7**.
- If the storage is insufficient, click **Next** and go to **Step 5**.



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 3-74 Storage change



Step 6 After the storage capacity is expanded, go to **Step 2** to add nodes again.

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

- Yearly/Monthly
  - 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**.

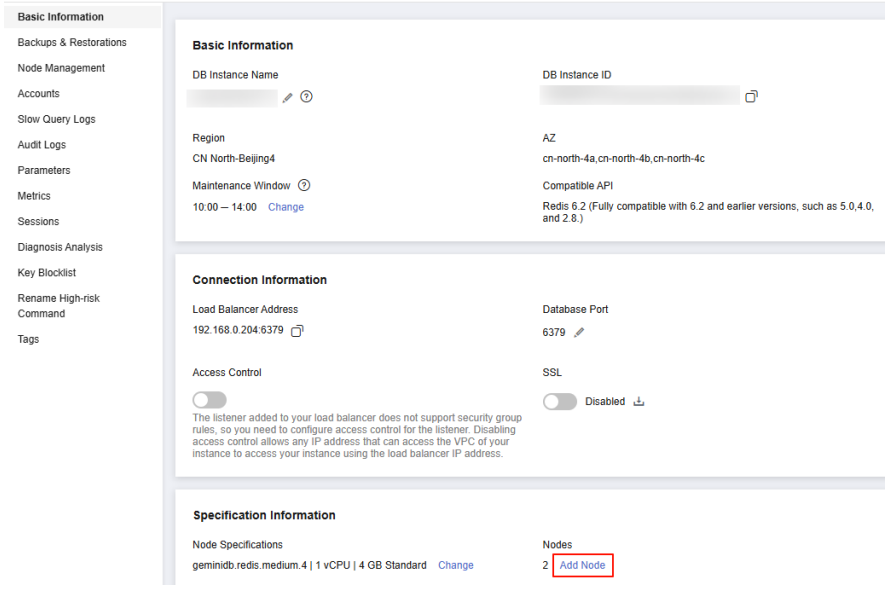
----End

Method 3

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

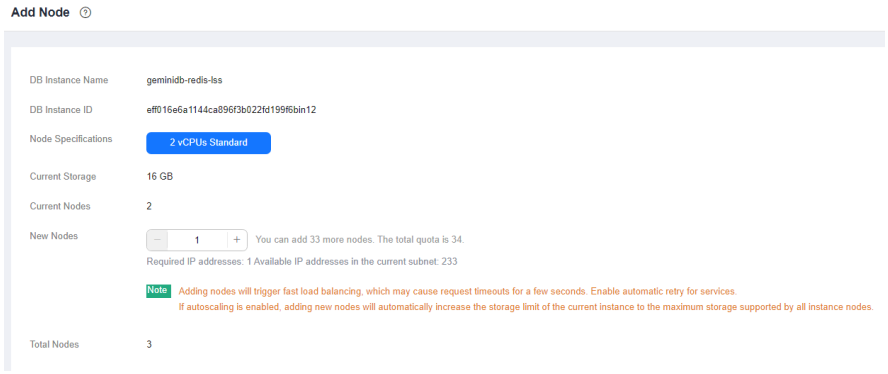
Step 2 On the **Instances** page, click the target instance.

Step 3 In the **Specification Information** area on the **Basic Information** page, click **Add Node**.



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

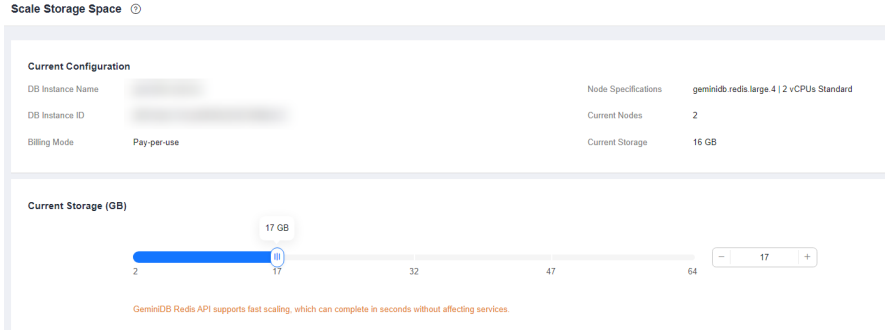
- If the storage is sufficient, click **Next** and go to **Step 7**.
- If the storage is insufficient, click **Next** and go to **Step 5**.



By default, specifications of the new node are the same as the instance specifications and cannot be modified.

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

Figure 3-75 Storage change



Step 6 After the storage is scaled up, go to **Step 2** to add nodes again.

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

- Yearly/Monthly
  - 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**.

----End

3.7.5 Adding Instance Shards

Scenarios

This section describes how to add shards for an instance to suit your service requirements.

Precautions

- Shards can be added only for standard instances with cloud native storage.
- Adding shards will trigger fast load balancing, which may cause a request timeout for a few seconds. Enable automatic retry for services.
- You can only add shards when the instance status is **Available** or **Checking restoration**.
- A DB instance cannot be deleted when one or more shards are being added.
- After shards are successfully added, the system automatically expands the storage capacity (*Number of new shards* x *Shard specification (GB)*).
- Currently, shards can be added only for proxy cluster instances.

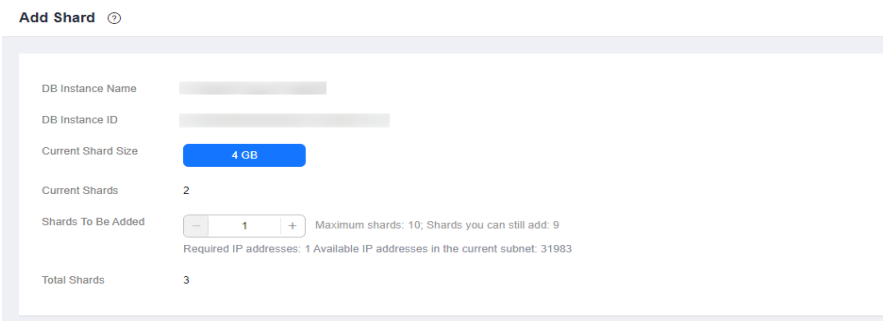
Procedure

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

Step 2 On the **Instances** page, click the target instance.

Step 3 In the navigation pane, choose **Shard Management**.

Step 4 Click **Add Shard**. On the displayed page, select the number of shards to be added.



New shards are of the same specifications as existing shards. Once a shard is added, its specification cannot be changed.

Step 5 On the displayed page, confirm the shard configuration.

- Pay-per-use
  - To modify the configuration, click **Previous** to go back to the page where you specify details.
  - If you do not need to modify the configuration, click **Submit**.

----End

3.7.6 Deleting Instance Nodes

Scenarios

You can delete nodes of pay-per-use or yearly/monthly instances to release resources.

Usage Notes

- 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***.
- Deleting nodes of yearly/monthly instances is now in OBT. To use it, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Procedure

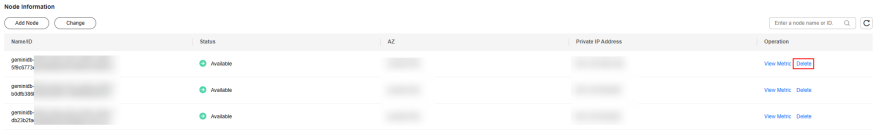
Step 1 Log in to the **GeminiDB console**.

Step 2 On the **Instances** page, click the target instance. The **Basic Information** page is displayed.

Step 3 In the navigation pane, choose **Node Management**. On the displayed page, check the node you want to delete.

- Pay-per-use
  - In the **Node Information** area, locate the target node and click **Delete** in the **Operation** column.

    Figure 3-76 Node information



Step 4 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 5 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 Manually Scaling Up Storage Space

Scenarios

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

Usage Notes

- Storage space can only be scaled up.
- When the disk usage of a GeminiDB Redis instance exceeds 95%, the instance becomes read-only. 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.**
- Cloud native storage of standard instances cannot be changed. You can **add shards** or **upgrade instance specifications** instead.

Setting an Instance Status to 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 3-34 Setting an instance status 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	• 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.

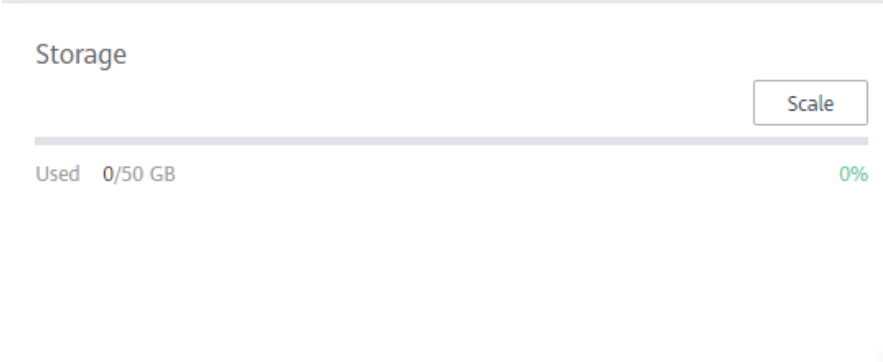
Method 1

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

Step 2 On the **Instances** page, click the target instance.

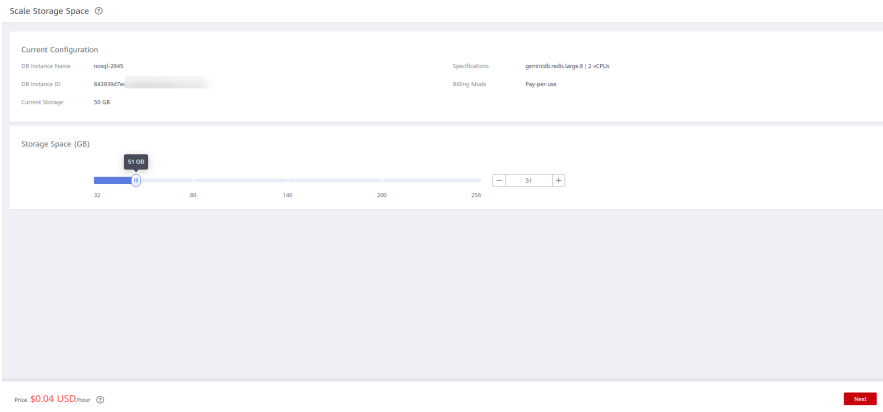
Step 3 In the **Specification Information** area on the **Basic Information** page, click **Scale** for an instance.

Figure 3-77 Scaling storage



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

Figure 3-78 Scaling storage



- To scale up classic storage, you need to add at least 1 GB each time. The value must be an integer.
- To scale up cloud native storage, you need to add at least 10 GB each time. The value must be an integer multiple of 10.

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

- Yearly/Monthly
  - 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 results.

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

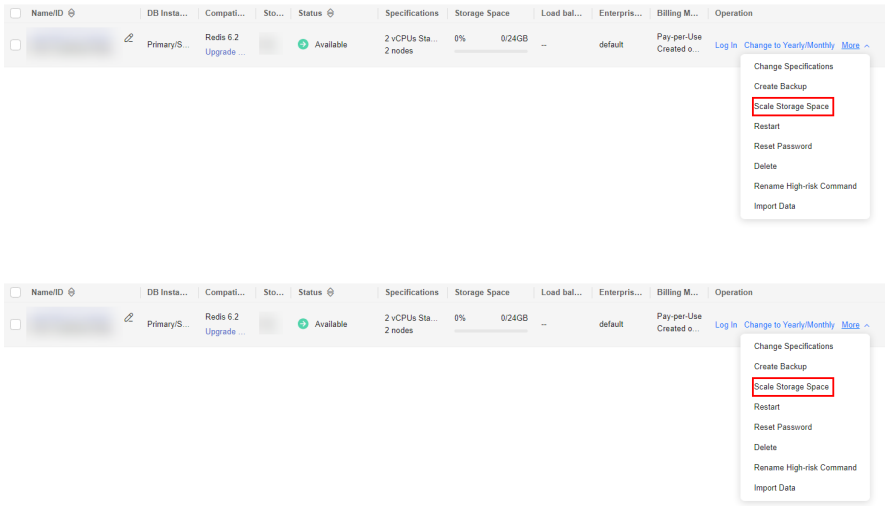
----End

Method 2

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

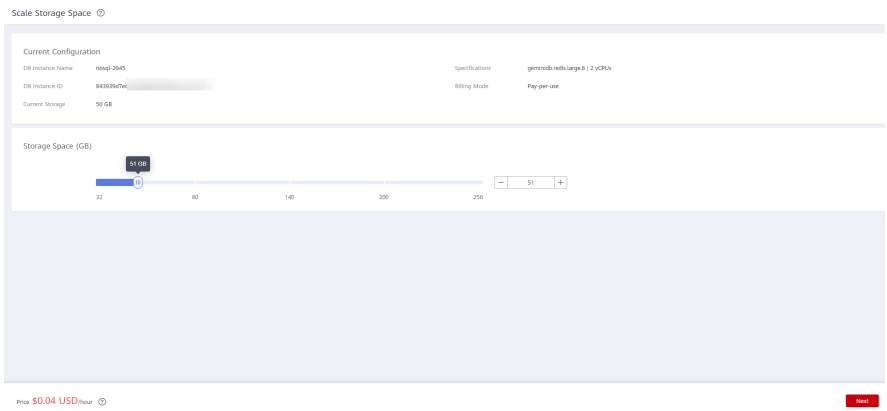
Step 2 On the **Instances** page, locate the instance whose storage space you want to scale and choose **More** > **Scale Storage Space** in the **Operation** column.

Figure 3-79 Scale Storage Space



Step 3 On the displayed page, specify the new storage space and click **Next**.

Figure 3-80 Scaling storage



- To scale up classic storage, you need to add at least 1 GB each time. The value must be an integer.
- To scale up cloud native storage, you need to add at least 10 GB each time. The value must be an integer multiple of 10.

Step 4 On the displayed page, confirm the storage space.

- Yearly/Monthly
  - 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 5 Check the results.

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

----End

3.8 Data Backup

3.8.1 Overview

You can create backups for GeminiDB Redis instances to ensure data reliability. After an instance is deleted, its automated backups are also deleted while manual backups are retained. The backups cannot be exported. GeminiDB Redis instances support only full backups.

Usage Notes

Backing up data consumes a few CPUs. Uploading backup files to OBS occupies bandwidth of compute nodes, causing slight latency and jitter.

Backup Methods

Both automatic backup and manual backup are supported.

- Automated backup

  You can **modify a 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. A 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 a backup.

Table 3-35 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 and 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. 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.

3.8.2 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.

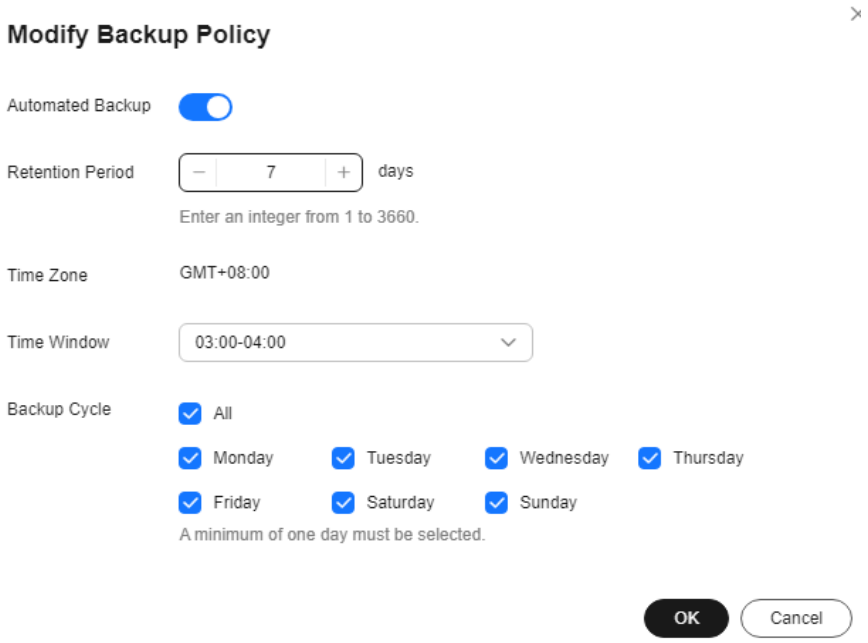
Configuring an Automated Backup Policy

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 3-81 Modifying a backup policy



- **Retention Period**: Automated backup files are saved for seven days by default. The retention period ranges from 1 to 3660 days. 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 shorter 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 Wednesday of the following week. 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 the next Monday and will expire on the next Wednesday. So the full backup generated on Tuesday will not be automatically deleted until the next 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.
  - 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 an instance is created, you can set an automated backup policy. The system will back up data based on the automated backup policy.
- If **Automated Backup** is disabled, any automated backups in progress stop immediately.

Modifying an Automated Backup Policy

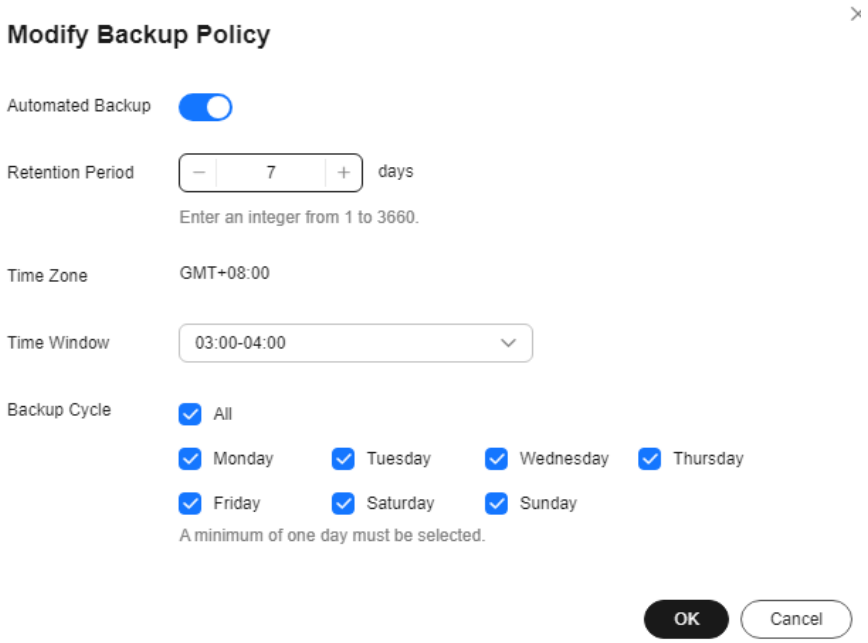
Step 1 Log in to the **GeminiDB console**.

Step 2 On the **Instances** page, click the instance whose backup policy you want to modify.

Step 3 Choose **Backups & Restorations** in the navigation pane one the left, and click **Modify Backup Policy**. In the displayed dialog box, set the backup policy. Click **OK**.

For details about how to set a backup policy, see **Configuring an Automated Backup Policy**.

Figure 3-82 Modifying a backup policy



Step 4 Check or manage the generated backups on the **Backups** or **Backups & Restorations** page.

----End

Disabling Automated Backup

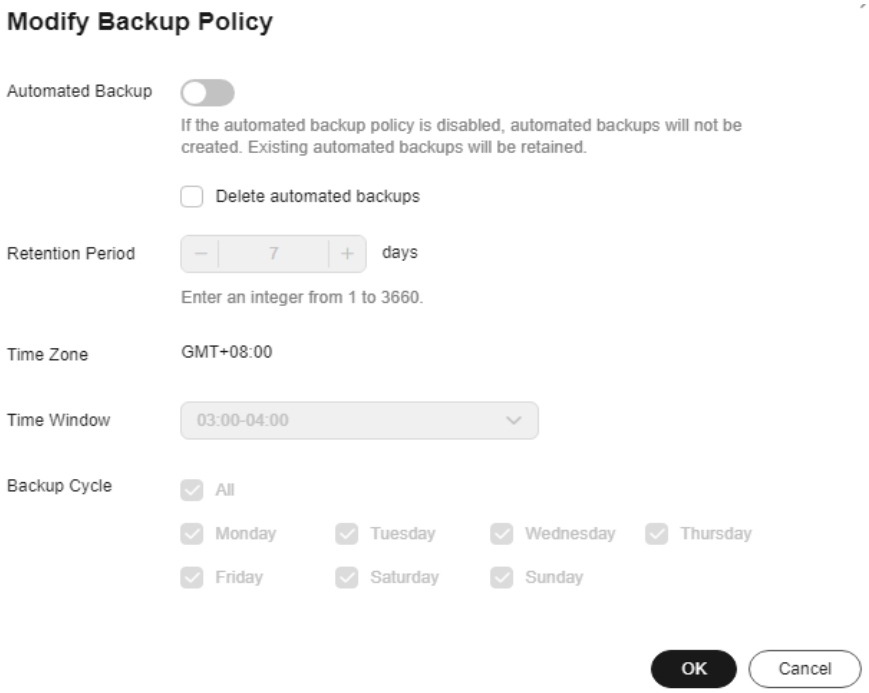
Step 1 Log in to the **GeminiDB console**.

Step 2 On the **Instances** page, click the instance whose backup policy you want to modify.

Step 3 Choose **Backups & Restorations** in the navigation pane one the left, and click **Modify Backup Policy**.

Step 4 In the displayed dialog box, click  to disable **Automated Backup** and click **OK**.

Figure 3-83 Disabling backup policies



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.

CAUTION

To delete an automated backup, disable the automated backup policy first. For details, see **Disabling Automated Backup**.

Deleted backups cannot be restored.

- **Method 1**
  a. Log in to the **GeminiDB console**.
  b. On the **Instances** page, click the instance whose backup you want to delete.
  c. Choose **Backups & Restorations** in the navigation pane, locate the target backup and click **Delete** in the **Operation** column.
  d. In the **Delete Backup** dialog box, confirm the backup details and click **Yes**.
- **Method 2**
  a. Log in to the **GeminiDB console**.
  b. On the **Backups** page, locate the backup that you want to delete and click **Delete**.
  c. In the **Delete Backup** dialog box, confirm the backup details and click **Yes**.

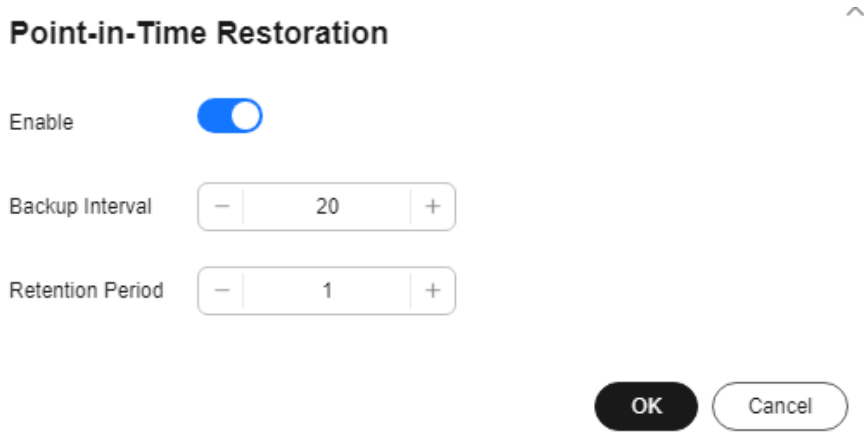
Setting the Policy for Restoring Data to a Specified Time Point

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

Step 2 On the **Instances** page, click the target instance.

Step 3 Choose **Backups & Restorations** in the navigation pane one the left, and click **Point in Time Restoration**. After the setting is complete, click **OK**.

Figure 3-84 Setting the policy for restoring data to a specified time point



- You can toggle on or off **Enable** to configure point-in-time restoration.
- **Backup Interval** determines an interval at which backups are automatically created. The value ranges from 5 to 120 minutes.
  For example, if it is set to 5 minutes and the first backup is created at 04:00, the next backup will be created 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

3.8.3 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. Manual backups are full backups.

Usage Notes

- Manual backups are charged for instances with cloud native storage during OBT.

Creating a Manual Backup

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

Step 2 Create a manual backup.

Method 1

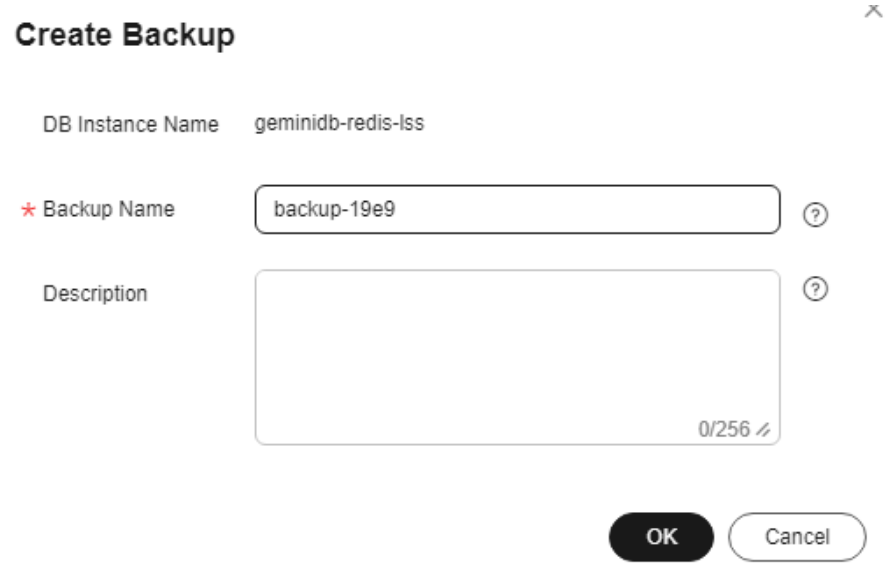
On the **Instances** page, locate the instance you want to back up and choose **More** > **Create Backup** in the **Operation** column.

Figure 3-85 Creating a manual backup

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**.

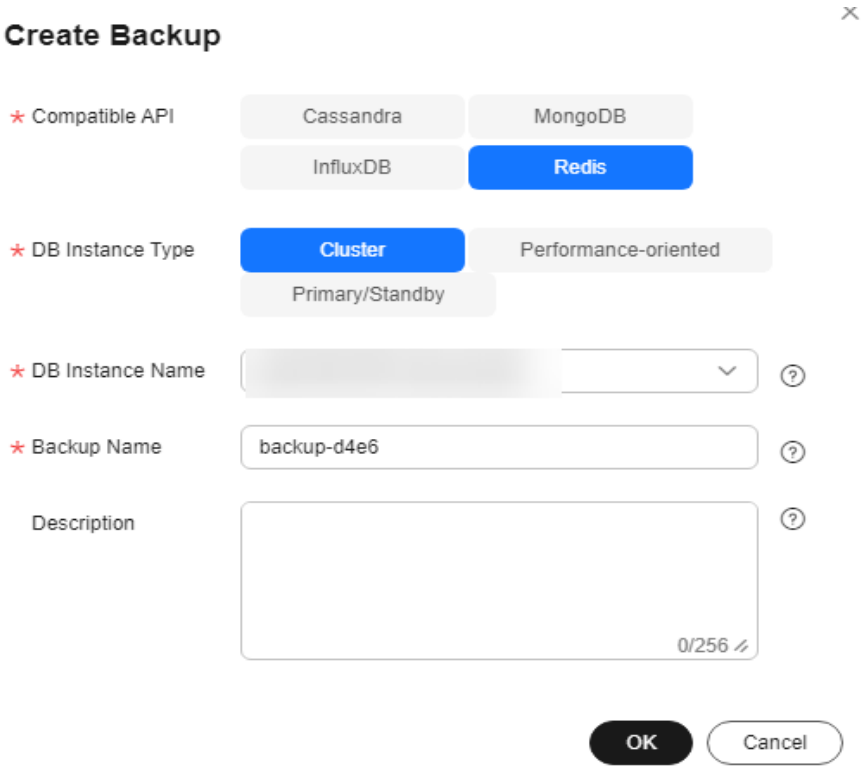
Figure 3-86 Creating a manual backup



Method 3

In the navigation pane on the left, choose **Backups**. On the displayed page, click **Create Backup**.

Figure 3-87 Creating a manual backup



Step 3 In the displayed dialog box, specify a backup name and description and click **OK**.

Table 3-36 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 4 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 no longer need a manual backup, delete it on the **Backups** or **Backups & Restorations** page.

Deleted backups are not displayed in the backup list.

CAUTION

Deleted backups cannot be restored.

Method 1

1. Log in to the **GeminiDB console**.
2. On the **Instances** page, locate the instance whose backup you want to delete and click its name.
3. Choose **Backups & Restorations** in the navigation pane, locate the target backup and click **Delete** in the **Operation** column.
4. In the displayed dialog box, confirm the backup details and click **Yes**.

Method 2

1. Log in to the **GeminiDB console**.
2. On the **Backups** page, locate the backup that you want to delete and click **Delete**.
3. In the displayed dialog box, confirm the backup details and click **Yes**.

3.9 Data Restoration

3.9.1 Restoration Methods

GeminiDB Redis supports multiple forms of data restoration. You can select one based on service requirements.

Table 3-37 Restoration methods

Reference	Scenario
Rebuilding an Instance	If an instance is deleted by mistake, you can rebuild it within a retention period in the recycle bin.
Restoring Data to a New Instance	You can restore an existing backup file to a new instance.

3.9.2 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 Restore a DB instance from the backup.

Method 1

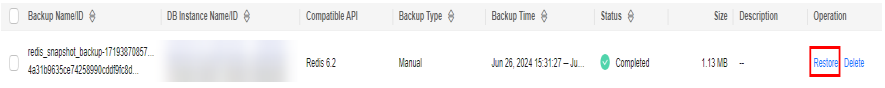
1. On the **Instances** page, locate the instance whose backup you want to restore and click its name.
2. Choose **Backups & Restorations** in the navigation pane on the left, locate the backup that you want to restore and click **Restore** in the **Operation** column.

Figure 3-88 Restoration

Method 2

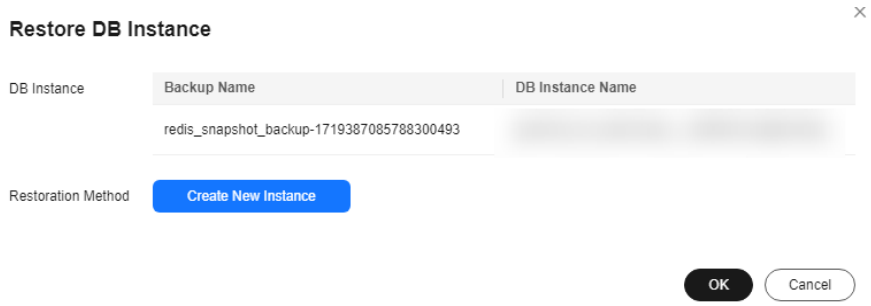
On the **Backups** page, locate the backup that you want to restore and click **Restore** in the **Operation** column.

Figure 3-89 Restoration



Step 3 In the displayed dialog box, confirm the current instance details and restoration method and click **OK**.

Figure 3-90 Restoring data to a new DB 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 4 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

3.10 CTS Audit

3.10.1 Key Operations Supported by CTS

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

Table 3-38 GeminiDB Redis key operations

Operation	Resource Type	Trace Name
Creating an instance	instance	NoSQLCreateInstance
Deleting an instance	instance	NoSQLDeleteInstance

Operation	Resource Type	Trace Name
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
Resetting a parameter template	parameterGroup	NoSQLResetConfigurations
Applying a parameter template	parameterGroup	NoSQLApplyConfigurations

Operation	Resource Type	Trace Name
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

3.10.2 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 **CTS console**.

Step 2 Choose **Trace List** in the navigation pane on the left.

Step 3 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 4 Click ⌄ on the left of the record to be queried to extend its details.

Step 5 Locate a trace and click **View Trace** in the **Operation** column.

----End

3.11 Viewing Metrics and Configuring Alarms

3.11.1 Supported Metrics

Description

This section describes GeminiDB Redis API metrics reported to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics and alarms.

Namespace

SYS.NoSQL

Monitoring Metrics

NOTE

You can view the instance-level and node-level metrics described in **Table 3-39** on each instance node by referring to **Viewing Metrics**. The instance-level metrics displayed on each instance node are the same.

Table 3-39 GeminiDB Redis API metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
nosql001_cpu_usage	CPU Usage	CPU usage of the monitored system	0–100%	GeminiDB Redis instance nodes	1 minute
nosql002_mem_usage	Memory Usage	Memory usage of the monitored system	0–100%	GeminiDB Redis instance nodes	1 minute
nosql005_disk_usage	Storage Space Usage	Disk usage of the monitored container	0–100%	GeminiDB Redis instances	1 minute
nosql006_disk_total_size	Total Disk Size	Total disk capacity of the monitored container	≥ 0 GB	GeminiDB Redis instances	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
nosql007_disk_used_size	Used Storage Space	Used disk space of the monitored container	≥ 0 GB	GeminiDB Redis instances	1 minute
redis017_proxy_accept	Total Clients Received by Proxy	Total number of clients received by the proxy	≥ 0 counts	GeminiDB Redis instance nodes	1 minute
redis018_proxy_request_ps	Request Acceptance Rate	Rate at which the proxy receives client requests	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis019_proxy_response_ps	Proxy Response Rate	Rate at which the proxy returns requests to the client	≥ 0 counts/s	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 /s	≥ 0 bytes/s	GeminiDB Redis instance nodes	1 minute
redis021_proxy_send_client_bps	Proxy Byte Stream Send Rate	Rate at which the proxy sends byte streams to the client /s	≥ 0 bytes/s	GeminiDB Redis instance nodes	1 minute
redis032_shard_qps	Shard QPS	QPS of the shard Unit: count	≥ 0 counts	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 µs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis037_exists_max_usec	Maximum Proxy Latency of exists Command	Maximum latency when the proxy executes the exists command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis039_exists_qps	Proxy exists Command Rate	Rate at which the proxy executes the exists command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis040_expire_avg_usec	Average Proxy Latency of expire Command	Average latency when the proxy executes the expire command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis043_expire_qps	Proxy expire Command Rate	Rate at which the proxy executes the expire command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis046_del_p99	Proxy P99 Latency of del Command	P99 latency when the proxy executes the del command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis047_del_qps	Proxy del Command Rate	Rate at which the proxy executes the del command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis050_ttl_p99	Proxy P99 Latency of ttl Command	P99 latency when the proxy executes the ttl command Unit: µs	≥ 0 µs	GeminiDB Redis instance nodes	1 minute
redis051_ttl_qps	Proxy ttl Command Rate	Rate at which the proxy executes the ttl command	≥ 0 counts/s	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 µs	GeminiDB Redis instance nodes	1 minute
redis053_persist_max_usec	Maximum Proxy Latency of persist Command	Maximum latency when the proxy executes the persist command Unit: µs	≥ 0 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis055_persist_qps	Proxy persist Command Rate	Rate at which the proxy executes the persist command	≥ 0 counts/s	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 µs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis057_scan_max_usec	Maximum Proxy Latency of scan Command	Maximum latency when the proxy executes the scan command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis059_scan_qps	Proxy scan Command Rate	Rate at which the proxy executes the scan command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis063_set_qps	Proxy set Command Rate	Rate at which the proxy executes the set command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis064_get_avg_usec	Average Proxy Latency of get Command	Average latency when the proxy executes the get command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis066_get_p99	Proxy P99 Latency of get Command	P99 latency when the proxy executes the get command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis067_get_qps	Proxy get Command Rate	Rate at which the proxy executes the get command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis071_getset_qps	Proxy getset Command Rate	Rate at which the proxy executes the getset command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis075_append_qps	Proxy append Command Rate	Rate at which the proxy executes the append command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	GeminiDB Redis instance nodes	1 minute
redis079_mget_qps	Proxy mget Command Rate	Rate at which the proxy executes the mget command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis083_mset_qps	Proxy mset Command Rate	Rate at which the proxy executes the mset command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis087_getrange_qps	Proxy getrange Command Rate	Rate at which the proxy executes the getrange command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis090_setrange_p99	Proxy P99 Latency of setrange Command	P99 latency when the proxy executes the setrange command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis091_setrange_qps	Proxy setrange Command Rate	Rate at which the proxy executes the setrange command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis095_substr_qps	Proxy substr Command Rate	Rate at which the proxy executes the substr command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis096_strlen_avg_usec	Average Proxy Latency of strlen Command	Average latency when the proxy executes the strlen command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis097_strlen_max_usec	Maximum Proxy Latency of strlen Command	Maximum latency when the proxy executes the strlen command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis099_strlen_qps	Proxy strlen Command Rate	Rate at which the proxy executes the strlen command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis102_incr_p99	Proxy P99 Latency of incr Command	P99 latency when the proxy executes the incr command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis103_incr_qps	Proxy incr Command Rate	Rate at which the proxy executes the incr command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis105_decr_max_usec	Maximum Proxy Latency of decr Command	Maximum latency when the proxy executes the decr command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis107_decr_qps	Proxy decr Command Rate	Rate at which the proxy executes the decr command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis109_hset_max_usec	Maximum Proxy Latency of hset Command	Maximum latency when the proxy executes the hset command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis111_hset_qps	Proxy hset Command Rate	Rate at which the proxy executes the hset command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis112_hget_avg_usec	Average Proxy Latency of hget Command	Average latency when the proxy executes the hget command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis115_hget_qps	Proxy hget Command Rate	Rate at which the proxy executes the hget command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis116_hmset_avg_usec	Average Proxy Latency of hmset Command	Average latency when the proxy executes the hmset command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis119_hmset_qps	Proxy hmset Command Rate	Rate at which the proxy executes the hmset command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis120_hmget_avg_usec	Average Proxy Latency of hmget Command	Average latency when the proxy executes the hmget command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis122_hmget_p99	Proxy P99 Latency of hmget Command	P99 latency when the proxy executes the hmget command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis123_hmget_qps	Proxy hmget Command Rate	Rate at which the proxy executes the hmget command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis127_hdel_qps	Proxy hdel Command Rate	Rate at which the proxy executes the hdel command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis131_hgetall_qps	Proxy hgetall Command Rate	Rate at which the proxy executes the hgetall command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 µs	GeminiDB Redis instance nodes	1 minute
redis135_hexists_qps	Proxy hexists Command Rate	Rate at which the proxy executes the hexists command	≥ 0 counts/s	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 µs	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 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis139_hincrby_qps	Proxy hincrby Command Rate	Rate at which the proxy executes the hincrby command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis143_hkeys_qps	Proxy hkeys Command Rate	Rate at which the proxy executes the hkeys command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis146_hlen_p99	Proxy P99 Latency of hlen Command	P99 latency when the proxy executes the hlen command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis147_hlen_qps	Proxy hlen Command Rate	Rate at which the proxy executes the hlen command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis148_hstrlen_avg_usec	Average Proxy Latency of hstrlen Command	Average latency when the proxy executes the hstrlen command Unit: µs	≥ 0 µs	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 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis151_hstrlen_qps	Proxy hstrlen Command Rate	Rate at which the proxy executes the hstrlen command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis152_hvals_avg_usec	Average Proxy Latency of hvals Command	Average latency when the proxy executes the hvals command Unit: µs	≥ 0 µs	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 µs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis154_hvals_p99	Proxy P99 Latency of hvals Command	P99 latency when the proxy executes the hvals command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis155_hvals_qps	Proxy hvals Command Rate	Rate at which the proxy executes the hvals command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis158_hscan_p99	Proxy P99 Latency of hscan Command	P99 latency when the proxy executes the hscan command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis159_hscan_qps	Proxy hscan Command Rate	Rate at which the proxy executes the hscan command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis160_lpush_avg_usec	Average Proxy Latency of lpush Command	Average latency when the proxy executes the lpush command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis163_lpush_qps	Proxy lpush Command Rate	Rate at which the proxy executes the lpush command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis164_lpop_avg_usec	Average Proxy Latency of lpop Command	Average latency when the proxy executes the lpop command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis166_lpop_p99	Proxy P99 Latency of lpop Command	P99 latency when the proxy executes the lpop command Unit: µs	≥ 0 µs	GeminiDB Redis instance nodes	1 minute
redis167_lpop_qps	Proxy lpop Command Rate	Rate at which the proxy executes the lpop command	≥ 0 counts/s	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 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis170_rpush_p99	Proxy P99 Latency of rpush Command	P99 latency when the proxy executes the rpush command Unit: µs	≥ 0 µs	GeminiDB Redis instance nodes	1 minute
redis171_rpush_qps	Proxy rpush Command Rate	Rate at which the proxy executes the rpush command	≥ 0 counts/s	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 µs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis173_rpop_max_usec	Maximum Proxy Latency of rpop Command	Maximum latency when the proxy executes the rpop command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis175_rpop_qps	Proxy rpop Command Rate	Rate at which the proxy executes the rpop command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis176_rpoplpush_avg_usec	Average Proxy Latency of rpoplpush Command	Average latency when the proxy executes the rpoplpush command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis179_rpoplpush_qps	Proxy rpoplpush Command Rate	Rate at which the proxy executes the rpoplpush command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis180_llen_avg_usec	Average Proxy Latency of llen Command	Average latency when the proxy executes the llen command Unit: µs	≥ 0 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis182_llen_p99	Proxy P99 Latency of llen Command	P99 latency when the proxy executes the llen command Unit: µs	≥ 0 µs	GeminiDB Redis instance nodes	1 minute
redis183_llen_qps	Proxy llen Command Rate	Rate at which the proxy executes the llen command	≥ 0 counts/s	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 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis187_lindex_qps	Proxy lindex Command Rate	Rate at which the proxy executes the lindex command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis191_linsert_qps	Proxy linsert Command Rate	Rate at which the proxy executes the linsert command	≥ 0 counts/s	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 µs	GeminiDB Redis instance nodes	1 minute
redis195_lrange_qps	Proxy lrange Command Rate	Rate at which the proxy executes the lrange command	≥ 0 counts/s	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 µs	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 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis199_lrem_qps	Proxy lrem Command Rate	Rate at which the proxy executes the lrem command	≥ 0 counts/s	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 µs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric 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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis203_lset_qps	Proxy lset Command Rate	Rate at which the proxy executes the lset command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis207_ltrim_qps	Proxy ltrim Command Rate	Rate at which the proxy executes the ltrim command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis208_sadd_avg_usec	Average Proxy Latency of sadd Command	Average latency when the proxy executes the sadd command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis211_sadd_qps	Proxy sadd Command Rate	Rate at which the proxy executes the sadd command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis213_spop_max_usec	Maximum Proxy Latency of spop Command	Maximum latency when the proxy executes the spop command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis214_spop_p99	Proxy P99 Latency of spop Command	P99 latency when the proxy executes the spop command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis215_spop_qps	Proxy spop Command Rate	Rate at which the proxy executes the spop command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis219_scard_qps	Proxy scard Command Rate	Rate at which the proxy executes the scard command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis220_smembers_avg_usec	Average Proxy Latency of smembers Command	Average latency when the proxy executes the smembers command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis223_smembers_qps	Proxy smembers Command Rate	Rate at which the proxy executes the smembers command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis225_srem_max_usec	Maximum Proxy Latency of srem Command	Maximum latency when the proxy executes the srem command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis226_srem_p99	Proxy P99 Latency of srem Command	P99 latency when the proxy executes the srem command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute
redis227_srem_qps	Proxy srem Command Rate	Rate at which the proxy executes the srem command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis231_sunion_qps	Proxy sunion Command Rate	Rate at which the proxy executes the sunion command	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis232_sinter_avg_usec	Average Proxy Latency of sinter Command	Average latency when the proxy executes the sinter command Unit: μs	≥ 0 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis235_sinter_qps	Proxy sinter Command Rate	Rate at which the proxy executes the sinter command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis237_sismember_max_usec	Maximum Proxy Latency of sismember Command	Maximum latency when the proxy executes the sismember command Unit: μs	≥ 0 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis238_sismember_p99	Proxy P99 Latency of sismember Command	P99 latency when the proxy executes the sismember command Unit: µs	≥ 0 µs	GeminiDB Redis instance nodes	1 minute
redis239_sismember_qps	Proxy sismember Command Rate	Rate at which the proxy executes the sismember command	≥ 0 counts/s	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 µs	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 µs	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 µs	GeminiDB Redis instance nodes	1 minute
redis243_sdiff_qps	Proxy sdiff Command Rate	Rate at which the proxy executes the sdiff command	≥ 0 counts/s	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 µs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis245_srandmember_max_usec	Maximum Proxy Latency of srandmember Command	Maximum latency when the proxy executes the srandmember command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis247_srandmember_qps	Proxy srandmember Command Rate	Rate at which the proxy executes the srandmember command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis249_zadd_max_usec	Maximum Proxy Latency of zadd Command	Maximum latency when the proxy executes the zadd command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis251_zadd_qps	Proxy zadd Command Rate	Rate at which the proxy executes the zadd command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis255_zcard_qps	Proxy zcard Command Rate	Rate at which the proxy executes the zcard command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis257_zscan_max_usec	Maximum Proxy Latency of zscan Command	Maximum latency when the proxy executes the zscan command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis259_zscan_qps	Proxy zscan Command Rate	Rate at which the proxy executes the zscan command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis261_zincrby_max_usec	Maximum Proxy Latency of zincrby Command	Maximum latency when the proxy executes the zincrby command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis263_zincrby_qps	Proxy zincrby Command Rate	Rate at which the proxy executes the zincrby command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis267_zrevrange_qps	Proxy zrevrange Command Rate	Rate at which the proxy executes the zrevrange command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis269_zrange_max_usec	Maximum Proxy Latency of zrange Command	Maximum latency when the proxy executes the zrange command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis271_zrange_qps	Proxy zrange Command Rate	Rate at which the proxy executes the zrange command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis273_zcount_max_usec	Maximum Proxy Latency of zcount Command	Maximum latency when the proxy executes the zcount command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis275_zcount_qps	Proxy zcount Command Rate	Rate at which the proxy executes the zcount command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis279_zrem_qps	Proxy zrem Command Rate	Rate at which the proxy executes the zrem command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis281_zscore_max_usec	Maximum Proxy Latency of zscore Command	Maximum latency when the proxy executes the zscore command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis283_zscore_qps	Proxy zscore Command Rate	Rate at which the proxy executes the zscore command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis285_zrank_max_usec	Maximum Proxy Latency of zrank Command	Maximum latency when the proxy executes the zrank command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis287_zrank_qps	Proxy zrank Command Rate	Rate at which the proxy executes the zrank command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis291_zrevrank_qps	Proxy zrevrank Command Rate	Rate at which the proxy executes the zrevrank command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis293_zlexcount_max_usec	Maximum Proxy Latency of zlexcount Command	Maximum latency when the proxy executes the zlexcount command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis295_zlexcount_qps	Proxy zlexcount Command Rate	Rate at which the proxy executes the zlexcount command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis297_zpopmax_max_usec	Maximum Proxy Latency of zpopmax Command	Maximum latency when the proxy executes the zpopmax command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis299_zpopmax_qps	Proxy zpopmax Command Rate	Rate at which the proxy executes the zpopmax command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis303_zpopmin_qps	Proxy zpopmin Command Rate	Rate at which the proxy executes the zpopmin command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis305_zremrangebyrank_max_usec	Maximum Proxy Latency of zremrangebyrank Command	Maximum latency when the proxy executes the zremrangebyrank command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis307_zremrangebyrank_qps	Proxy zremrangebyrank Command Rate	Rate at which the proxy executes the zremrangebyrank command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute
redis309_zremrangebyscore_max_usec	Maximum Proxy Latency of zremrangebyscore Command	Maximum latency when the proxy executes the zremrangebyscore command Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis311_zremrangebyscore_qps	Proxy zremrangebyscore Command Rate	Rate at which the proxy executes the zremrangebyscore command	≥ 0 counts/s	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 μs	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 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis315_zremrangebylex_qps	Proxy zremrangebylex Command Rate	Rate at which the proxy executes the zremrangebylex command	≥ 0 counts/s	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis317_all_max_usec	Maximum Proxy Latency of Commands	Maximum latency when the proxy executes commands Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute
redis319_all_qps	Proxy Command Rate	Rate at which the proxy executes commands	≥ 0 counts/s	GeminiDB Redis instance nodes	1 minute
redis661_rsync_ops	rsync Rate	Rate that rsync transfers data in a collection period	≥ 0 counts/s	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
redis663_rsync_push_cost	Average Push Time	Average time required for rsync to push data in a collection period Unit: μs	≥ 0 μs	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 μs	GeminiDB Redis instance nodes	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
redis665_rsync_max_push_cost	Maximum Push Time	Maximum time required for a push operation in a collection period Unit: μs	≥ 0 μs	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 μs	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

3.11.2 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.

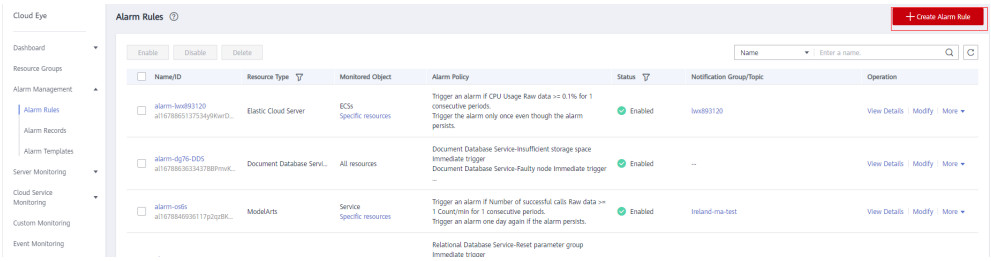
Procedure

Step 1 Log in to the **Cloud Eye console**.

Step 2 In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.

Step 3 On the **Alarm Rules** page, click **Create Alarm Rule**.

Figure 3-91 Creating an alarm rule



Step 4 Set alarm parameters.

1. Configure basic alarm information.

Figure 3-92 Configuring basic information for an alarm rule

Table 3-40 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
Description	(Optional) Alarm rule description.	-

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

Table 3-41 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**.	-

Parameter	Description	Example Value
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 3-93 Configuring the alarm policy

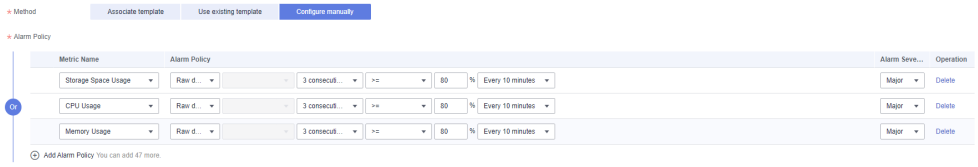


Table 3-42 Parameter description

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**.	-

Parameter	Description	Example Value
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 **Manually 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 **increase node specifications** in a timely manner. For more metrics, see **Supported 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 3-93** 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 3-94 Configuring alarm notification information

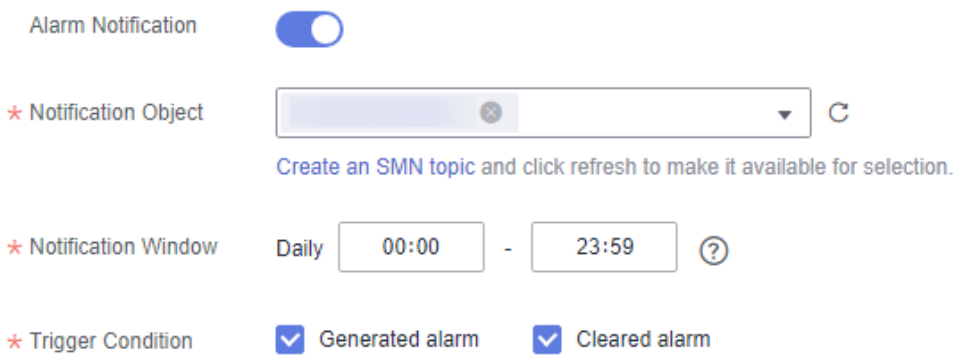


Table 3-43 Parameter description

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.	-
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 3-95 Advanced settings

Table 3-44 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 5 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.

NOTE

For more information about alarm rules, see **Cloud Eye User Guide**.

----End

3.11.3 Viewing Metrics

Scenarios

Cloud Eye monitors the status of GeminiDB Redis instances. You can view metrics on the 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 On the **Instances** page, click the instance whose metrics you want to view and click its name.

- Instance metrics: In the upper right corner, click **View Metric**.
- Node metrics: In the navigation pane, choose **Node Management**. In the **Node Information** area, browse to the target node and click **View Metric** in the **Operation** column.

Step 3 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

Method 2

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

Step 2 On the **Instance** page, click the instance whose metrics you want to view and click its name.

Step 3 In the navigation pane, choose **Metrics**.

Step 4 On the **Metrics** page, view real-time monitoring data.

- Click the **DB Instance** tab to view real-time monitoring data, such as the instance QPS, average hit ratio, and connection usage.
- Click the **Node-level Metrics** tab to view real-time monitoring data, such as CPU, memory, and connection usage.
- The following monitoring time windows are supported: last 1 hour, last 3 hours, last 12 hours, last 24 hours, last 7 days, and a custom time period.
- You can also enable auto refresh (every 60s).
- The monitoring period can be 1 minute or 5 minutes.

----End

3.11.4 Configuring a Dashboard

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

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

Procedure

Step 1 Log in to the **Cloud Eye console**.

Step 2 Create a dashboard.

1. In the navigation pane, choose **My Dashboards** > **Custom Dashboards**. On the displayed page, click **Create Dashboard**.
2. In the displayed **Create Dashboard** dialog box, set parameters.

Figure 3-96 Configuring parameters

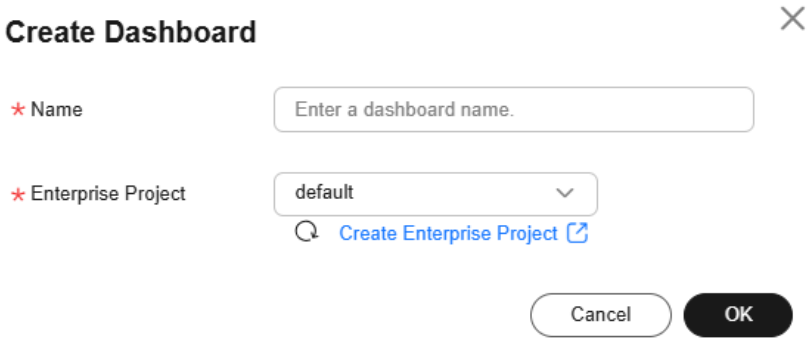


Table 3-45 Parameter description

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.

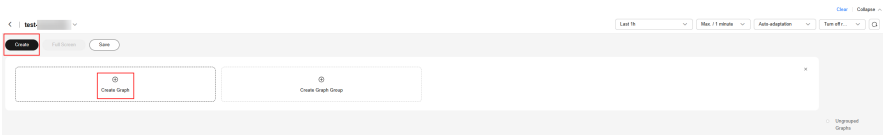
3. Click **OK**.

Step 3 Create a graph for the dashboard.

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

1. On the **My Dashboards** page, click the target dashboard name. On the displayed dashboard details page, click **Create** to create a graph or graph group.
   - Graph: The trend or instantaneous values of a metric are displayed in different charts.
   - Graph group: Graphs in a dashboard can be grouped into different groups, which are similar to file directories.

Figure 3-97 Creating a graph



2. Click **Create Graph**. On the displayed page, configure parameters by following .

   a. In the **Graph Settings** area, select **One graph for multiple metrics** or **One graph for a single metric**. Select an existing group from the **Graph Group** drop-down list or click **Create Graph Group** to create a graph group.

   b. You can select **Line chart** , **Stacked area line chart**, **Bar chart**, **Horizonal bar chart**, **Donut chart**, or **Table chart** for **Graph Type**.

   c. Earlier edition: In the **Monitoring Item Configuration** area in the lower left corner, set **Monitoring Scope**, **Compare With**, and **Quantity**.

      New console: In the **Select Metric** area, set the metric, monitoring scope (**All resources** or **Specified resources**), and whether to enable **Aggregation** and aggregation rules. Select **same period last week** or **same period yesterday** for **Compare With** and set the number of records displayed in a graph for the metric.

   d. In the upper right corner of **Select Metric** area, select **Left Y axis** or **Right Y axis**. View the configured chart in the **Preview** area.

      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.

3. On the selected dashboard, you can view the metric trend in the new graph.

----End

3.12 Tag Management

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 tags. TMS manages tags globally while 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.

Usage Notes

- 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 3-46**.
- A maximum of 20 tags can be added for each instance.
- The tag name must comply with the naming rules described in **Table 3-46**.

Table 3-46 Naming rules

Parameter	Requirement	Example Value
Tag key	• Cannot be left blank. • Must be unique for each instance. • Can contain a maximum of 128 characters. • Can only consist of digits, letters, underscores (_), and hyphens (-).	Organization
Tag value	• Can be left blank. • Can contain a maximum of 255 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 On the **Instances** page, click the instance that you want to add tags to and click its name.

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

Step 4 On the **Tags** page, click **Add Tag**. In the displayed dialog box, enter a tag key and value, and click **OK**.

Step 5 View and manage the tag on the **Tags** page.

----End

Editing a Tag

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

Step 2 On the **Instances** page, locate the instance whose tags you want to edit and click its name.

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

Step 4 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 5 View and manage the tag on the **Tags** page.

----End

Deleting a Tag

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

Step 2 On the **Instances** page, locate the instance whose tags you want to delete and click its name.

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

Step 4 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 5 View that the tag is no longer displayed on the **Tags** page.

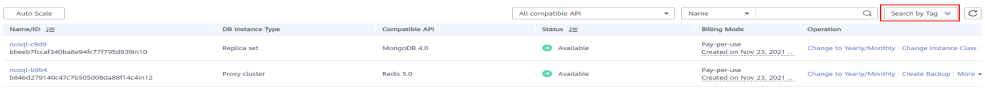
----End

Searching an Instance by Tag

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

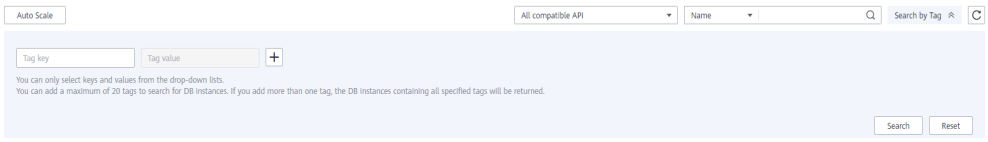
Step 2 On the **Instances** page, click **Search by Tag** in the upper right corner of the instance list.

Figure 3-98 Search by Tag



Step 3 Enter a tag key or value and click **Search** to query the instance associated with the tag.

Figure 3-99 Searching by tag key



----End

3.13 Memory Acceleration

3.13.1 RDS Memory Acceleration

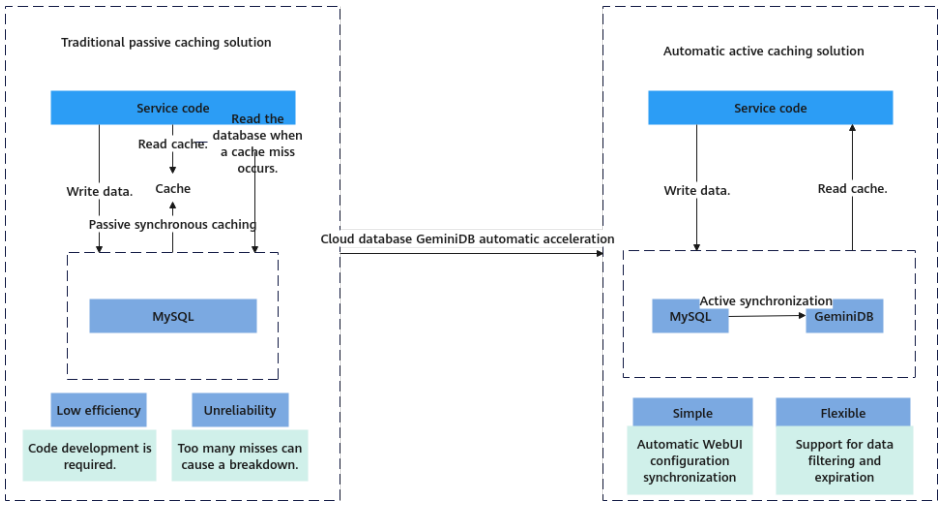
3.13.1.1 Memory Acceleration Overview

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 3-100 Diagram



3.13.1.2 Enabling and Using Memory Acceleration

This section describes how to enable memory acceleration. The process is as follows:

Selecting a GeminiDB Instance

Creating a Mapping Rule

Using the Memory Acceleration Module

Usage Notes

- After memory acceleration is enabled, commands such as RESET MASTER and FLUSH LOGS used to delete binlogs on MySQL instances are not allowed.
- Currently, only hash data from MySQL can be converted to GeminiDB Redis API.
- A Redis key prefix and a delimiter in a new rule can neither include those nor be included in those specified for an existing rule. For example, if the key prefix in a new rule is **pre1:** and is separated by a comma (,) and the key prefix in an existing rule is **pre1** and is separated by a colon (:), the new rule cannot be created.
- Currently, the ENUM, SET, and JSON data cannot be synchronized.
- Currently, only single-table queries are supported during lightweight incremental synchronization. Joint queries are not supported.
- Only GeminiDB Redis instances are charged. There are no other fees for this function.
- If you delete an RDS instance, the GeminiDB Redis instance with DB Cache enabled will not be deleted. If you do not need the GeminiDB Redis instance, delete it in a timely manner to avoid extra fees.

- When you purchase an RDS instance, if you select **Buy Now** for memory acceleration, a GeminiDB instance is automatically provisioned with DB Cache enabled. You can skip instance creation and start from **Creating a Mapping Rule**. This function is now in OBT. To use it, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

Selecting a GeminiDB Instance

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

Step 2 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 3 In the navigation pane, choose **Memory Acceleration**. On the **Memory Acceleration** page, click **Create GeminiDB Instance** or **Use Existing GeminiDB Instance**.

- Click **Create GeminiDB Instance** and perform **Step 4**.
- Click **Use Existing GeminiDB Instance** and select an existing GeminiDB Redis instance.

  NOTE

  When you select **Use Existing GeminiDB Instance**, only primary/standby instances are supported. The region, VPC, subnet, and security group of the GeminiDB and RDS instances must be the same.

Step 4 Set parameters listed in **Table 3-47** and click **Submit**.

Table 3-47 Basic information

Parameter	Description
Instance Specifications	CPU and memory of the instance. For details, see **Table 3-48**.
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 (_).

Parameter	Description
Database Password	Database password set by a user: • Can contain 8 to 32 characters. • Can contain at least two types of the following characters: 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.

NOTE

By default, the region, AZ, VPC, and subnet of the GeminiDB and RDS instances are the same.

Table 3-48 GeminiDB Redis instance specifications

Storage (GB)	Nodes	vCPUs	QPS	Maximum Connections per Single-node Instance	Databases
16	2	1	10,000	10,000	1,000
24	2	2	20,000	10,000	1,000
32	2	2	20,000	10,000	1,000
48	2	4	40,000	20,000	1,000
64	2	4	40,000	20,000	1,000
96	2	8	80,000	20,000	1,000
128	2	16	160,000	20,000	1,000

----End

Creating a Mapping Rule

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

Step 2 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 3 In the navigation pane, choose **Memory Acceleration**. In the **Mapping Rule** area, click **Create Mapping Rule**.

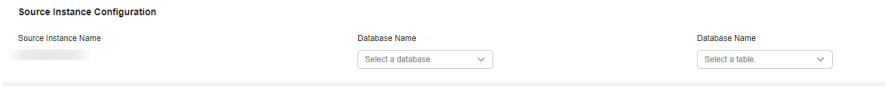
Step 4 On the displayed page, configure parameters.

1. Enter a rule name.

   Rule Name: Enter a mapping rule name. 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 a database of the acceleration instance.
   - **Table Name**: Select a table of the acceleration instance.

   Figure 3-101 Configuring source instance information

3. Configure acceleration instance information.
   - **Redis Key Prefix**: This parameter is optional. The default value is in the format of *Database name*:*Table name*:*Field name 1*:*Field name 2...* and can contain a maximum of 1,024 characters. If you have created a custom prefix, it will be used instead of the default one.
   - **Value Storage Type**: Data type of the cache. Currently, only hash data is supported.
   - **Database No. (0-999)**: ID of a database that stores cached data in the acceleration instance. The default value is **0**.
   - **TTL (s) Default value: 30 days**: Validity period 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 never expire.
   - **Key Delimiter**: Separator among the Redis key prefix, key, and key fields. It is a single character in length.

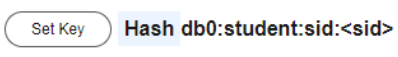
4. Click **Set Key**, select a key field of the acceleration instance, and click **OK**.

   NOTE

   If an acceleration instance key consists of multiple source instance fields, the key must be unique in a MySQL instance. You can click **Up** or **Down** to adjust the sequence of each field.

   After the parameters are set, the key is displayed.

   Figure 3-102 Key

5. Configure the acceleration instance fields.

   Move the required fields in the source instance to the acceleration instance.

6. After setting the parameters, click **Submit**.

----End

Using the Memory Acceleration Module

1. Create database **db1** in the source MySQL instance and create table **students** in **db1**.

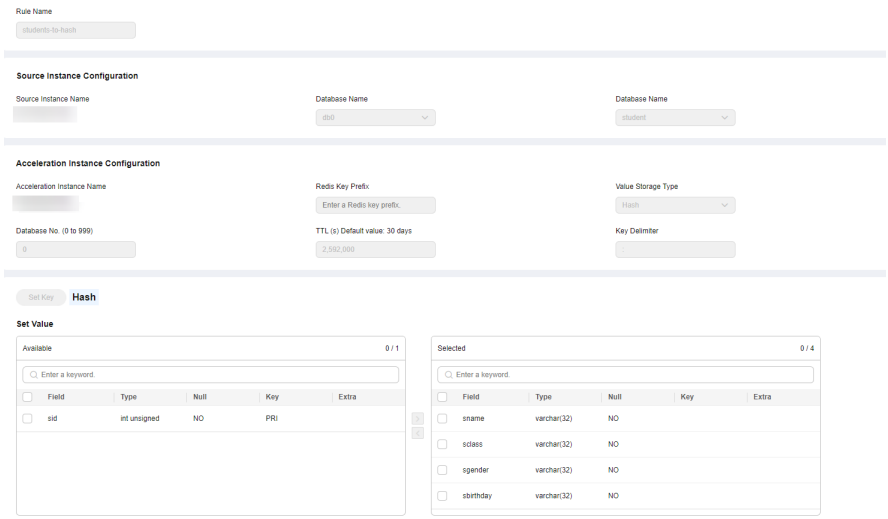
```
mysql> CREATE DATABASE db1;
Query OK, 1 row affected (0.00 sec)

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)
```

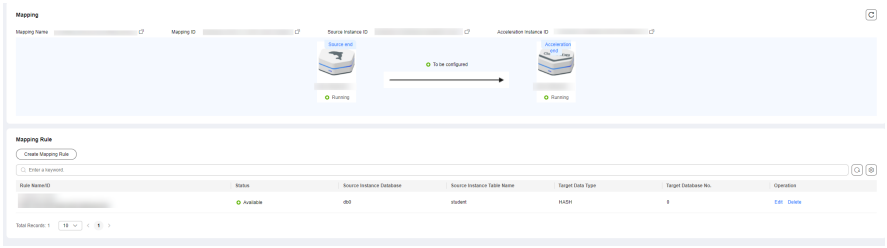
2. After the table is created, on the memory acceleration page, create a mapping rule to convert each row in the **students** table into a Redis hash. The key of a hash is in the format of *Database name*:*Data table name*:**sid:**<*sid value*>. The selected fields are **sname**, **sclass**, **sgender**, and **sbirthday**.

Figure 3-103 Configuring a mapping rule



3. After a mapping rule is created, check the mapping rule and information.

Figure 3-104 Mapping information



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. Check whether the new data is 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. Check whether the data is updated in 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: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. Check whether the data is deleted from the GeminiDB instance.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:2"
```

3.13.1.3 Modifying and Deleting a Memory Acceleration Rule

A memory acceleration rule can enable automated data synchronization from MySQL to GeminiDB. You can also modify and delete this rule.

Precautions

- Currently, only hashes from MySQL can be converted to GeminiDB Redis API.
- If a table name of the MySQL instance in the memory acceleration rule is changed, you need to reconfigure the rule.
- Currently, the ENUM, SET, and JSON data cannot be synchronized.
- If you rename or delete one or more key fields of a memory acceleration rule, the rule becomes invalid.

Modifying a Mapping Rule

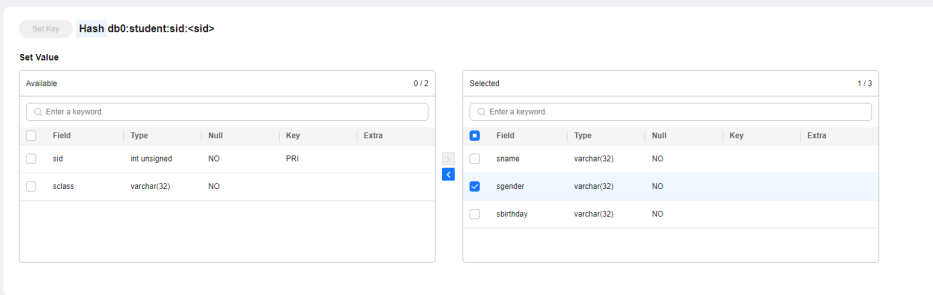
Step 1 Log in to the **RDS console**.

Step 2 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 3 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 4 After editing the fields, click **Submit**.

Figure 3-105 Editing a mapping rule



----End

Deleting a Mapping Rule

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

Step 2 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 3 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

3.13.1.4 Viewing and Removing Mappings

You can view the mapping list on the **Memory Acceleration Management** page and remove mappings.

Usage Notes

- After a mapping is removed, service applications cannot obtain the latest data of the source database from the acceleration instance.
- 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 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

Removing a Mapping

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

Step 2 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 3 In the displayed dialog box, click **OK**.

----End

4 FAQs

4.1 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 Application Code Needs to Be Refactored for Connecting to a Redis Client?
- Can Data Be Migrated from Open-Source Redis to GeminiDB Redis API? 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?
- How Do I Use Multiple Node IP Addresses Provided by GeminiDB Redis API?
- 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

- Why Is the Key Not Returned Using Scan Match?
- How Do I Process Existing Data Shards After Migrating Workloads to GeminiDB Redis API?
- How Long Does It Take to Add GeminiDB Redis Nodes at the Same Time? What Are the Impacts on Services?
- 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?
- Can I Download Backups of a GeminiDB Redis Instance to a Local PC and Restore Data Offline?
- What Is the Data Backup Mechanism of GeminiDB Redis API? What Are the Impacts on Services?
- Why Does the CPU Usage Remain High Despite Low Service Access Volume on a GeminiDB Redis Preferential Instance with 1 CPU and 2 Nodes?
- Why Does the Number of Keys Decrease and Then Become Normal on the Monitoring Panel on the GUI of GeminiDB Redis API?
- Why Is CPU Usage of GeminiDB Redis Instance Nodes Occasionally High?
- Which Commands Require Hashtags on GeminiDB Redis Cluster Instances?
- How Do I Resolve the Error "CROSSSLOT Keys in request don't hash to the same slot"?
- What Should I Do If "ERR unknown command sentinel" Is Displayed?

Backup and Restoration

How Long Can a GeminiDB Redis Instance Backup Be Saved?

4.2 About GeminiDB Redis API

4.2.1 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 comparison between GeminiDB Redis instances and open-source KV databases, see .

4.2.2 How Is the Performance of GeminiDB Redis API Compared with Open-Source Redis?

GeminiDB Redis API uses the multi-thread architecture. More CPUs can improve QPS (10,000–10,000,000).

Generally, the average latency of single-point access is less than 1 ms, and the p99 latency is less than 2 ms, similar to that of open-source Redis.

.

4.2.3 What Redis Versions and Commands Are Compatible with GeminiDB Redis API? Whether Application Code Needs to Be Refactored for Connecting to a Redis Client?

GeminiDB Redis API is fully compatible with Redis 6.2 (including 6.2.*x*) and earlier versions, such as 5.0, 4.0, and 2.8. It is partially compatible with Redis 7.0.

You can migrate data of Redis 6.2 and earlier instances (such as 5.0, 4.0, and 2.8) to GeminiDB Redis instances, without the need of code modifications. Any Redis client can be connected to GeminiDB Redis instances.

4.2.4 Can Data Be Migrated from Open-Source Redis to GeminiDB Redis API? What Are the Precautions?

Yes. Take care with the version and specifications before migration:

- 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. In the upper right corner of the console, choose **Service Tickets > Create Service Ticket** and contact the customer service.
- Specifications: Configure proper specifications based on QPS and data volumes of the source instance.

4.2.5 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.

4.2.6 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.

4.2.7 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**.

4.2.8 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. Synchronous writes greatly reduce write performance. To achieve higher data reliability, you need to enable synchronous writes. You can choose in the upper right corner of the console.

4.2.9 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.

4.2.10 Does GeminiDB Redis API Support Modules Such as a Bloom Filter?

A Bloom filter can be used to check whether an element is in a large-size data set. It is suitable for scenarios such as web interceptors and anti-cache penetration.

GeminiDB Redis API supports Bloom filters.

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.

4.3 Billing

4.3.1 What Are the Differences Between Yearly/Monthly and Pay-per-Use Billing Modes?

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 postpaid mode. You are only billed for how long you have actually used your instance. This mode can be a good option when future requirements are unpredictable. Pay-per-use instances are priced by the hour, but if an instance is used for less than one hour, you will be billed based on the actual duration.

4.3.2 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 a Yearly/Monthly Instance to Pay-per-Use**.
- For details about how to change the billing mode from pay-per-use to yearly/monthly, see **Changing a Pay-per-Use Instance to Yearly/Monthly**.

4.4 Database Usage

4.4.1 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.

```
139.9.177.148:6379> scan 1 match tes*
1) "21"
2) (empty list or set)
139.9.177.148:6379> get test
"abc"
139.9.177.148:6379>scan 0 match tes*
1) "21"
2) (empty list or set)
139.9.177.148:6379>
```

Possible Causes

The MATCH command is used to iterate elements that only match a specified pattern. Pattern matching is performed after the command obtains 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.

4.4.2 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.

4.4.3 Does GeminiDB Redis API Support Fuzzy Queries Using KEYS?

Yes.

Fuzzy queries using KEYS may cause OOM and longer latency.

KEYS can be used only in a test environment. In a production environment, use SCAN and MATCH instead.

4.4.4 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.

4.4.5 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

4.4.6 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.

4.4.7 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 hashtags to ensure that the keys are in the same slot. Avoid data skew when you use hashtags. For more information, see **Which Commands Require Hashtags on GeminiDB Redis Cluster Instances?**
- When hashtags cannot be used, change the instance type to primary/standby. For details, see **Compatible API and Versions**.

4.4.8 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).

4.4.9 Which Commands Require Hashtags on GeminiDB Redis Cluster Instances?

Hashtag Overview and Usage

Multi-key commands in a Redis Cluster must comply with the **hashtag mechanism**. Keys with the same hashtag must be allocated to the same hash slot to ensure the atomicity and performance of the multi-key commands. Otherwise, error "CROSSSLOT Keys in request don't hash to the same slot" will be reported. Rules for using a hashtag are as follows:

1. Basic format

If a key contains a "{}" pattern, only the substring between the braces is hashed to obtain the hash slot.

For example, the hashtags of {user:1000}.profile and {user:1000}.settings are both user:1000. Therefore, they are allocated to the same hash slot.

2. Location

{} can appear anywhere in a key.

For example, the hashtag of foo{user:1000}bar is still user:1000.

Only the first {} is valid.

If a key contains multiple braces, only the substring between the first braces is used as a hashtag.

For example, the hashtag of {user:1000}.{profile} is user:1000.

3. Scenarios

- Transactions: In a Redis Cluster, all operations within a MULTI/EXEC block must be performed on keys on the same node. A hashtag ensures that these keys are allocated to the same hash slot.
- Lua scripts: A hashtag ensures all keys used by Lua scripts are in the same slot.
- Multi-key operations: string (MSET and MGET), LIST (BLPOP, BRPOP, BRPOPLPUSH, and RPOPLPUSH), SET (SDIFF, SDIFFSTORE, SINTER, SINTERSTORE, SINTERCARD, SUNION, and SUNIONSTORE), and ZSET (ZINTER, ZINTERSTORE, ZINTERCARD, ZUNION, ZUNIONSTORE, ZDIFF, ZDIFFSTORE, and ZRANGESTORE), key management (DEL, EXISTS, UNLINK, TOUCH, RENAME, RENAMENX, and SORT), STREAM (XREAD and XREADGROUP), and BITOP

Example of using a cluster:

1. String: MSET/MGET

- Setting multiple keys (user data)

```
mset {user:1000}:name "Alice" {user:1000}:email "alice@example.com" {user:1000}:age 30
```

- Obtaining multiple keys

```
mget {user:1000}:name {user:1000}:email {user:1000}:age
```

2. Transaction: MULTI/EXEC

- Starting a transaction

```
MULTI
SET {order:1234}:status "processing"
EXPIRE {order:1234}:status 3600
EXEC
```

3. LUA script:

- Reducing inventory scripts and recording logs

```
EVAL "redis.call('DECR', KEYS[1]); redis.call('SET', KEYS[2], 'updated')" 2 {product:100}:stock
{product:100}:log
```

Splitting Commands Supported by Proxy Cluster GeminiDB Redis Instances

A proxy cluster can route commands, balance loads, and perform failovers. It can simplify the client-side logic while providing advanced features such as handling connections to multiple databases. You do not need to bother with shard management. The proxy cluster is recommended because it is compatible with a standalone Redis node, Redis Sentinel, and Redis Cluster.

The proxy cluster can simplify the logic of some multi-key commands by splitting and routing them to different backend nodes. After being executed, the commands are aggregated on the proxy and then returned to the client. Proxy cluster GeminiDB Redis instances support the following splitting commands:

- Key management: DEL, EXISTS, UNLINK, and TOUCH
- String: MGET and MSET
- SET: SDIFF, SDIFFSTORE, SINTER, SINTERSTORE, SINTERCARD, SUNION, and SUNIONSTORE
- ZSET: ZINTER, ZINTERSTORE, ZINTERCARD, ZUNION, ZUNIONSTORE, ZDIFF, ZDIFFSTORE, and ZRANGESTORE

- Multiple commands in a transaction can be split. If a transaction contains multi-key commands that cannot be split, hashtags must be added to keys involved in these commands.

Other commands cannot be split. You are advised to use a hashtag in the cluster to ensure the atomicity and performance of multi-key commands. Key management and string multi-key commands are more efficient than SET and ZSET. After being executed on shards, the commands are aggregated on the proxy. The outputs are returned to the client. To execute SET and ZSET, each key needs to be read to the proxy before related logic operations are performed. SET and ZSET are not recommended for big keys which lead to slower access and increased memory usage.

4.4.10 What Should I Do If "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 cluster GeminiDB Redis instances is not **3**, **SENTINEL** commands are not allowed.

Solution

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

Step 2 On the **Instances** page, click the target instance. The **Basic Information** page is displayed.

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

Step 4 Change the value of **CompatibleMode** and click **Save**.

- For a cluster instance, set **CompatibleMode** to **3**.

----End

4.4.11 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.

NOTE

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.

4.4.12 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, cluster instances contain three internal management nodes, which are changed at the same time. For example, three worker nodes and three internal management nodes are created for a GeminiDB Redis instance. The online change takes about 30 to 60 minutes. While specifications are changed, the node is disconnected, and its slots become disabled and are taken over by a functional node. In addition to node disconnection, there are also other interruptions in several seconds, for example, access timeout and invisible data partitions, so a reconnection mechanism must be established. 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 vCPUs and Memory**.

4.4.13 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**.

4.4.14 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**.

4.4.15 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.

For details about instance specifications, see **Instance Specifications**.

4.4.16 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.

4.4.17 Why Is CPU Usage of GeminiDB Redis Instance Nodes Occasionally High?

There are many possible reasons, such as sudden spike in service traffic, big 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 choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact the customer service.

4.4.18 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 4-1 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.

4.5 Database Connection

4.5.1 How Do I Connect to a GeminiDB Redis Instance?

You can connect to a GeminiDB Redis instance using a private network, public network, load balancer IP address, DAS, or program code. For details, see **Connecting to a GeminiDB Redis Instance**.

- You can connect to a GeminiDB Redis instance using a **web-based console client**.
- You can connect to a GeminiDB Redis instance through a **private IP address**, **private domain name**, or **load balancer address**.
- You can connect to a GeminiDB Redis instance through a **public domain name** or an **EIP**.

4.5.2 How Do I Use Multiple Node IP Addresses Provided by GeminiDB Redis API?

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. Choose **Service Tickets > Create Service Ticket** in the upper right corner of the console. Contact the customer service to configure Elastic Load Balance (ELB) 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)**.

4.5.3 How Does Load Balancing Work in GeminiDB Redis API?

GeminiDB Redis API uses dedicated load balancers with scalable specifications and supports a maximum bandwidth of 10 Gbit/s. For details, see .

4.5.4 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*.

4.5.5 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**.

4.5.6 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 a GeminiDB Redis Instance Over a Private Network**.

4.5.7 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 **Setting Security Group Rules for a GeminiDB Redis Instance**.

- 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 **Configuring Private Network Access to a GeminiDB Redis Instance**.

4.6 Backup and Restoration

4.6.1 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**.

4.7 Memory Acceleration

4.7.1 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.

4.7.2 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.

4.7.3 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.

4.7.4 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.

4.7.5 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.

4.7.6 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**.

4.8 Freezing, Releasing, Deleting, and Unsubscribing from Instances

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. Expired yearly/monthly instances cannot 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.
  - If they are yearly/monthly resources, no changes can be made to them.
  - 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 a GeminiDB Redis 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 a Yearly/Monthly Instance**.

