GaussDB(for MySQL)

User Guide

Issue 01

Date 2024-11-06

Contents

1 Permissions Management

1.1 Creating a User and Granting GaussDB(for MySQL) Permissions

This section describes how to use **IAM** for fine-grained permissions control over your GaussDB(for MySQL) 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 GaussDB(for MySQL) resources.

- Grant only the permissions required for users to perform specific tasks.

- Entrust a cloud service account to perform efficient O&M on your GaussDB(for MySQL) resources.

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

Figure 1-1 describes the procedure for granting permissions.

Prerequisites

Learn about the permissions (see **system-defined policies**) supported by GaussDB(for MySQL) and choose policies or roles according to your requirements. For the permissions of other services, see **System Permissions**.

Process Flow

Figure 1-1 Process for granting GaussDB(for MySQL) permissions
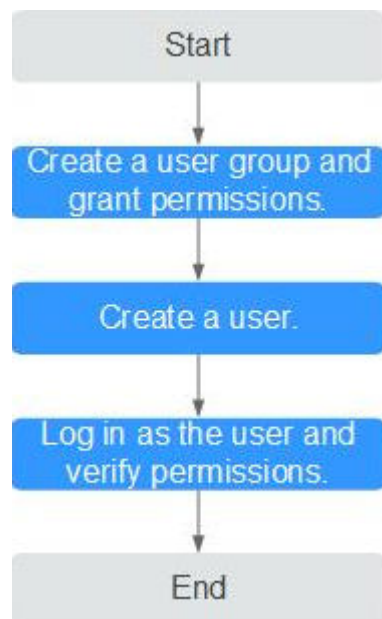
1. **Create a user group and assign permissions to it**.

   Create a user group on the IAM console, and attach the **GaussDB(for MySQL) GaussDB FullAccess** policy to the group.

   ☐ NOTE

   > To use some functions of other services, you need to configure the GaussDB(for MySQL) FullAccess permission and the permission of the corresponding 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**.

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

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

   Log in to the GaussDB(for MySQL) console using the created user, and verify that the user only has read permissions for GaussDB(for MySQL).

   Choose **Service List** > GaussDB(for MySQL) and click **Buy DB Instance**. If you can buy an instance, the required permission policy has already been applied.

1.2 Creating a GaussDB(for MySQL) Custom Policy

Custom policies can be created to supplement the system-defined policies of GaussDB(for MySQL).

You can create a custom policy 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: Write policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. This section provides examples of common GaussDB(for MySQL) custom policies.

Example Custom Policies

- Example 1: Allowing users to create GaussDB(for MySQL) instances

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

- Example 2: Denying GaussDB(for MySQL) instance deletion

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

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

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

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

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

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

```
    ]
}
```

- Example 4: Allowing users to manage specified instances and some functions of instances

  Assume that your account has multiple instances and you are a database administrator. If you want to allow users to manage specified instances and some functions of instances, you can create the following permission policy.

  ```
  {
      "Version": "1.1",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "gaussdb:instance:restart",
                  "gaussdb:instance:modify"
              ],
              "Resource": [
                  "GAUSSDB:*:*:instance:test*"
              ]
          },
          {
              "Effect": "Allow",
              "Action": [
                  "gaussdb:param:list",
                  "gaussdb:tag:list",
                  "gaussdb:backup:list",
                  "gaussdb:instance:create",
                  "gaussdb:instance:list"
              ]
          }
      ]
  }
  ```

  ◻ NOTE

  - Users granted these permissions can view all instances, but can manage only authorized instances. In addition, the database administrator can still use APIs to directly manage these instances. Users granted the permissions can only reboot and modify all instances under this account.

  - **test*** is an example of an instance name for fuzzy match and is mandatory in the permission policy. Otherwise, the authorized users cannot view any instance on the console.

  - GaussDB(for MySQL) supports API-level access control. You can use related APIs to perform fine-grained access control on GaussDB(for MySQL). For details, see **API Overview**.

2 Buying a DB Instance

2.1 Buying a Pay-per-Use DB Instance

Scenarios

This section describes how to create a pay-per-use DB instance on the GaussDB(for MySQL) console.

Billing

After you purchase a pay-per-use DB instance, you will be billed for resources you actually use. For more information, see **Pay-per-Use Billing**.

Procedure

Step 1 Go to the **Buy DB Instance** page.

Step 2 On the displayed page, configure required information and click **Next**.

Figure 2-1 Basic information

Table 2-1 Basic information

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	Region where the DB instance is deployed **NOTICE** ● Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For lower network latency and quicker resource access, select the nearest region. ● You cannot change the region of a DB instance once it is purchased.
DB Instance Name	The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. ● If you create multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on. ● The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0
DB Instance Type	Select **Primary/Standby** or **Single**. ● **Primary/Standby**: A primary/standby instance contains one primary node and up to 15 read replicas. The primary node processes read and write requests, and the read replicas process only read requests. If the primary node becomes unavailable, GaussDB(for MySQL) automatically fails over to a read replica. Primary/standby instances apply to medium- and large-sized enterprises in the Internet, taxation, banking, and insurance sectors. ● **Single**: A single-node instance contains only one primary node and there are no read replicas. Single-node instances do not involve data synchronization of nodes and can easily ensure atomicity, consistency, isolation, and durability of transactions. They are only recommended for development and testing of microsites, and small and medium enterprises, or for learning about GaussDB(for MySQL).

Parameter	Description
Storage Type	**DL6** The original **Shared storage**. The default storage type of GaussDB(for MySQL) instances created before July 2024 is **Shared storage**, while that of GaussDB(for MySQL) instances created after July 2024 is **DL6**. DL6-based instances achieve zero RPO with a 3 AZ deployment and deliver better performance and higher peak throughput. They are suitable for core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming. **DL5** A new storage type. It uses Huawei Cloud's hardware and network infrastructure technologies, ensuring that DL5-based instances maintain the same high availability as DL6-based instances with 3 AZs and zero RPO. Compared with DL6-based instances, although the peak performance of DL5-based instances may decrease, the cost per unit capacity is significantly reduced. DL5-based instances are suitable for CPU-intensive sub-core business systems or cost-effective application modules. For more information about storage types, see **Storage Types**.
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. **Single-AZ**: The primary node and read replicas are deployed in the same AZ. **Multi-AZ**: If your workloads require cross-AZ DR or are insensitive to cross-AZ latency, you are advised to deploy the primary node and read replicas in different AZs to achieve higher availability and reliability.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.

Figure 2-2 Specifications and storage of a pay-per-use DB instance

Table 2-2 Specifications and storage

Parameter	Description
Instance Specifications	GaussDB(for MySQL) is a cloud-native database that uses the shared storage. To ensure that DB instances run stably under high read/write pressure, GaussDB(for MySQL) controls the read/write peaks of DB instances based on instance specifications. For details about how to select specifications, see **Performance White Paper**. For more information about specifications, see **Instance Specifications**. After a DB instance is created, you can change its vCPUs and memory. For details, see **Changing the vCPUs or Memory of a DB Instance or Node**.
CPU Architecture	Select **x86** or **Kunpeng**. ● **x86**: x86 instances use Intel® Xeon® Scalable processors and feature robust and stable computing performance. When working on high-performance networks, the instances provide the additional performance and stability that enterprise-class applications demand. ● **Kunpeng**: Kunpeng instances use Kunpeng 920 processors and 25GE high-speed intelligent NICs for powerful compute and high-performance networks, making them an excellent choice for enterprises needing cost-effective, secure, and reliable cloud services.
Nodes	This parameter is mandatory for primary/standby DB instances. ● By default, each DB instance contains a primary node and multiple read replicas. ● You can create up to 9 read replicas for a pay-per-use DB instance at a time. ● You can also add read replicas after a DB instance is created. For details, see **Adding Read Replicas to a DB Instance**.

Parameter	Description
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage of a pay-per-use DB instance will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
TDE	Transparent Data Encryption (TDE) encrypts data files and backup files using certificates to implement real-time I/O encryption and decryption. This function effectively protects the security of databases and data files. After TDE is enabled, you need to select a cryptographic algorithm **AES256** or **SM4** as needed. **NOTE** ● To use TDE, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. ● TDE has certain constraints. For details, see **Enabling TDE for a DB Instance**.
Backup Space	GaussDB(for MySQL) provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis.

Figure 2-3 Network settings
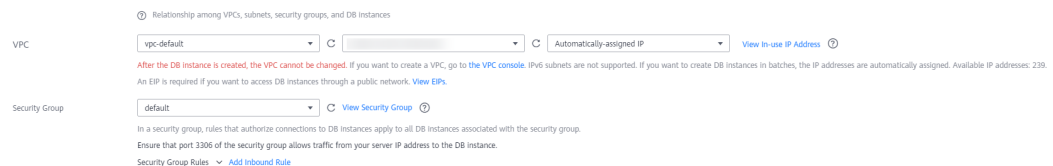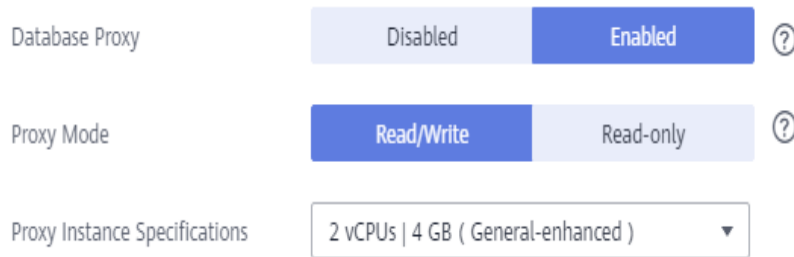
Table 2-3 Network

Parameter	Description
VPC	• A dedicated virtual network where your instances are located. It isolates networks for different workloads to enhance security. GaussDB(for MySQL) allocates a default VPC (default_vpc) for your instance. You can also use an existing, new, or shared VPC and subnet. – To use an existing VPC and subnet, select an existing VPC and subnet under the current account from the drop-down lists. – To use a new VPC and subnet, create a VPC and subnet, and then select the VPC and subnet from the drop-down lists. For details about how to create a VPC and subnet, see "Creating a VPC and Subnet" in ***Virtual Private Cloud User Guide***. – To use a shared VPC and subnet, select a VPC and subnet that another account shares with the current account from the drop-down lists. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts so you can centrally manage resources in multiple accounts. This helps improve resource management efficiency and reduces O&M costs. For more information about VPC subnet sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*. **NOTICE** After a DB instance is created, the VPC cannot be changed.
Security Group	Enhances security by controlling access to GaussDB(for MySQL) from other services. When you select a security group, you must ensure that it allows the client to access instances. If no security group is available or has been created, GaussDB(for MySQL) allocates a security group to you by default. **NOTE** To ensure subsequent database connections and access, click **Add Inbound Rule** to allow all IP addresses to access your DB instance through port 3306 and over ICMP. For details, see **Configuring Security Group Rules**.

Figure 2-4 Proxy instance settings

Table 2-4 Database proxy

Parameter	Description
Database Proxy	It is enabled by default. After the database proxy is enabled, you can use the IP address of a proxy instance to connect to your DB instance. **NOTE** ● To use this function, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. ● You can also create proxy instances after a DB instance is created. For details, see **Step 1: Create a Proxy Instance**.
Proxy Mode	You can select **Read/Write** or **Read-only** as needed. ● **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights. ● **Read-only**: The primary node does not process write and read requests, and all read requests are forwarded to the selected read replicas based on read weights.
Proxy Instance Specifications	You can select the proxy instance specifications as needed.

Figure 2-5 Database settings

Table 2-5 Database settings

Parameter	Description
Administrator	The default login name for the database is **root**.

Parameter	Description
Administrator Password	Must consist of 8 to 32 characters and contain at least three of the following: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.). Enter a strong password and periodically change it to improve security and defend against threats such as brute force cracking attempts. **NOTICE** If you select a custom parameter template during instance creation, the administrator password must comply with the values of validate_password parameters in the custom parameter template. Otherwise, the instance creation will fail. To check the parameter values, go to the **Parameter Templates** page, find the target parameter template and click its name. In the upper right corner of the page, search for **validate_password**. **Figure 2-6** Checking the password-related parameters Keep this password secure. If lost, the system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see **Resetting the Administrator Password**.
Confirm Password	Must be the same as **Administrator Password**.

Figure 2-7 Other information settings

Table 2-6 Other information

Parameter	Description
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can view the default parameter template (**Default-GaussDB-for-MySQL 8.0**), high-performance parameter template (**Default-GaussDB-for-MySQL 8.0-High Performance)**, and all custom parameter templates in the current region. You can select an appropriate parameter template as required. **NOTICE** ● If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not applied. Instead, the default values are used. "innodb_buffer_pool_size" "innodb_log_buffer_size" "max_connections" "innodb_buffer_pool_instances" "innodb_page_cleaners" "innodb_parallel_read_threads" "innodb_read_io_threads" "innodb_write_io_threads" "threadpool_size" ● The value of **innodb_parallel_select_count** is determined by your instance specifications, instead of the parameter value you configured in the parameter template. The default value is **OFF** for instance with 16 vCPUs or less and **ON** for instances with more than 16 vCPUs. For more information about parameter templates, see **Parameter Template Management**. For more information about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. You can modify the instance parameters as required after a DB instance is created. For details, see **Modifying Parameters of a DB Instance**.
Table Name	Specifies whether table names are case sensitive. This option cannot be changed later. ● **Case sensitive**: Table names are case sensitive. ● **Case insensitive**: Table names are case insensitive and are stored in lowercase letters by default.

Parameter	Description
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.

Figure 2-8 Tag settings
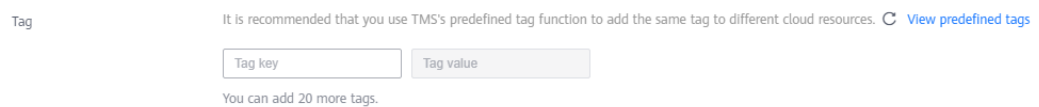
Table 2-7 Tags

Parameter	Description
Tag	(Optional) Tags a GaussDB(for MySQL) instance. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

Figure 2-9 Purchase quantity
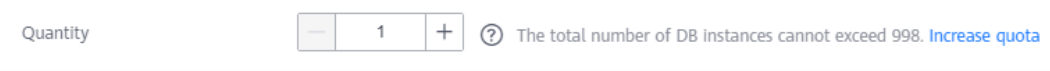
Table 2-8 Purchase quantity

Parameter	Description
Quantity	You can buy DB instances in batches. The default value is **1**. The value ranges from **1** to **10**.

If you have any questions about the price, click **Pricing details** at the bottom of the page.

☐ **NOTE**

The performance of a DB instance depends on its configuration. Hardware configuration items include the instance specifications, storage type, and storage space.

Step 3 Confirm the settings.

- If you need to modify your settings, click **Previous**.

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

Step 4 To view and manage instances, go to the **Instances** page.

- During the creation process, the instance status is **Creating**. After the status of the instance is **Available**, you can use the instance.

- Automated backup is enabled by default during instance creation. After your instance was created, the backup policy cannot be disabled and a full backup will be automatically created.

- After the instance is created, you can confirm the DB instance type on the **Instances** page.

- After the instance is created, you can add a description.

- The default database port is **3306**, but you can change it after instance creation is complete. To ensure data and instance security, change the database port immediately after the instance is created.

  For details, see **Changing a Database Port**.

----End

APIs

- **Creating a DB Instance**
- **Querying DB Instances**
- **Deleting a Pay-per-Use DB Instance**

2.2 Buying a Yearly/Monthly DB Instance

Scenarios

This section describes how to create a yearly/monthly DB instance on the GaussDB(for MySQL) console.

Billing

Yearly/Monthly DB instances are billed based on the purchase period. For more information, see **Yearly/Monthly Billing**.

Prerequisites

- You have **created a Huawei ID and enabled Huawei Cloud services**.

- You can create an IAM user or user group on the IAM console and grant it specific operation permissions, to perform refined management on Huawei Cloud. For details, see **Creating a User and Granting Permissions**.

- Your account balance is not below zero.

Procedure

Step 1 Go to the **Buy DB Instance** page.

Step 2 On the displayed page, configure required information and click **Next**.

Figure 2-10 Basic information

Table 2-9 Basic information

Parameter	Description
Billing Mode	Select **Yearly/Monthly**.
Region	A region where the DB instance is located. You can change this on the creation page, or go back to the **Instances** page and change it in the upper left corner. **NOTICE** Products in different regions cannot communicate with each other through a private network. After a DB instance is purchased, the region cannot be changed.
DB Instance Name	● If you create multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on. ● The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0

Parameter	Description
DB Instance Type	Select **Primary/Standby** or **Single**. ● **Primary/Standby**: A primary/standby instance contains one primary node and up to 15 read replicas. The primary node processes read and write requests, and the read replicas process only read requests. If the primary node becomes unavailable, GaussDB(for MySQL) automatically fails over to a read replica. Primary/standby instances apply to medium- and large-sized enterprises in the Internet, taxation, banking, and insurance sectors. ● **Single**: A single-node instance contains only one primary node and there are no read replicas. Single-node instances do not involve data synchronization of nodes and can easily ensure atomicity, consistency, isolation, and durability of transactions. They are only recommended for development and testing of microsites, and small and medium enterprises, or for learning about GaussDB(for MySQL).
Storage Type	● **DL6** The original **Shared storage**. The default storage type of GaussDB(for MySQL) instances created before July 2024 is **Shared storage**, while that of GaussDB(for MySQL) instances created after July 2024 is **DL6**. DL6-based instances achieve zero RPO with a 3 AZ deployment and deliver better performance and higher peak throughput. They are suitable for core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming. ● **DL5** A new storage type. It uses Huawei Cloud's hardware and network infrastructure technologies, ensuring that DL5-based instances maintain the same high availability as DL6-based instances with 3 AZs and zero RPO. Compared with DL6-based instances, although the peak performance of DL5-based instances may decrease, the cost per unit capacity is significantly reduced. DL5-based instances are suitable for CPU-intensive sub-core business systems or cost-effective application modules. For more information about storage types, see **Storage Types**.
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. ● **Single-AZ**: The primary node and read replicas are deployed in the same AZ. ● **Multi-AZ**: The primary node and read replicas are deployed in different AZs to ensure high reliability.

Parameter	Description
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.

Figure 2-11 Specifications and storage of a yearly/monthly DB instance
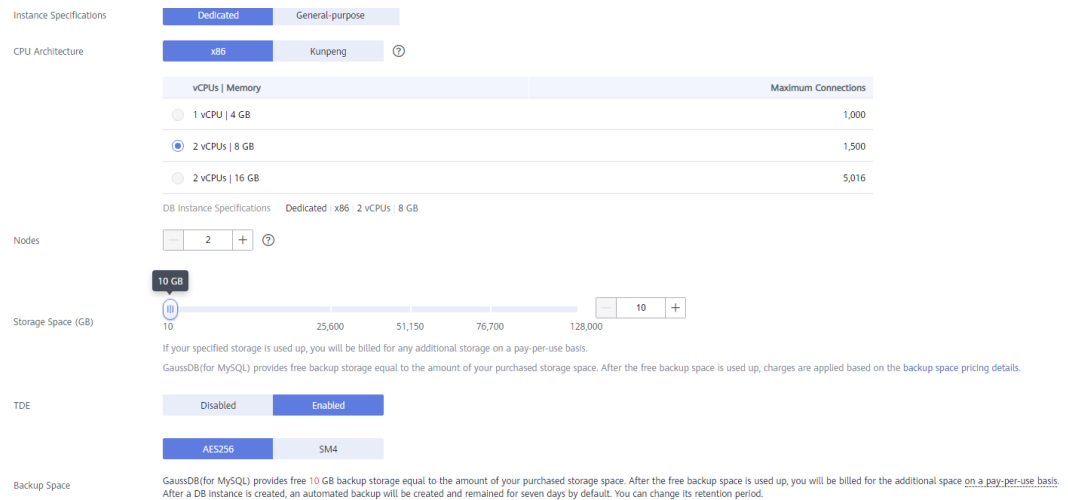
Table 2-10 Specifications and storage

Parameter	Description
Instance Specifications	GaussDB(for MySQL) is a cloud-native database that uses the shared storage. To ensure that DB instances run stably under high read/write pressure, GaussDB(for MySQL) controls the read/write peaks of DB instances based on instance specifications. For details about how to select specifications, see **Performance White Paper**. For more information about specifications, see **Instance Specifications**. After a DB instance is created, you can change its vCPUs and memory. For details, see **Changing the vCPUs or Memory of a DB Instance or Node**.

Parameter	Description
CPU Architecture	Select **x86** or **Kunpeng**. ● **x86**: x86 instances use Intel® Xeon® Scalable processors and feature robust and stable computing performance. When working on high-performance networks, the instances provide the additional performance and stability that enterprise-class applications demand. ● **Kunpeng**: Kunpeng instances use Kunpeng 920 processors and 25GE high-speed intelligent NICs for powerful compute and high-performance networks, making them an excellent choice for enterprises needing cost-effective, secure, and reliable cloud services.
Nodes	Total number of one primary node and read replicas you created for the instance. You can create up to 9 read replicas for a yearly/monthly instance at a time. After a DB instance is created, you can add read replicas based on service requirements. Up to 15 read replicas can be created for a DB instance. For details, see **Adding Read Replicas to a DB Instance**.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage space ranges from 40 GB to 128,000 GB and must be a multiple of 10. After a DB instance is created, you can change its storage space. **NOTE** If you want to create a DB instance with storage of at least 10 GB, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
TDE	Transparent Data Encryption (TDE) encrypts data files and backup files using certificates to implement real-time I/O encryption and decryption. This function effectively protects the security of databases and data files. After TDE is enabled, you need to select a cryptographic algorithm **AES256** or **SM4** as needed. **NOTE** ● To use TDE, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. ● For details about TDE constraints, see **Enabling TDE**.
Backup Space	GaussDB(for MySQL) provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis. If you purchase X GB storage billed on a yearly/monthly basis and Y GB storage billed on a pay-per-use basis, you will get (X + Y) GB backup space for free.

Figure 2-12 Network settings
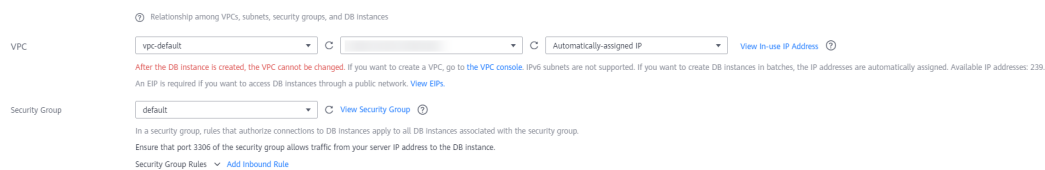
Table 2-11 Network

Parameter	Description
VPC	- A dedicated virtual network where your instances are located. It isolates networks for different workloads to enhance security. GaussDB(for MySQL) allocates a default VPC (default_vpc) for your instance. You can also use an existing, new, or shared VPC and subnet. - To use an existing VPC and subnet, select an existing VPC and subnet under the current account from the drop-down lists. - To use a new VPC and subnet, create a VPC and subnet, and then select the VPC and subnet from the drop-down lists. For details about how to create a VPC and subnet, see "Creating a VPC and Subnet" in *Virtual Private Cloud User Guide*. - To use a shared VPC and subnet, select a VPC and subnet that another account shares with the current account from the drop-down lists. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts so you can centrally manage resources in multiple accounts. This helps improve resource management efficiency and reduces O&M costs. For more information about VPC subnet sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*. **NOTICE** After a DB instance is created, the VPC cannot be changed.
Security Group	Enhances security by controlling access to GaussDB(for MySQL) from other services. When you select a security group, you must ensure that it allows the client to access instances. If no security group is available or has been created, GaussDB(for MySQL) allocates a security group to you by default. **NOTE** To ensure subsequent database connections and access, click **Add Inbound Rule** to allow all IP addresses to access your DB instance through port 3306 and over ICMP. For details, see **Configuring Security Group Rules**.

Figure 2-13 Proxy instance settings
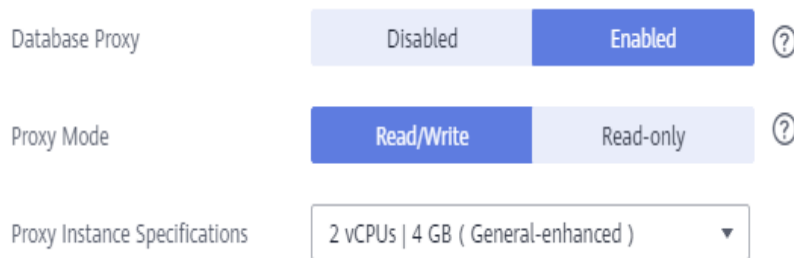
Table 2-12 Database proxy

Parameter	Description
Database Proxy	It is enabled by default. After the database proxy is enabled, you can use the IP address of a proxy instance to connect to your DB instance. **NOTE** ● To use this function, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. ● You can also create proxy instances after a DB instance is created. For details, see **Step 1: Create a Proxy Instance**.
Proxy Mode	You can select **Read/Write** or **Read-only** as needed. ● **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights. ● **Read-only**: The primary node does not process write and read requests, and all read requests are forwarded to the selected read replicas based on read weights.
Proxy Instance Specifications	You can select the proxy instance specifications as needed.

Figure 2-14 Database settings
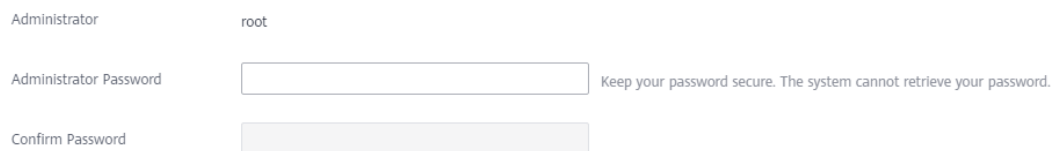
Table 2-13 Database settings

Parameter	Description
Administrator	The default login name for the database is **root**.

Parameter	Description
Administrator Password	Must consist of 8 to 32 characters and contain at least three of the following: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.). Enter a strong password and periodically change it to improve security and defend against threats such as brute force cracking attempts. **NOTICE** If you select a custom parameter template during instance creation, the administrator password must comply with the values of validate_password parameters in the custom parameter template. Otherwise, the instance creation will fail. To check the parameter values, go to the **Parameter Templates** page, find the target parameter template and click its name. In the upper right corner of the page, search for **validate_password**. **Figure 2-15** Checking the password-related parameters Keep this password secure. If lost, the system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see **Resetting the Administrator Password**.
Confirm Password	Must be the same as **Administrator Password**.

Figure 2-16 Other information settings

Table 2-14 Other information

Parameter	Description
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can view the default parameter template (**Default-GaussDB-for-MySQL 8.0**), high-performance parameter template (**Default-GaussDB-for-MySQL 8.0-High Performance)**, and all custom parameter templates in the current region. You can select an appropriate parameter template as required. **NOTICE** ● If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not applied. Instead, the default values are used. "innodb_buffer_pool_size" "innodb_log_buffer_size" "max_connections" "innodb_buffer_pool_instances" "innodb_page_cleaners" "innodb_parallel_read_threads" "innodb_read_io_threads" "innodb_write_io_threads" "threadpool_size" ● The value of **innodb_parallel_select_count** is determined by your instance specifications, instead of the parameter value you configured in the parameter template. The default value is **OFF** for instance with 16 vCPUs or less and **ON** for instances with more than 16 vCPUs. For more information about parameter templates, see **Parameter Template Management**. For more information about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. You can modify the instance parameters as required after a DB instance is created. For details, see **Modifying Parameters of a DB Instance**.
Table Name	Specifies whether table names are case sensitive. This option cannot be changed later. ● **Case sensitive**: Table names are case sensitive. ● **Case insensitive**: Table names are case insensitive and are stored in lowercase letters by default.

Parameter	Description
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service.
	An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis.
	You can select an enterprise project from the drop-down list. The default project is **default**.

Figure 2-17 Tag settings
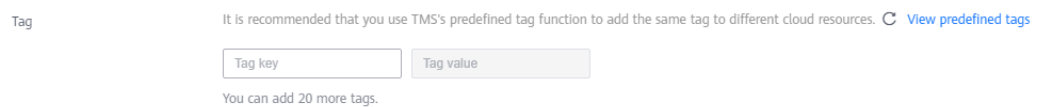
Table 2-15 Tags

Parameter	Description
Tag	(Optional) Tags a GaussDB(for MySQL) instance. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags.
	After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

Figure 2-18 Purchase period and quantity
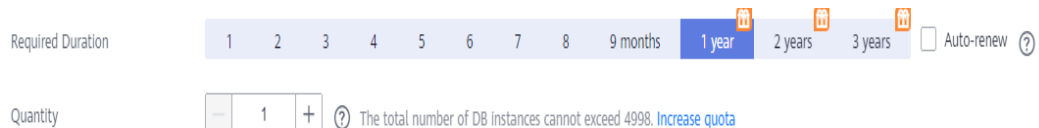
Table 2-16 Purchase period and quantity

Parameter	Description
Required Duration	This parameter is available only for yearly/monthly instances. The system will automatically calculate the fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	• This parameter is available only for yearly/monthly instances and is not selected by default.
	• If you select this parameter, the auto-renew cycle is determined by the selected required duration.

Parameter	Description
Quantity	You can create DB instances in batches. The default value is **1**. The value ranges from **1** to **10**.

If you have any questions about the price, click **Pricing details** at the bottom of the page.

◰ NOTE

> The performance of a DB instance depends on its configuration. Hardware configuration items include the instance specifications, storage type, and storage space.

Step 3 Confirm your order for yearly/monthly instances.

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

Yearly/Monthly instances are created only after you complete the payment.

Step 4 To view and manage instances, go to the **Instances** page.

- During the creation process, the instance status is **Creating**. After the status of the instance is **Available**, you can use the instance.
- Automated backup is enabled by default during instance creation. After your instance was created, the backup policy cannot be disabled and a full backup will be automatically created.
- After the instance is created, you can confirm the DB instance type on the **Instances** page.
- After the instance is created, you can add a description.
- The default database port is **3306**, but you can change it after instance creation is complete. To ensure data and instance security, change the database port immediately after the instance is created.

  For details, see **Changing a Database Port**.

----End

APIs

- **Creating a DB Instance**
- **Querying DB Instances**
- **Unsubscribing from a Yearly/Monthly DB Instance**

2.3 Buying a Serverless DB Instance

Scenarios

The capacities of serverless DB instances automatically change based on application requirements.

This section describes how to create a serverless DB instance on the GaussDB(for MySQL) console.

Constraints

Currently, instances billed at a serverless billing can be purchased in the following regions:

- CN North-Beijing4
- CN East-Shanghai1
- AP-Singapore

Billing

For details, see **Serverless Billing**.

Prerequisites

- You have **created a Huawei ID and enabled Huawei Cloud services**.
- You can create an IAM user or user group on the IAM console and grant it specific operation permissions, to perform refined management on Huawei Cloud. For details, see **Creating a User and Granting Permissions**.
- Your account balance is not below zero.

Procedure

Step 1 Go to the **Buy DB Instance** page.

Step 2 On the displayed page, configure required information and click **Next**.

Figure 2-19 Basic information

Table 2-17 Basic information

Parameter	Description
Billing Mode	Select **Serverless**.

Parameter	Description
Region	A region where the DB instance is located. You can change this on the creation page, or go back to the **Instances** page and change it in the upper left corner. **NOTICE** Products in different regions cannot communicate with each other through a private network. After a DB instance is purchased, the region cannot be changed.
DB Instance Name	● If you create multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on. ● The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0
DB Instance Type	Only primary/standby DB instances are supported. **Primary/Standby**: A primary/standby instance contains one primary node and up to seven read replicas. The primary node processes read and write requests, and the read replicas process only read requests. If the primary node becomes unavailable, GaussDB(for MySQL) automatically fails over to a read replica. Primary/standby instances apply to medium- and large-sized enterprises in the Internet, taxation, banking, and insurance sectors.
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. ● **Single AZ**: The primary node and read replicas are deployed in the same AZ. ● **Multi-AZ**: The primary node and read replicas are deployed in different AZs to ensure high reliability.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.

Figure 2-20 Specifications and storage of a serverless DB instance
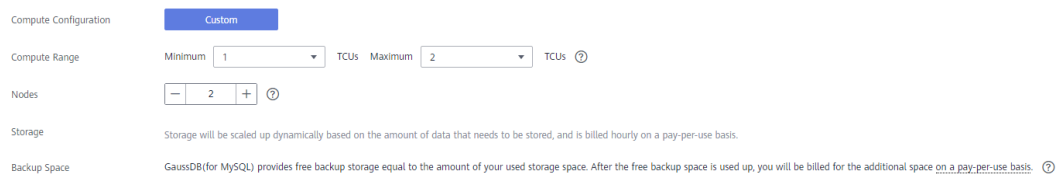
Table 2-18 Specifications and storage

Parameter	Description
Compute Configuration	Currently, only **Custom** is supported.
Compute Range	1 TCU is approximately equal to 1 vCPU and 2 GB of memory. Value range: 1 TCU to 32 TCUs
Nodes	Total number of one primary node and read replicas you created for the instance. You can create up to 8 nodes at a time.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
Backup Space	GaussDB(for MySQL) provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis.

Figure 2-21 Network settings
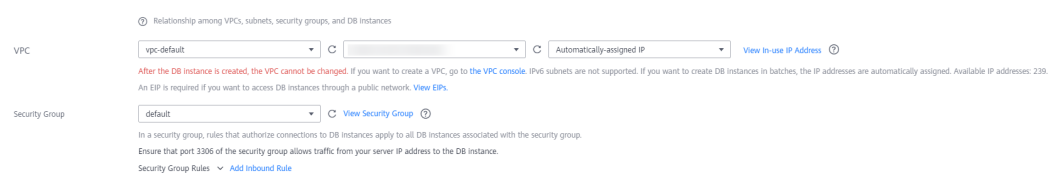
Table 2-19 Network

Parameter	Description
VPC	• A dedicated virtual network where your instances are located. It isolates networks for different workloads to enhance security. GaussDB(for MySQL) allocates a default VPC (default_vpc) for your instance. You can also use an existing, new, or shared VPC and subnet. – To use an existing VPC and subnet, select an existing VPC and subnet under the current account from the drop-down lists. – To use a new VPC and subnet, create a VPC and subnet, and then select the VPC and subnet from the drop-down lists. For details about how to create a VPC and subnet, see "Creating a VPC and Subnet" in **_Virtual Private Cloud User Guide_**. – To use a shared VPC and subnet, select a VPC and subnet that another account shares with the current account from the drop-down lists. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts so you can centrally manage resources in multiple accounts. This helps improve resource management efficiency and reduces O&M costs. For more information about VPC subnet sharing, see **VPC Sharing** in _Virtual Private Cloud User Guide_. **NOTICE** After a DB instance is created, the VPC cannot be changed.
Security Group	Enhances security by controlling access to GaussDB(for MySQL) from other services. When you select a security group, you must ensure that it allows the client to access instances. If no security group is available or has been created, GaussDB(for MySQL) allocates a security group to you by default. **NOTE** To ensure subsequent database connections and access, click **Add Inbound Rule** to allow all IP addresses to access your DB instance through port 3306 and over ICMP. For details, see **Configuring Security Group Rules**.

Figure 2-22 Proxy instance settings

Table 2-20 Database proxy

Parameter	Description
Database Proxy	It is enabled by default. After the database proxy is enabled, you can use the IP address of a proxy instance to connect to your DB instance. **NOTE** ● To use this function, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. ● You can also create proxy instances after a DB instance is created. For details, see **Step 1: Create a Proxy Instance**.
Proxy Mode	You can select **Read/Write** or **Read-only** as needed. ● **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights. ● **Read-only**: The primary node does not process write and read requests, and all read requests are forwarded to the selected read replicas based on read weights.
Proxy Instance Specifications	You can select the proxy instance specifications as needed.

Figure 2-23 Database settings

Table 2-21 Database settings

Parameter	Description
Administrator	The default login name for the database is **root**.

Parameter	Description

Administrator Password	Must consist of 8 to 32 characters and contain at least three of the following: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.). Enter a strong password and periodically change it to improve security and defend against threats such as brute force cracking attempts.
	
	NOTICE
	  If you select a custom parameter template during instance creation, the administrator password must comply with the values of validate_password parameters in the custom parameter template. Otherwise, the instance creation will fail.
	
	  To check the parameter values, go to the **Parameter Templates** page, find the target parameter template and click its name. In the upper right corner of the page, search for **validate_password**.
	
	  Figure 2-24 Checking the password-related parameters
	
	Keep this password secure. If lost, the system cannot retrieve it.
	
	After a DB instance is created, you can reset this password. For details, see **Resetting the Administrator Password**.

Confirm Password	Must be the same as **Administrator Password**.

Figure 2-25 Other information settings

Table 2-22 Other information

Parameter	Description
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can view the default parameter template (**Default-GaussDB-for-MySQL 8.0**), high-performance parameter template (**Default-GaussDB-for-MySQL 8.0-High Performance)**, and all custom parameter templates in the current region. You can select an appropriate parameter template as required. **NOTICE** - If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not applied. Instead, the default values are used. "innodb_buffer_pool_size" "innodb_log_buffer_size" "max_connections" "innodb_buffer_pool_instances" "innodb_page_cleaners" "innodb_parallel_read_threads" "innodb_read_io_threads" "innodb_write_io_threads" "threadpool_size" - The value of **innodb_parallel_select_count** is determined by your instance specifications, instead of the parameter value you configured in the parameter template. The default value is **OFF** for instance with 16 vCPUs or less and **ON** for instances with more than 16 vCPUs. For more information about parameter templates, see **Parameter Template Management**. For more information about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. You can modify the instance parameters as required after a DB instance is created. For details, see **Modifying Parameters of a DB Instance**.
Table Name	Specifies whether table names are case sensitive. This option cannot be changed later. **Case sensitive**: Table names are case sensitive. **Case insensitive**: Table names are case insensitive and are stored in lowercase letters by default.

Parameter	Description
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service.
	An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis.
	You can select an enterprise project from the drop-down list. The default project is **default**.

Figure 2-26 Tag settings
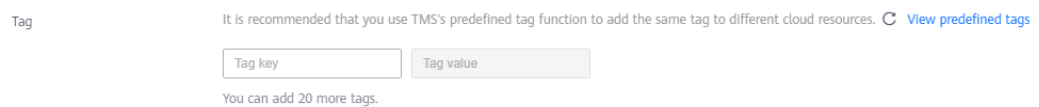
Table 2-23 Tags

Parameter	Description
Tag	(Optional) Tags a GaussDB(for MySQL) instance. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags.
	After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

 NOTE

> The performance of a DB instance depends on its configuration. Hardware configuration items include the instance specifications, storage type, and storage space.

Step 3 Confirm your specifications.

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

Step 4 To view and manage instances, go to the **Instances** page.

- During the creation process, the instance status is **Creating**. After the status of the instance is **Available**, you can use the instance.
- Automated backup is enabled by default during instance creation. After your instance was created, the backup policy cannot be disabled and a full backup will be automatically created.
- After the instance is created, you can confirm the DB instance type on the **Instances** page.
- After the instance is created, you can add a description.
- The default database port is **3306**, but you can change it after instance creation is complete. To ensure data and instance security, change the database port immediately after the instance is created.

For details, see **Changing a Database Port**.

----End

APIs

- **Creating a DB Instance**
- **Querying DB Instances**

3 Connecting to a DB Instance

3.1 Overview

You can connect to a GaussDB(for MySQL) instance through Data Admin Service (DAS), a private network, a public network, or JDBC.

Table 3-1 Connection methods

Connect Through	Connection Address	Description	Comments
DAS	Not required	DAS enables you to manage GaussDB(for MySQL) instances from a web-based console, simplifying database management and improving efficiency. By default, you have the remote login permission. It is recommended that you use DAS to connect to the instances because this connection method is more secure and convenient than other methods.	• Easy to use, secure, advanced, and intelligent • Recommended
Private network	Private IP address	A private IP address is provided by default. When your applications are deployed on an ECS that is in the same region and VPC as your GaussDB(for MySQL) instance, you are advised to connect the ECS to the instance over a private IP address.	• Secure and excellent performance • Recommended

Conne ct Throu gh	Connect ion Address	Description	Comments
Public netwo rk	EIP	If you cannot access your GaussDB(for MySQL) instance over a private IP address, bind an EIP to the instance and connect it to the ECS (or a public network host) over the EIP.	- A relatively lower level of security compared with other connection methods. - To achieve a higher data transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same VPC as your GaussDB(for MySQL) instance and use a private IP address to access the instance.
JDBC	Private IP address or EIP	JDBC is used to access GaussDB(for MySQL) instances.	-

NOTE

- VPC: indicates the Virtual Private Cloud.
- ECS: indicates the Elastic Cloud Server.
- You can log in to a DB instance using DAS or other database clients.
- If an ECS is in the same VPC as your GaussDB(for MySQL) instance, you do not need to apply for an EIP.
- If you are using GaussDB(for MySQL) for the first time, see **Constraints**.

3.2 Connecting to a DB instance Through DAS

Data Admin Service (DAS) is a one-stop management platform that allows you to manage Huawei Cloud databases on a web console. It offers database development, O&M, and intelligent diagnosis, making it easy to use and maintain databases.

This section describes how to connect to a DB instance through DAS.

Prerequisites

You have purchased a DB instance. If you have not, purchase one by referring to **Buying a DB Instance**.

Procedure
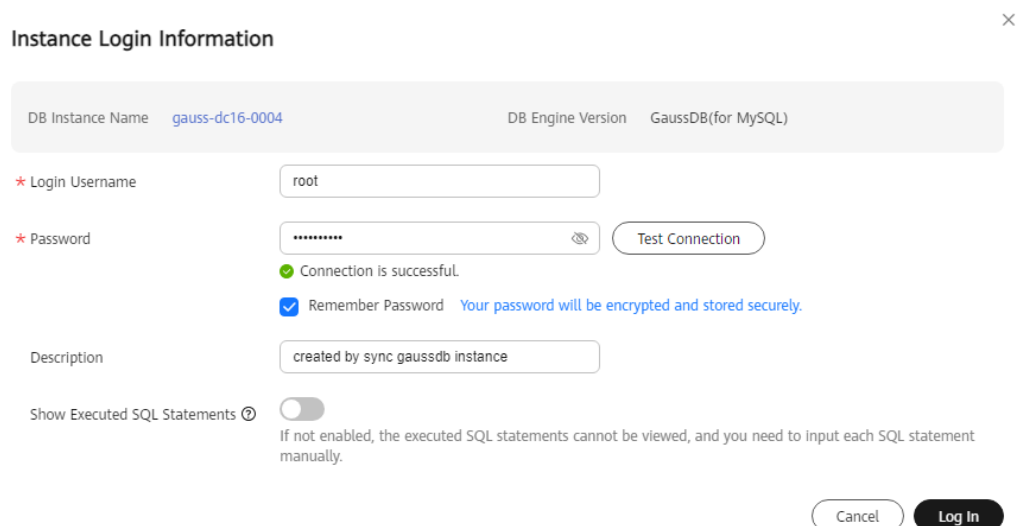
Step 1 **Log in to the management console.**

Step 2 Click ⓥ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate a DB instance and click **Log In** in the **Operation** column.

Figure 3-1 Logging in to a DB instance
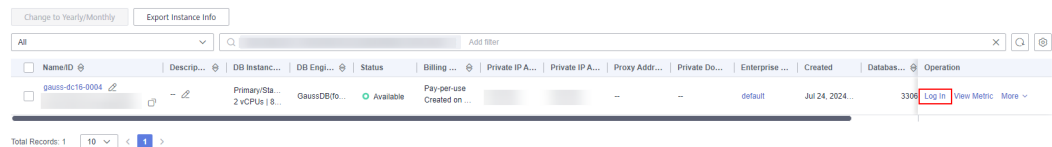
Step 5 Enter the login username and password and click **Test Connection**. After the connection test is successful, click **Log In**. Then you can access and manage your databases.

Figure 3-2 Login page

----End

Follow-up Operations

After logging in to a DB instance through DAS, you can manage your databases.

Step 1 Create a database.

After logging in to a GaussDB(for MySQL) instance, click **Create Database** on the home page, enter database information, and click **OK**.

Figure 3-3 Creating a database
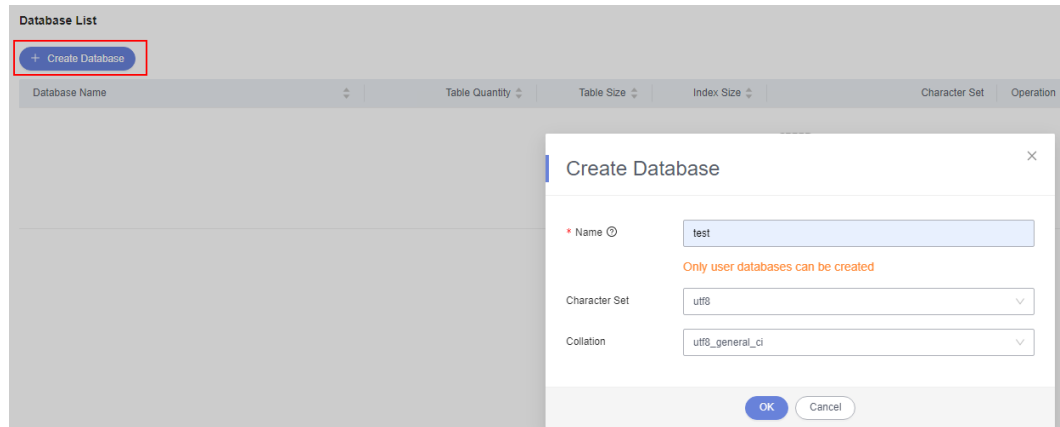
Database **test** is used as an example. After the database is created, you can view it in the database list.

Figure 3-4 Viewing the created database
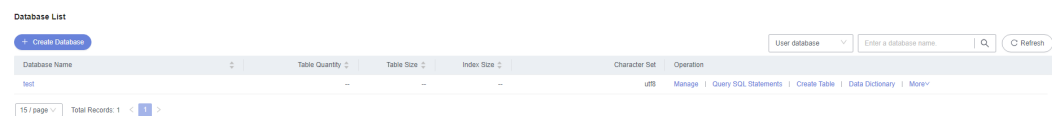
Step 2 Create a table.

Locate the database and click **Create Table** in the **Operation** column.

Figure 3-5 Creating a table
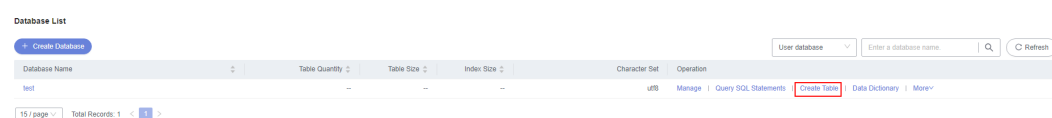
On the **Basic Information** tab, set the required parameters.

Figure 3-6 Entering basic table information

Click **Next** and enter column information.

Figure 3-7 Entering column information

Click **Create**. In the SQL preview window, view the SQL statements for creating a table and click **Execute**.

Figure 3-8 Previewing the SQL statements for creating a table

After the SQL statements are executed successfully, you can view the created table in the table list.

Figure 3-9 Viewing the created table
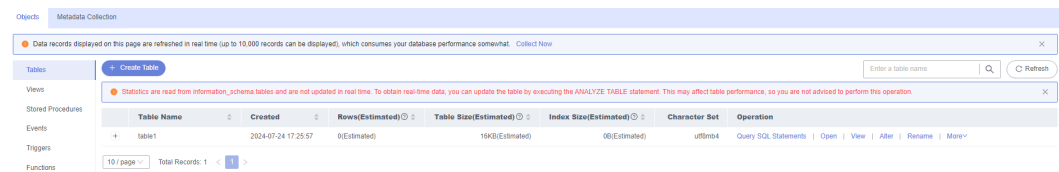
Step 3 Create a user and grant all permissions on the database created in **Step 1** to the user.

On the top menu bar, choose **Account Management** > **User Management**.

Figure 3-10 User management
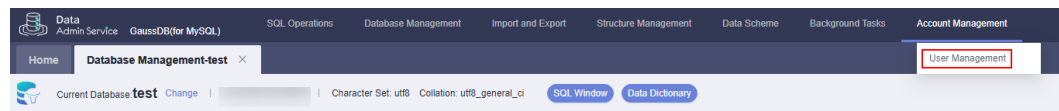
Click **Create User** and enter user information and authorization information.

Figure 3-11 Creating a user
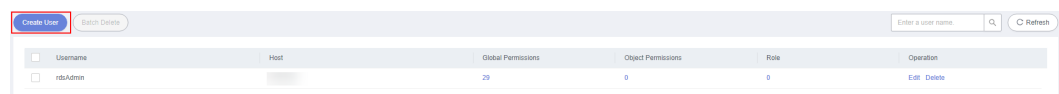
Figure 3-12 Entering user information and authorization information

For example, in the **Object Permissions** area, all permissions on the table (**table1**) in the database (**test**) are granted to the user (**user**).

Figure 3-13 Previewing the SQL statements for creating a user


```
SQL Preview                                              ×

1  /*COMMON SETTINGS*/
2  CREATE USER 'user'@'%' IDENTIFIED BY '******';
3
4  /*COMMON OBJECT SETTINGS*/
5  GRANT SELECT, EXECUTE ON `sys`.* TO 'user'@'%';
6  GRANT SELECT ON `performance_schema`.* TO 'user'@'%';
7
8  /*COMMON GLOBAL SETTINGS*/
9  GRANT PROCESS ON *.* TO 'user'@'%';

                   OK        Cancel
```

Figure 3-14 Viewing the created user

Step 4 Log in to the database as the created user and write data into the database.

On the DAS development tool page, add a database login as user **user**. Click **Log In** in the **Operation** column to log in to the DB instance.

Figure 3-15 Adding a login as **user**

In the row containing the **test** database, click **Query SQL Statements** in the **Operation** column. The SQL execution window is displayed.

Figure 3-16 Accessing the SQL execution window
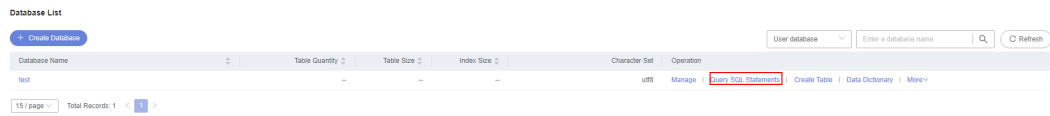
Run the following SQL statement in the SQL input box to query data in **table1**:

SELECT * FROM table1;

Figure 3-17 Viewing table data
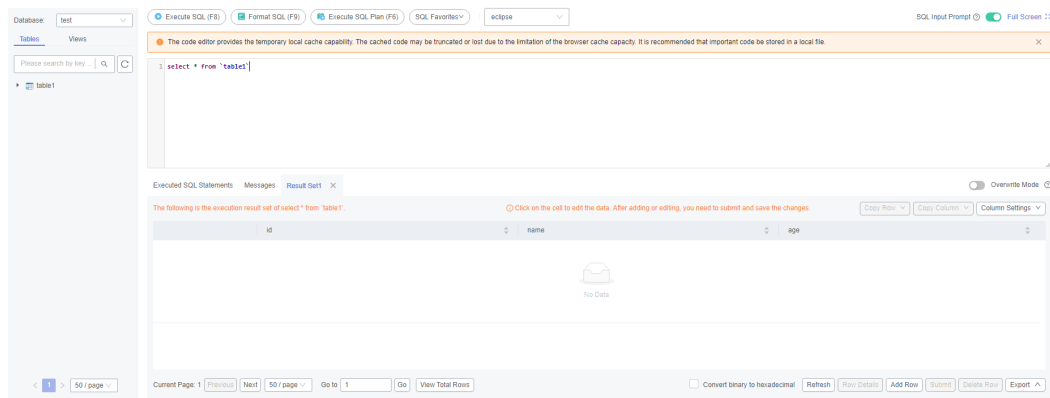
There is no data in **table1**.

Run the following SQL statements to write several data records to **table1**:

insert into table1(id, name, age) values(1, 'sam', 30);

insert into table1(id, name, age) values(2, 'cidy', 25);

insert into table1(id, name, age) values(3, 'lily', 27);

Figure 3-18 Writing data to the table
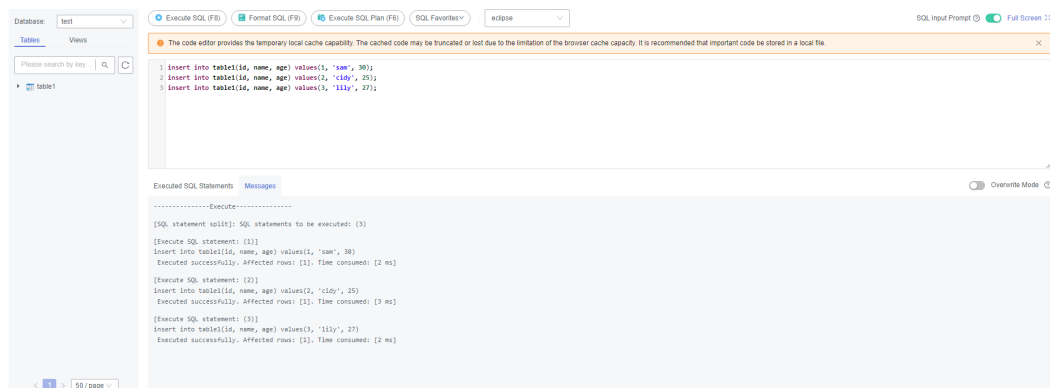
Data has been written into the table.

Run the following SQL statement again to check whether there is data in **table1**:

SELECT * FROM table1;

Figure 3-19 Verifying the written data
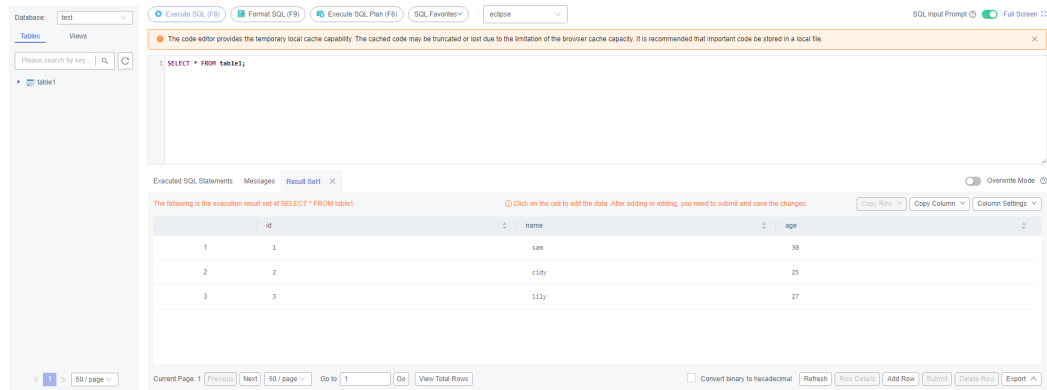
----End

3.3 Connecting to a DB Instance Using a MySQL Client

3.3.1 Using a MySQL Client to Connect to a DB Instance Over a Private Network

If your applications are deployed on an ECS that is in the same region and VPC as your DB instance, connect the ECS to the DB instance through a private IP address.

This section describes how to connect a Linux ECS to a DB instance with SSL enabled through a private IP address. SSL encrypts connections to the DB instance, making data more secure.

Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Linux ECS, go to **3**.
   - If there is a Windows ECS, see **Connecting to a DB Instance Using MySQL-Front**.
   - If no ECS is available, go to **2**.

   Figure 3-20 ECS
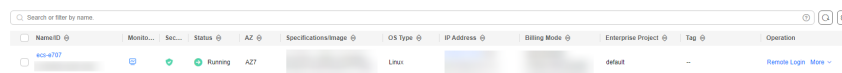
2. Buy an ECS and select Linux (for example, CentOS) as its OS.

   To download a MySQL client to the ECS, bind an EIP to the ECS. The ECS must be in the same region, VPC, and security group as the DB instance for mutual communications.

   For details about how to purchase a Linux ECS, see **Purchasing an ECS** in *Elastic Cloud Server Getting Started*.

3. On the **ECS Information** page, view the region and VPC of the ECS.

Figure 3-21 ECS information
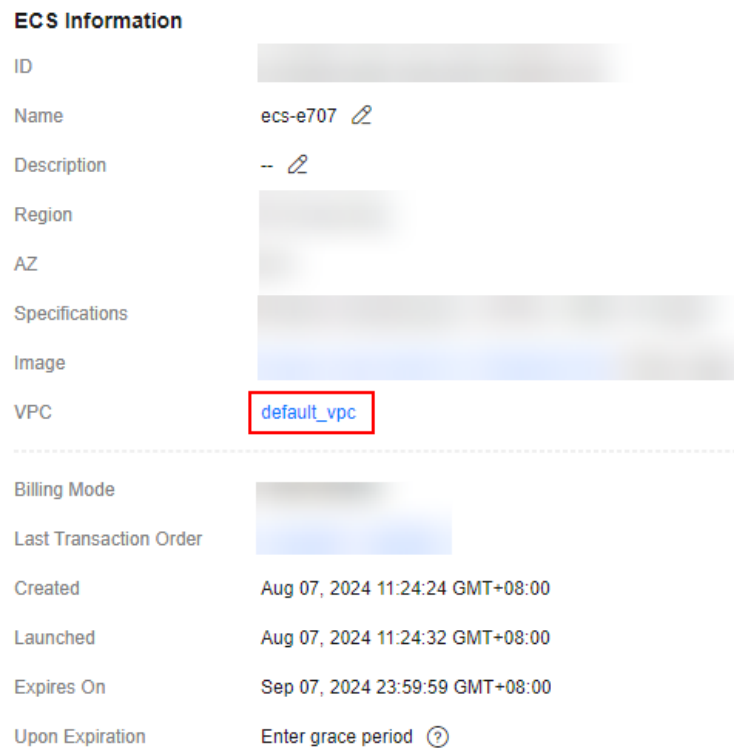
4. On the **Basic Information** page of the DB instance, view the region and VPC of the DB instance.

Figure 3-22 Viewing the region and VPC of the DB instance
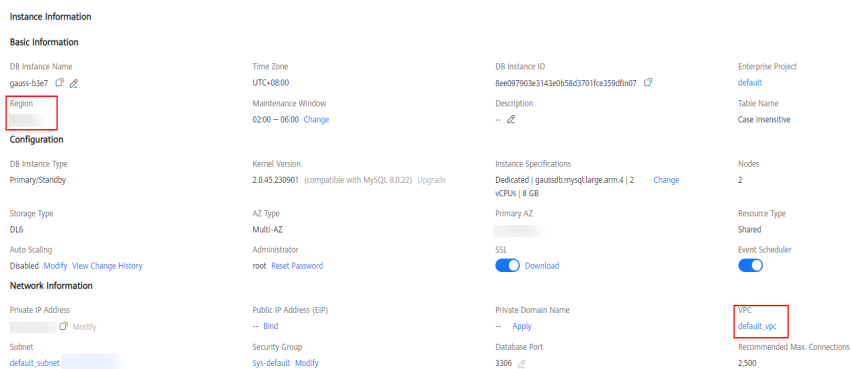
5. Check whether the ECS and DB instance are in the same region and VPC.

– If they are in the same region and VPC, go to **Step 2: Test Connectivity and Install a MySQL Client**.

– If they are in different regions, buy another instance. The ECS and DB instance in different regions cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.

– If they are in different VPCs, change the VPC of the ECS to that of the DB instance. For details, see **Changing a VPC**.

Step 2: Test Connectivity and Install a MySQL Client

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using an SSH Password** in *Elastic Cloud Server User Guide*.

2. On the **Instances** page of the GaussDB(for MySQL) console, click the instance name to go to the **Basic Information** page.

3. In the **Network Information** area, obtain the private IP address and database port.

   Figure 3-23 Viewing the private IP address and database port
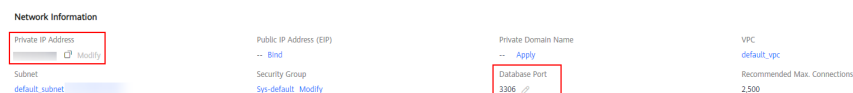
4. On the ECS, check whether the private IP address and database port of the DB instance can be connected.

   telnet *192.168.6.144 3306*

   – If yes, network connectivity is normal.

   – If no, check the security group rules.

     ▪ If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the private IP address and port of the DB instance.

     ▪ If in the security group of the DB instance, there is no inbound rule with **Source** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an inbound rule for the private IP address and port of the ECS. For details, see **Configuring Security Group Rules**.

   Figure 3-24 DB instance security group

5. Download the MySQL client installation package for Linux locally.

   Find the **corresponding version**, for example, **mysql-community-client-8.0.21-1.el6.x86_64.rpm**, and download the installation package. A MySQL client running a version later than that of the DB instance is recommended.

6. Upload the installation package to the ECS.

   You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

7. Run the following command on the ECS to install the MySQL client:

   rpm -ivh --nodeps mysql-community-client-8.0.21-1.el6.x86_64.rpm

📖 **NOTE**

- If any conflicts occur during the installation, add the **replacefiles** parameter to the command and try to install the client again.

  rpm -ivh --replacefiles mysql-community-client-8.0.21-1.el6.x86_64.rpm

- If a message is displayed prompting you to install a dependency package during the installation, add the **nodeps** parameter to the command and install the client again.

  rpm -ivh --nodeps mysql-community-client-8.0.21-1.el6.x86_64.rpm

Step 3: Connect to the DB Instance Using Commands (SSL Connection)

1. On the **Instances** page of the GaussDB(for MySQL) console, click the instance name to go to the **Basic Information** page.

2. In the **Instance Information** area, check whether SSL is enabled.

   - If yes, go to **3**.

   - If no, click ⬤. In the displayed dialog box, click **Yes** to enable SSL. Then, go to **3**.

3. Click ⬇ under **SSL** to download **Certificate Download.zip**, and obtain the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

4. Upload **ca.pem** to the ECS.

5. Run the following command on the ECS to connect to the DB instance:

   mysql -h *<host>* **-P** *<port>* **-u** *<userName>* **-p --ssl-ca=***<caName>*

   Example:

   mysql -h 192.168.0.79 -P 3306 -u root -p --ssl-ca=ca.pem

   Table 3-2 Parameter description

Parameter	Description
<host>	Private IP address of the DB instance.
<port>	Database port of the DB instance. The default value is **3306**.
<userName>	Administrator account **root**.
<caName>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

6. Enter the password of the database account if the following information is displayed:

   Enter password:

FAQs

What Should I Do If I Can't Connect to My GaussDB(for MySQL) Instance?

3.3.2 Using a MySQL Client to Connect to a DB Instance Over a Public Network

If you cannot access your DB instance through a private IP address, bind an EIP to the DB instance first and connect the ECS to the DB instance through the EIP.

This section describes how to connect a Linux ECS to a DB instance with SSL enabled through an EIP. SSL encrypts connections to the DB instance, making data more secure.

Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.

   – If there is a Linux ECS, go to **3**.

   – If there is a Windows ECS, see **Connecting to a DB Instance Using MySQL-Front**.

   – If no ECS is available, go to **2**.

2. Buy an ECS and select Linux (for example, CentOS) as its OS.

   To download a MySQL client to the ECS, bind an EIP to the ECS.

   For details about how to purchase a Linux ECS, see **Purchasing an ECS** in *Elastic Cloud Server Getting Started*.

3. On the **ECS Information** page, view the region and VPC of the ECS.

   Figure 3-25 ECS information

4. On the **Basic Information** page of the DB instance, view the region and VPC of the DB instance.

Figure 3-26 DB instance information
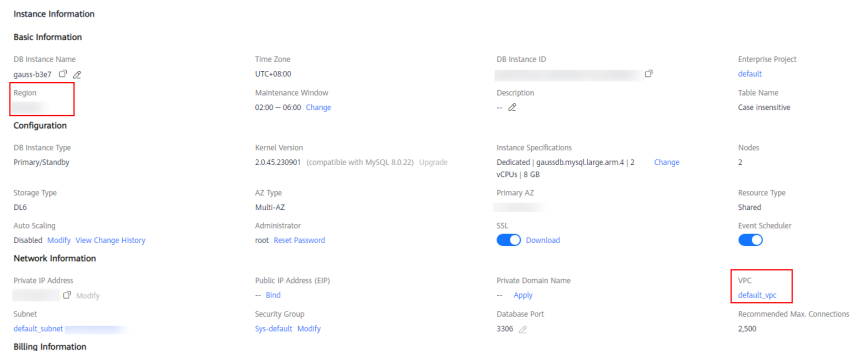
Step 2: Test Connectivity and Install a MySQL Client

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using an SSH Password** in *Elastic Cloud Server User Guide*.

2. On the **Instances** page of the GaussDB(for MySQL) console, click the instance name to go to the **Basic Information** page.

3. In the **Network Information** area, obtain the EIP and database port.

Figure 3-27 EIP and database port

If no EIP has been bound to the DB instance, see **Binding an EIP**.

4. On the ECS, check whether the EIP and database port of the DB instance can be connected.

telnet *EIP 3306*

– If yes, network connectivity is normal.

– If no, check the security group rules.

  ■ If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the EIP and port of the DB instance.

  ■ If in the security group of the DB instance, there is no inbound rule with **Source** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an inbound rule for the private IP address and port of the ECS. For details, see **Configuring Security Group Rules**.

Figure 3-28 DB instance security group

5. Download the MySQL client installation package for Linux locally.

   Find the **corresponding version**, for example, mysql-community-client-8.0.21-1.el6.x86_64.rpm, and download the installation package. A MySQL client running a version later than that of the DB instance is recommended.

6. Upload the installation package to the ECS.

   You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

7. Run the following command on the ECS to install the MySQL client:

   rpm -ivh --nodeps mysql-community-client-8.0.21-1.el6.x86_64.rpm

   ☐ NOTE

   - If any conflicts occur during the installation, add the **replacefiles** parameter to the command and try to install the client again.

     rpm -ivh --replacefiles mysql-community-client-8.0.21-1.el6.x86_64.rpm

   - If a message is displayed prompting you to install a dependency package during the installation, add the **nodeps** parameter to the command and install the client again.

     rpm -ivh --nodeps mysql-community-client-8.0.21-1.el6.x86_64.rpm

Step 3: Connect to the DB Instance Using Commands (SSL Connection)

1. On the **Instances** page of the GaussDB(for MySQL) console, click the instance name to go to the **Basic Information** page.

2. In the **Instance Information** area, check whether SSL is enabled.
   - If yes, go to **3**.

   - If no, click ⬤◯ . In the displayed dialog box, click **Yes** to enable SSL. Then, go to **3**.

3. Click ⬇ under **SSL** to download **Certificate Download.zip**, and obtain the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

4. Upload **ca.pem** to the ECS.

5. Run the following command on the ECS to connect to the DB instance:

   mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p --ssl-ca=**<*caName*>

   Example:

   mysql -h 172.16.0.31 -P 3306 -u root -p --ssl-ca=ca.pem

Table 3-3 Parameter description

Parameter	Description
<host>	EIP of the DB instance.
<port>	Database port of the DB instance. The default value is **3306**.
<userName>	Administrator account **root**.
<caName>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

6. Enter the password of the database account if the following information is displayed:
    Enter password:

FAQs

What Should I Do If I Can't Connect to My GaussDB(for MySQL) Instance?

3.4 Connecting to a DB Instance Using MySQL-Front

If your DB instance and ECS are not in the same region or VPC, you can connect to your DB instance using a Windows client through an EIP.

This section describes how to connect to a DB instance using a Windows ECS with the MySQL-Front client installed through an EIP.

Purchasing an ECS

Binding an EIP to a DB Instance

Querying the EIP of the DB Instance to Be Connected

Testing Connectivity and Installing MySQL-Front

Using MySQL-Front to Connect to a DB Instance

Purchasing an ECS

Step 1 **Log in to the management console** and check whether there is an ECS available.

- If there is a Linux ECS, see **Connecting to a DB Instance Using a MySQL Client**.
- If there is a Windows ECS, go to **Step 3**.
- If no ECS is available, go to **Step 2**.

Step 2 Buy an ECS and select Windows as its OS.

To download a MySQL client to the ECS, bind an EIP to the ECS.

For details about how to purchase a Windows ECS, see **Purchasing an ECS** in *Elastic Cloud Server Getting Started*.

Step 3 On the **ECS Information** page, view the region and VPC of the ECS.

Figure 3-29 Viewing ECS information
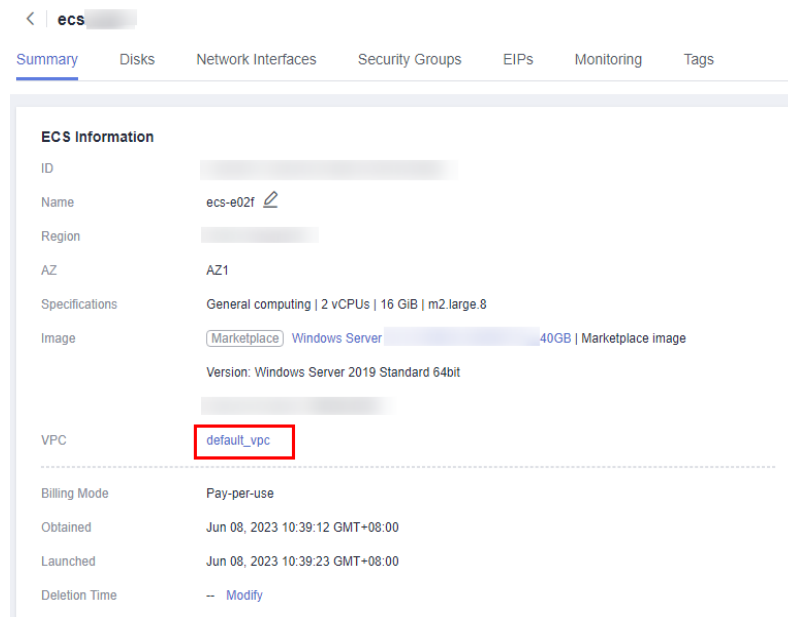
Step 4 On the **Basic Information** page of the DB instance, view the region and VPC of the DB instance.

Figure 3-30 Viewing the region and VPC of the DB instance
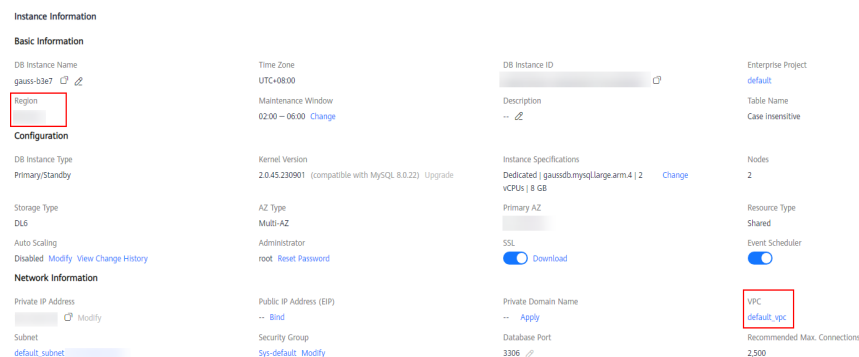
----End

Binding an EIP to a DB Instance

You can bind an EIP to a DB instance for public access and unbind it as required.

If an EIP has been bound to the DB instance, skip this step.

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

Step 2 Click [icon] in the upper left corner and select a region and project.

Step 3 Click [icon] in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click **Bind** under **Public IP Address (EIP)**.

Step 6 In the displayed dialog box, select an EIP and click **OK**.

If no EIPs are available, click **View EIP** to create an EIP on the network console. After the EIP is created, go back to the **Basic Information** page and bind the newly created EIP to the instance.

> **NOTICE**
>
> You need to configure security group rules and enable specific IP addresses and ports to access the DB instance. For details, see **Configuring Security Group Rules**.

Step 7 In the **Network Information** area, locate **Public IP Address (EIP)** and view the bound EIP.

----End

Querying the EIP of the DB Instance to Be Connected

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, obtain the EIP and database port.

Figure 3-31 Viewing the EIP and database port

----End

Testing Connectivity and Installing MySQL-Front

Step 1 Open the cmd window on your local server and check whether the EIP and database port of the DB instance can be connected.

telnet *EIP port*

Example:

telnet 192.168.0.16 3306

> ☐ NOTE
>
> If the message "command not found" is displayed, install the Telnet tool based on the OS used by the ECS.

- If yes, network connectivity is normal.
- If no, check the security group rules.
  - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the EIP and port of the DB instance.

Figure 3-32 Configuring rules of an ECS security group
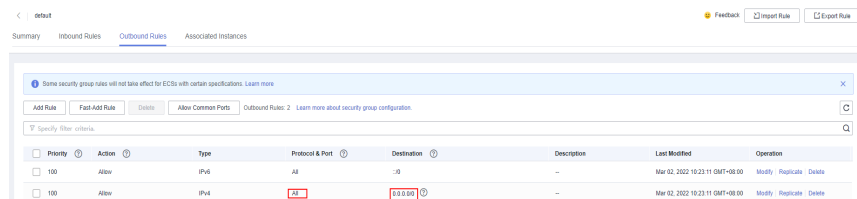
  - On the **Inbound Rules** page of the DB instance security group, add an inbound rule for the EIP and port of the ECS.. For details, see **Configuring Security Group Rules**.

Step 2 Open a browser, and download and install the MySQL-Front tool locally (version 5.4 is used as an example).

----End

Using MySQL-Front to Connect to a DB Instance

Step 1 Start MySQL-Front.

Step 2 In the displayed dialog box, click **New**.

Figure 3-33 Creating a connection
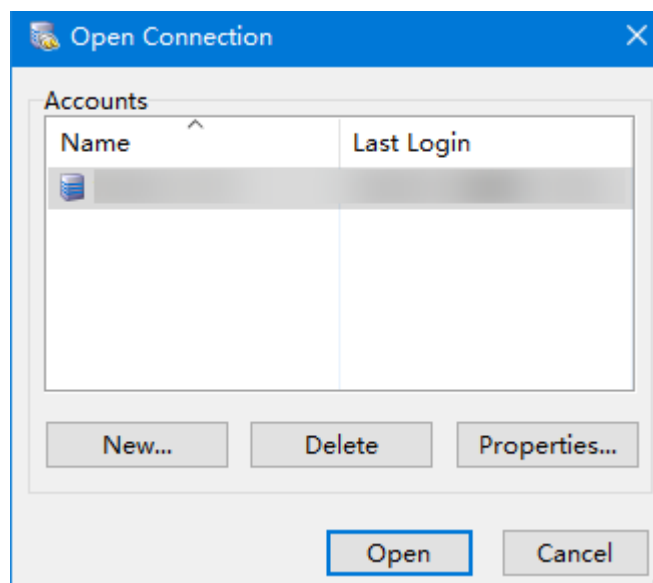
Step 3 Enter the information about the DB instance to be connected and click **Ok**.

Figure 3-34 Adding an account
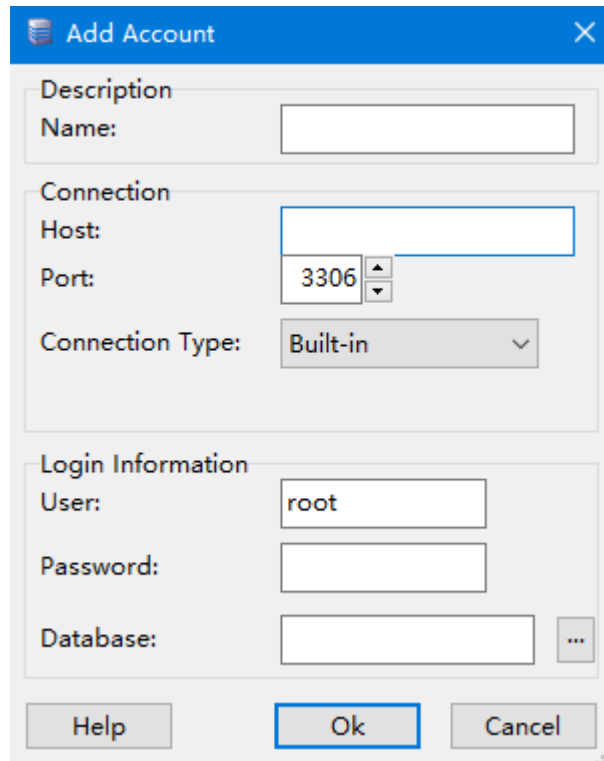
Table 3-4 Parameter description

Parameter	Description
Name	Database connection task name. If you do not specify this parameter, it will be the same as that configured for **Host** by default.
Host	Private IP address.
Port	Database port. The default value is **3306**.
User	Account name of the DB instance. The default value is **root**.
Password	Password of the account for accessing the DB instance.

Step 4 In the displayed window, select the connection that you created and click **Open**.

Figure 3-35 Opening a connection
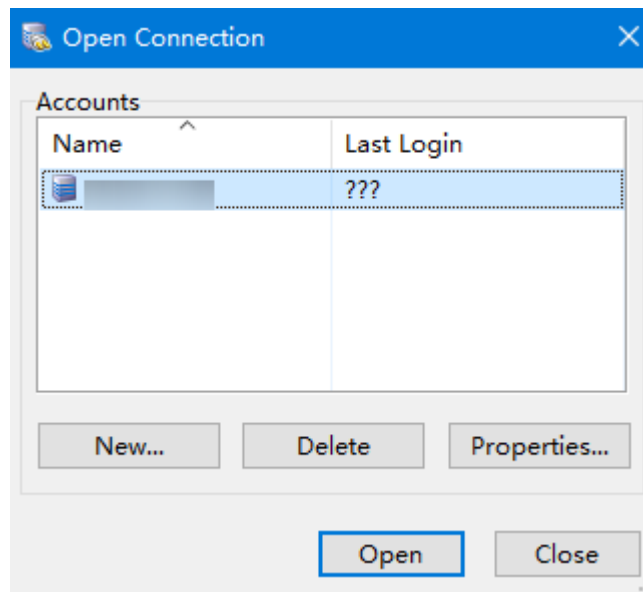
Step 5 Check whether the DB instance has been connected. If the connection information is correct, the DB instance has been connected.

Figure 3-36 Login succeeded
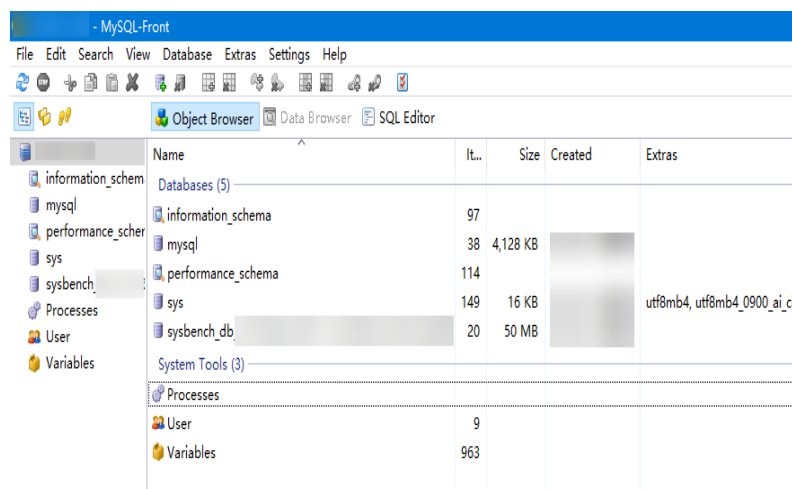
----End

FAQs

What Should I Do If I Can't Connect to My GaussDB(for MySQL) Instance?

3.5 Connecting to a DB Instance Using JDBC

Although the SSL certificate is optional if you choose to connect to a database through Java database connectivity (JDBC), you are advised to download the SSL

certificate to encrypt the connections for security purposes. By default, SSL is enabled for new GaussDB(for MySQL) instances. SSL encrypts connections to DB instances but prolongs connection response time and increases CPU usage. Before enabling SSL, evaluate the impact on service performance. For details about how to enable or disable SSL, see **Configuring SSL**.

Prerequisites

Familiarize yourself with:

- Computer basics
- Java programming language
- JDBC knowledge

Connection with the SSL Certificate

The SSL certificate needs to be downloaded and verified for connecting to databases.

> **NOTE**
>
> If the **ssl_type** value of a database user is **x509**, this method is unavailable.
>
> To check the **ssl_type** value of the current user, run the following command:
>
> select ssl_type from mysql.user where user = 'xxx';

Step 1 Download the CA certificate or certificate bundle.

1. On the **Instances** page, click the instance name to go to the **Basic Information** page.

2. Click **Download** under **SSL**.

Step 2 Use keytool to generate a truststore file using the CA certificate.

<keytool installation path> **./keytool.exe -importcert -alias** *<MySQLCACert>* **-file** *<ca.pem>* **-keystore** *<truststore_file>* **-storepass** *<password>*

Table 3-5 Parameter description

Parameter	Description
<keytool installation path>	Bin directory in the JDK or JRE installation path, for example, **C:\Program Files (x86)\Java\jdk11.0.7\bin**.
<MySQLCACert>	Name of the truststore file. Set it to a name specific to the service for future identification.
<ca.pem>	Name of the CA certificate downloaded and decompressed in **Step 1**, for example, **ca.pem**.
<truststore_file>	Path for storing the truststore file.
<password>	Password of the truststore file.

Code example (using keytool in the JDK installation path to generate the truststore file):

Owner: CN=MySQL_Server_8.0.22_Auto_Generated_CA_Certificate
Issuer: CN=MySQL_Server_8.0.22_Auto_Generated_CA_Certificate
Serial number: 1
Valid from: Thu Feb 16 11:42:43 EST 2017 until: Sun Feb 14 11:42:43 EST 2027
Certificate fingerprints:
   MD5: 18:87:97:37:EA:CB:0B:5A:24:AB:27:76:45:A4:78:C1
   SHA1: 2B:0D:D9:69:2C:99:BF:1E:2A:25:4E:8D:2D:38:B8:70:66:47:FA:ED
   SHA256:C3:29:67:1B:E5:37:06:F7:A9:93:DF:C7:B3:27:5E:09:C7:FD:EE:2D:18:86:F4:9C:40:D8:26:CB:DA:95:
A0:24
   Signature algorithm name: SHA256withRSA Subject Public Key Algorithm: 2048-bit RSA key
   Version: 1
   Trust this certificate? [no]: y
   Certificate was added to keystore

Step 3 Connect to your GaussDB(for MySQL) instance through JDBC.

jdbc:mysql://<*instance_ip*>**:**<*instance_port*>**/**<*database_name*>**?
requireSSL=**<*value1*>**&useSSL=**<*value2*>**&verifyServerCertificate=**<*value3*>**&trustCertificateKeyStoreUrl=f
ile:**
<*truststore_file*>**&trustCertificateKeyStorePassword=**<*password*>

Table 3-6 Parameter description

Parameter	Description
<*instance_ip*>	IP address of the DB instance. **NOTE** ● If you are accessing the instance through ECS, <*instance_ip*> is the private IP address of the instance. You can view the private IP address in the **Network Information** area on the **Basic Information** page. ● If you are accessing the instance through a public network, <*instance_ip*> is the EIP that has been bound to the instance. You can view the EIP in the **Network Information** area on the **Basic Information** page. ● If you are accessing the instance through a proxy instance, <*instance_ip*> is the proxy address. You can view the proxy address on the **Database Proxy** page.
<*instance_port*>	Database port of the DB instance. The default port is **3306**. **NOTE** You can view the database port in the **Network Information** area on the **Basic Information** page.
<*database_name*>	Database name used for connecting to the instance. The default value is **mysql**.
<*value1*>	Value of **requireSSL**, indicating whether the server supports SSL. It can be either of the following: ● **true**: The server supports SSL. ● **false**: The server does not support SSL. **NOTE** For details about the relationship between **requireSSL** and **sslmode**, see **Table 3-7**.

Parameter	Description

<value2>	Value of **useSSL**, indicating whether the client uses SSL to connect to the server. It can be either of the following:
	
	● **true**: The client uses SSL to connect to the server.
	● **false**: The client does not use SSL to connect to the server.
	    NOTE
	      For details about the relationship between **useSSL** and **sslmode**, see **Table 3-7**.

<value3>	Value of **verifyServerCertificate**, indicating whether the client verifies the server certificate. It can be either of the following:
	
	● **true**: The client verifies the server certificate.
	● **false**: The client does not verify the server certificate.
	    NOTE
	      For details about the relationship between **verifyServerCertificate** and **sslmode**, see **Table 3-7**.

<truststore_file>	Path for storing the truststore file configured in **Step 2**.
<password>	Password of the truststore file configured in **Step 2**.

Table 3-7 Relationship between connection parameters and sslmode

useSSL	requireSSL	verifyServerCer-tificate	sslMode
false	N/A	N/A	DISABLED
true	false	false	PREFERRED
true	true	false	REQUIRED
true	N/A	true	VERIFY_CA

Code example (Java code for connecting to a GaussDB(for MySQL) instance):

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;

public class JDBCTest {

 //There will be security risks if the username and password used for authentication are directly written
into code. Store the username and password in ciphertext in the configuration file or environment variables.
 //In this example, the username and password are stored in the environment variables. Before running the
code, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.
   static final String USER = System.getenv("EXAMPLE_USERNAME_ENV");
   static final String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");

   public static void main(String[] args) {
```

```
        Connection conn = null;
        Statement stmt = null;

        String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
requireSSL=true&useSSL=true&verifyServerCertificate=true&trustCertificateKeyStoreUrl=file:
<truststore_file>&trustCertificateKeyStorePassword=<password>";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, USER, PASS);

            stmt = conn.createStatement();
            String sql = "show status like 'ssl%'";
            ResultSet rs = stmt.executeQuery(sql);

            int columns = rs.getMetaData().getColumnCount();
            for (int i = 1; i <= columns; i++) {
                System.out.print(rs.getMetaData().getColumnName(i));
                System.out.print("\t");
            }

            while (rs.next()) {
                System.out.println();
                for (int i = 1; i <= columns; i++) {
                    System.out.print(rs.getObject(i));
                    System.out.print("\t");
                }
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // release resource ....
        }
    }
}
```

----End

Connection Without the SSL Certificate

☐ NOTE

You do not need to download the SSL certificate because certificate verification on the server is not required.

Step 1 Connect to your GaussDB(for MySQL) instance through JDBC.

jdbc:mysql://*<instance_ip>*:*<instance_port>***/***<database_name>***?useSSL**=false

Table 3-8 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** ● If you are accessing the instance through ECS, *<instance_ip>* is the private IP address of the instance. You can view the private IP address in the **Network Information** area on the **Basic Information** page. ● If you are accessing the instance through a public network, *<instance_ip>* is the EIP that has been bound to the instance. You can view the EIP in the **Network Information** area on the **Basic Information** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can view the database port in the **Network Information** area on the **Basic Information** page.
<database_name>	Database name used for connecting to the instance. The default value is **mysql**.

Code example (Java code for connecting to a GaussDB(for MySQL) instance):

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;
        // set sslmode here.
        // no ssl certificate, so do not specify path.
        String url = "jdbc:mysql://192.168.0.225:3306/my_db_test?useSSL=false";
        try {
            Class.forName("com.mysql.jdbc.Driver");
                    //There will be security risks if the username and password used for authentication are
directly written into code. Store the username and password in ciphertext in the configuration file or
environment variables.
                    //In this example, the username and password are stored in the environment variables.
Before running the code, set environment variables EXAMPLE_USERNAME_ENV and
EXAMPLE_PASSWORD_ENV as needed.
                    conn = DriverManager.getConnection(url, System.getenv("EXAMPLE_USERNAME_ENV"),
System.getenv("EXAMPLE_PASSWORD_ENV"));
            System.out.println("Database connected");

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM mytable WHERE columnfoo = 500");
            while (rs.next()) {
                System.out.println(rs.getString(1));
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Test failed");
        } finally {
            // release resource ....
        }
```

```
    }
}
```

----End

Related Issues

- Symptom

  When you use JDK 8.0 or a later version to connect to your instance with an SSL certificate downloaded, an error similar to the following is reported:

  ```
  javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or
  cipher suites are inappropriate)
      at sun.security.ssl.HandshakeContext.<init>(HandshakeContext.java:171) ~[na:1.8.0_292]
      at sun.security.ssl.ClientHandshakeContext.<init>(ClientHandshakeContext.java:98) ~
  [na:1.8.0_292]
      at sun.security.ssl.TransportContext.kickstart(TransportContext.java:220) ~
  [na:1.8.0_292]
      at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:428) ~
  [na:1.8.0_292]
      at
  com.mysql.cj.protocol.ExportControlled.performTlsHandshake(ExportControlled.java:316) ~
  [mysql-connector-java-8.0.17.jar:8.0.17]
      at
  com.mysql.cj.protocol.StandardSocketFactory.performTlsHandshake(StandardSocketFactory.java
  :188) ~[mysql-connector-java8.0.17.jar:8.0.17]
      at
  com.mysql.cj.protocol.a.NativeSocketConnection.performTlsHandshake(NativeSocketConnection.
  java:99) ~[mysql-connector-java8.0.17.jar:8.0.17]
      at
  com.mysql.cj.protocol.a.NativeProtocol.negotiateSSLConnection(NativeProtocol.java:331) ~
  [mysql-connector-java8.0.17.jar:8.0.17]
  ... 68 common frames omitted
  ```

- Solution

  Specify the corresponding parameter values in the code link of **Step 3** based on the JAR package used by the client. Example:

  – mysql-connector-java-5.1.*xx*.jar (For 8.0.18 and earlier versions, use the **enabledTLSProtocols** parameter. For details, see **Connecting Securely Using SSL**.)

    jdbc:mysql://<*instance_ip*>**:**<*instance_port*>**/**<*database_name*>**?**

    requireSSL=true**&useSSL**=true**&verifyServerCertificate**=true**&trustCertificateKeyStoreUrl=file:**

    <*truststore_file*>**&trustCertificateKeyStorePassword**=<*password*>**& enabledTLSProtocols=TLSv1.2**

  – mysql-connector-java-8.0. *xx*.jar (For connection drivers later than 8.0.18, use the **tlsVersions** parameter.)

    jdbc:mysql://<*instance_ip*>**:**<*instance_port*>**/**<*database_name*>**?**

    requireSSL=true&**useSSL**=true&**verifyServerCertificate**=true&**trustCertificateKeyStoreUrl=file:**
    <*truststore_file*>**&trustCertificateKeyStorePassword**=<*password*>**& tlsVersions** =TLSv1.2

3.6 Connection Information Management

3.6.1 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and GaussDB(for MySQL) instances that are within the same VPC, have the same security

requirements, and are mutually trusted. To ensure database security and reliability, you need to configure security group rules to allow only specific IP addresses and ports to access the GaussDB(for MySQL) instances.

When you attempt to connect to a GaussDB(for MySQL) instance through a private network, check whether the ECS and GaussDB(for MySQL) instance are in the same security group.

- If they 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, you need to configure security group rules for the ECS and GaussDB(for MySQL) instance, respectively.
  - GaussDB(for MySQL) instance: Configure an **inbound rule** for the security group with which the GaussDB(for MySQL) instance is associated.
  - ECS: The default security group rule allows all outbound data packets. In this case, you do not need to configure a security group rule for the ECS. If **not all outbound traffic** is allowed in the security group, you may need to configure an **outbound rule** for the ECS to allow all outbound packets.

This section describes how to configure an inbound rule for a GaussDB(for MySQL) instance.

For details about the requirements of security group rules, see **Adding a Security Group Rule** in the *Virtual Private Cloud User Guide*.

Precautions

The default security group rule allows all outbound data packets. This means that ECSs and GaussDB(for MySQL) instances associated with the same security group can access each other by default. After a security group is created, you can configure security group rules to control access to and from GaussDB(for MySQL) instances associated with that security group.

- By default, you can create up to 500 security group rules.
- Too many security group rules will increase the first packet latency. You are advised to create up to 50 rules for each security group.
- One instance can be associated with only one security group.
- To access a GaussDB(for MySQL) instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the instance.

📖 NOTE

To ensure data and instance security, use permissions properly. You are advised to use the minimum access permission, change the default database port **3306**, and set the accessible IP address to the remote server's address or the remote server's minimum subnet address to control the access scope of the remote server.

The default value of **Source** is **0.0.0.0/0**, indicating that all IP addresses can access the GaussDB(for MySQL) instance as long as they are associated with the same security group as the instance.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 Configure security group rules.

In the **Network Information** area, click the security group name under **Security Group**.

Figure 3-37 Configuring security group rules

Step 6 On the **Inbound Rules** tab, click **Add Rule**. In the displayed dialog box, set required parameters and click **OK**.

You can click ⊕ to add more inbound rules.

Figure 3-38 Adding inbound rules
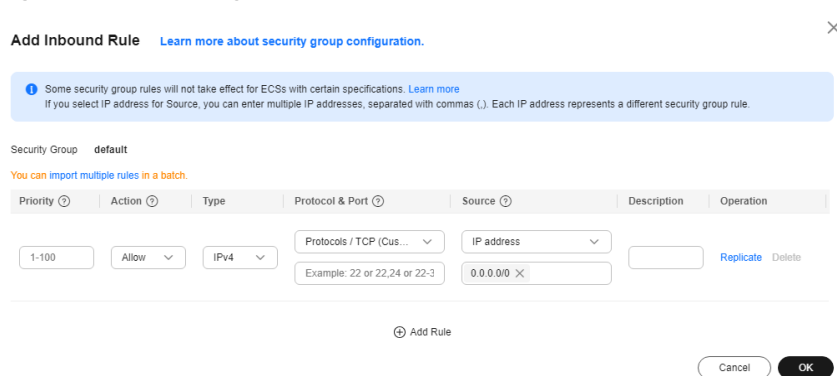
Table 3-9 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	Network protocol for which the security group rule takes effect. • Currently, the value can be **All**, **TCP (All ports)**, **TCP (Custom ports)**, **UDP (All ports)**, **UDP (Custom ports)**, **ICMP**, **GRE**, or others. • **All**: indicates all protocol ports are supported.	TCP (Custom ports)
	Port: the port over which the traffic can reach your DB instance.	When connecting to the instance through a private network, enter the port of the instance. • Individual port: Enter a port, such as **22**. • Consecutive ports: Enter a port range, such as **22-30**. • All ports: Leave it empty or enter **1-65535**.
Type	Currently, only **IPv4** and **IPv6** are supported.	IPv4
Source	Source of the security group rule. The value can be a security group or an IP address. xxx.xxx.xxx.xxx/32 (IPv4 address) xxx.xxx.xxx.0/24 (subnet) 0.0.0.0/0 (any IP address)	0.0.0.0/0
Description	Supplementary information about the security group rule. This parameter is optional. The description can contain up to 255 characters and cannot contain angle brackets (<>).	-
Operation	You can replicate or delete a security group rule. However, if there is only one security group rule, you cannot delete it.	-

----End

3.6.2 Binding an EIP

Scenarios

You can bind an EIP to a GaussDB(for MySQL) instance for public access and unbind it as required.

Precautions

- Binding EIPs to DB instances reduces the security of the DB instances. Exercise caution when performing this operation. To achieve a higher transmission rate and security level, you are advised to migrate your applications to the ECS that is in the same region as the GaussDB(for MySQL) instance.
- Traffic generated by the public network is billed. You can unbind the EIP from your DB instance when the EIP is no longer used.
- After an EIP billed on a pay-per-use basis is unbound from a GaussDB(for MySQL) instance, it is still billed. To save money, you can release the EIP or bind it to another DB instance.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click **Bind** under **Public IP Address (EIP)**.

Step 6 In the displayed dialog box, select an EIP and click **OK**.

If no EIPs are available, click **View EIP** to create an EIP on the network console. After the EIP is created, go back to the **Basic Information** page and bind the newly created EIP to the instance.

> **NOTICE**
>
> You need to configure security group rules and enable specific IP addresses and ports to access the DB instance.

Step 7 In the **Network Information** area, locate **Public IP Address (EIP)** and view the bound EIP.

----End

3.6.3 Changing a Database Port

You can change the database port of a GaussDB(for MySQL) instance.

Constraints

- The database port of a DB instance with database proxy enabled cannot be changed.
- If there is an HTAP instance, the database port of the GaussDB(for MySQL) instance cannot be changed.
- The change will be applied to the ports of the primary node and read replicas.
- If you change the database port of a DB instance, the ports of the primary node and read replicas are changed accordingly and all of them are rebooted.
- It takes about 1 to 5 minutes to change a database port.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click ✐ under **Database Port**.

The database port of a GaussDB(for MySQL) instance ranges from 1025 to 65534, excluding 5342, 5343, 5344, 5345, 12017, 20000, 20201, 20202, 33060, 33062, and 33071, which are reserved for system use.

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

----End

APIs

Changing a Database Port

3.6.4 Applying for and Changing a Private Domain Name

You can use a private network domain name to connect to a GaussDB(for MySQL) instance.

After your DB instance is created, you can change the private domain name as needed.

Constraints

- Domain Name Service (DNS) is deployed.
- Changing the private domain name will interrupt your database connection. To reconnect to the DB instance, change the connection address of your applications. The new private domain name is applied to the instance about 5 minutes after the change.

Applying for a Private Domain Name

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click **Apply** under **Private Domain Name**.

Step 6 View the generated private domain name under **Private Domain Name**.

----End

Changing a Private Domain Name

Step 1 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 2 In the **Network Information** area, click **Modify** under **Private Domain Name**.

Figure 3-39 Modifying a private domain name

Step 3 In the displayed dialog box, enter a new domain name and click **OK**.

📖 NOTE

- Only the prefix of a private domain name can be modified.
- The prefix of a private domain name contains 8 to 63 characters, and can include only lowercase letters and digits.
- The new private domain name must be different from existing ones.

Step 4 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see *Identity and Access Management User Guide*.

----End

3.6.5 Configuring and Changing a Private IP Address

Scenarios

You can change private IP addresses after migrating data from on-premises databases or other cloud databases to GaussDB(for MySQL).

Constraints

- After read/write splitting is enabled, the private IP address cannot be changed.
- If there is an HTAP instance, the private IP address of the GaussDB(for MySQL) instance cannot be changed.
- After a private IP address is changed, the domain name needs to be resolved again. This operation takes several minutes and may interrupt database connections. You are advised to change a private IP address during off-peak hours.

Configuring the Private IP Address of a DB Instance

When you buy an instance, select a VPC and subnet on the **Buy DB Instance** page. Then, a private IP address will be automatically assigned to your instance. You can also enter a private IP address.

Procedure

You can change the private IP address of an existing GaussDB(for MySQL) instance.

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click **Modify** under **Private IP Address**.

Step 6 In the displayed dialog box, enter a new private IP address and click **OK**.

An in-use IP address cannot be used as the new private IP address of the DB instance.

Step 7 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

----End

APIs

Changing a Private IP Address

4 Database Usage

4.1 Usage Guidelines

4.1.1 Database Permissions

- All DDL operations (such as creating tables and modifying table structures) are performed by DBAs through DAS only after being reviewed. Services are launched during off-peak hours.
- Permissions must be managed in a fine-grained manner by separating read permissions from write permissions, and O&M permissions from development permissions.
- DDL operations are recorded in operation logs.

4.1.2 Table Design

- All created MySQL tables must use the InnoDB engine.
- The decimal type must be DECIMAL. Do not use FLOAT or DOUBLE.

  FLOAT and DOUBLE have lower precision than DECIMAL and may cause rounding errors. If a value to be stored is beyond the range of DECIMAL, split the value into INTEGER and DECIMAL parts and store them separately.
- The following reserved words cannot be used: DESC, RANGE, MATCH, and DELAYED.

  For details about the keywords and reserved words of MySQL 8.0 Community Edition, see **Keywords and Reserved Words**.

  In addition to the keywords and reserved words of MySQL 8.0 Community Edition, some other keywords and reserved words are added to GaussDB(for MySQL). Do not use these keywords and reserved words when naming objects.

  Table 4-1 lists the new keywords and reserved words in GaussDB(for MySQL).

Table 4-1 GaussDB(for MySQL) new keywords and reserved words

Reserved Word	Related Scenario
EXTRA_HEALTH	High availability
PBS	Backup and restoration
REDO	Primary/standby replication
SLICEID	Shared storage
SLOWIO	Shared storage
SPACEUSAGE	Shared storage
RDS_INSTANT	Recycle bin
RECYCLE_BIN	Recycle bin
RDS_RECYCLE	Recycle bin
RDS_TAC	Recycle bin
RDS_GDB_CTRL	RegionlessDB

- Every data table must have a primary key, which can be either an ordered and unique field related to business or an auto-increment field unrelated to business.

- Each table field must have a default value and NOT NULL. If the field is the numeric type, use 0 as its default value. If the field is the character type (such as VARCHAR), use an empty string (").

  ◯ **NOTE**

    The absence of a primary key may cause slow execution of the primary database and replication delay.

- You are not advised to use partitioned tables. If necessary, use multiple independent tables.

  ◯ **NOTE**

    Disadvantages of partitioned tables:

    - All partitions will be locked during DDL operations. As a result, operations on the partitions will be blocked.

    - When a partitioned table contains a large amount of data, it is difficult and risky to perform DDL or other O&M operations on the table.

    - Partition tables are seldom used, which may cause unknown risks.

    - When a single server is poor in performance, splitting a partitioned table is expensive.

    - When all partitions are accessed due to improper operations on a partitioned table, severe performance problems may occur.

- Each table contains two DATETIME fields: **CREATE_TIME** and **UPDATE_TIME**.

You can obtain the required data from a data warehouse based on these two fields without consulting services.

When an exception occurs in the database, you can use the two fields to determine the time when the data is inserted and updated. In extreme cases, you can determine whether to restore data based on the fields.

- VARCHAR is a variable-length character data type. The length of VARCHAR cannot exceed 2,048.

  If the length of a field exceeds 2,048, define the field type as TEXT or create an independent table and use a primary key to associate the related tables. In this way, the index efficiency of other fields is not affected.

- The length of a single row in a table cannot exceed 1,024 bytes.

- The maximum number of fields in a single table is 50.

- If the lengths of all strings are almost the same, use the fixed-length character strings.

- On the premise of ensuring data consistency, cross-table redundant fields are allowed to avoid join queries and improve query performance.

  ☐ NOTE

  Redundant fields must comply with the following rules:
  - Fields are not frequently modified.
  - Fields are not large VARCHAR and TEXT.

- The data types with proper storage size can save database tablespace and index storage space while improving the search speed. LONG TEXT and BLOB are not recommended.

- Ensure that all characters are stored and represented in UTF-8 or utf8mb4 encoding. Comments must be provided for tables and fields.

- Avoid using large transactions.

  For example, if multiple SELECT and UPDATE statements are executed in a high-frequency transaction, the database concurrency capability is severely affected because resources such as locks held by the transaction can be released only when the transaction is rolled back or committed. In this case, data write consistency must also be considered.

- Full-text indexes are not recommended because there are many limitations on them.

- For ultra-large tables, you also need to comply with the following rules:
  - Use TINYINT, SMALLINT, and MEDIUM_INT as integer types instead of INT. If a value is non-negative, add UNSIGNED. Keep the field type as short as possible while meeting service evolution requirements.
  - Configure the VARCHAR length as needed.

    Example:

    CREATE TABLE T1 (A VARCHAR(255));

    After optimization:

    CREATE TABLE T1 (A VARCHAR(*Length that meets service requirements*));
  - Use enumerations or integers instead of strings.

- Use TIMESTAMP instead of DATETIME.

- Keep the number of fields in a single table below 20.

- Avoid using UNIQUE. Programs can enforce the constraints.

- Store IP addresses as integers.

- Partition fields with strong sequence and add range conditions during queries to improve efficiency.

- If there is obvious hot data and cold data, place the hot data in a separate partition.

- Use a proxy instance to connect to a database. In scenarios that do not require high consistency, distribute read requests to read replicas. If you have a high volume of queries, adding read replicas can help speed them up.

4.1.3 Index Design

- Use the same field type to prevent implicit conversion from causing invalid indexes.

- Create unique indexes on all minimum sets of fields or combinations of fields with uniqueness.

  For example, there is a table containing the fields **a**, **b**, **c**, **d**, **e**, and **f**. If the combinations of fields **ab** and **ef** have uniqueness, you are advised to create unique indexes for **ab** and **ef**, respectively.

  ◯ NOTE

  Even if complete verification control is implemented at the application layer, dirty data is generated as long as there is no unique index according to Murphy's Law.

  Before creating a unique index, consider whether it is helpful for queries. Useless indexes can be deleted.

  Evaluate the impact of extra indexes on INSERT operations. Determine whether to create unique indexes based on the requirements for the correctness and performance of data with uniqueness.

- Create indexes on fixed-length fields (for example, INT). When creating an index on a VARCHAR field, the index length must be specified. It is not necessary to create an index on the whole field. The index length is determined according to the actual text distinction.

  ◯ NOTE

  The index length and distinction are a pair of contradictions. Generally, for string type data, the distinction of an index with a length of 20 bytes will be higher than 90%. The distinction formula is COUNT(DISTINCT LEFT(Column_name, Index_length))/ COUNT(*). Place the column names with a high distinction on the left.

- If possible, do not use left fuzzy search (for example, **SELECT * FROM users WHERE u_name LIKE ' %hk'**) or full fuzzy search on the page to avoid degradation from index scan to full table scan. Solve the problem at the application layer.

  ◯ NOTE

  An index file has the leftmost prefix matching feature of B-tree. If the value on the left is not determined, the index cannot be used.

- Use a covering index to query data and avoid returning to the table. However, do not add too many fields to the covering index, or the write performance will be compromised.

  📖 **NOTE**

  > Types of indexes that can be created include primary key indexes, unique indexes, and normal indexes. A covering index indicates that if you execute EXPLAIN statements, "using index" will be displayed in the **Extra** column.

- Optimize the SQL performance as follows: range (minimum requirement), ref (basic requirement), and consts (maximum requirement).

- When creating a composite index, place the column with the highest distinction on the left.

- Ensure that the number of indexes in a single table is at most 5, or does not exceed 20% of the number of table fields.

- Avoid the following misunderstandings when creating indexes:

  – Indexes should be frequently used. An index needs to be created for a query.

  – Indexes should be as few as possible. Indexes consume space and slow down updates and insertions.

  – Unique indexes cannot be used. Unique features must be resolved at the application layer using the "query first and then insert" method.

- Reduce the use of ORDER BY that cannot be used with indexes based on the actual service requirements. The statements such as ORDER BY, GROUP BY, and DISTINCT consume many CPU resources.

- If a complex SQL statement is involved, use the existing index design and add EXPLAIN before the SQL statement. EXPLAIN can help you optimize the index by adding some query restrictions.

- Execute new SELECT, UPDATE, or DELETE statements with EXPLAIN to check the index usage and ensure no **Using filesort** and **Using temporary** are displayed in the **Extra** column. If the number of scanned rows exceeds 1,000, exercise caution when executing these statements. Analyze slow query logs and delete unused slow query statements every day.

> ☐ NOTE

EXPLAIN:

- **type**: ALL, index, range, ref, eq_ref, const, system, NULL (The performance is sorted from poor to good from left to right.)

- **possible_keys**: indicates the indexes from which MySQL can choose to find rows in this table. If there is an index on a field, the index is listed but may not be used by the query.

- **key**: indicates the key (index) that MySQL actually decided to use. If key is NULL, MySQL found no index to use for executing the query more efficiently. To force MySQL to use or ignore an index listed in the **possible_keys** column, use FORCE INDEX, USE INDEX, or IGNORE INDEX in your query.

- **ref**: shows which columns or constants are compared to the index named in the key column to select rows from the table.

- **rows**: indicates the estimated number of rows to be read for required records based on table statistics and index selection.

- **Extra**:
    - **Using temporary**: To resolve the query, MySQL needs to create a temporary table to hold the result. This typically happens if the query contains GROUP BY and ORDER BY clauses that list columns differently.

    - **Using filesort**: MySQL must do an extra pass to find out how to retrieve rows in sorted order.

    - **Using index**: The column information is retrieved from the table using only information in the index tree without having to do an additional seek to read the actual row. If **Using where** is displayed at the same time, it indicates that desired information needs to be obtained by using the index tree and reading rows of the table.

    - **Using where**: In WHERE clause, **Using where** is displayed when the desire data is obtained without reading all the data in the table or the desire data cannot be obtained by only using indexes. Unless you specifically intend to fetch or examine all rows from the table, you may have something wrong in your query if the **Extra** value is not **Using where** and the table join type is ALL or index.

- If a function is used on a WHERE statement, the index becomes invalid.

  For example, in **WHERE left(name, 5) = 'zhang'**, the left function invalidates the index on **name**.

  You can modify the condition on the service side and delete the function. When the returned result set is small, the service side filters the rows that meet the condition.

- For ultra-large tables, you also need to comply with the following rules when using indexes:

  - Create indexes for columns involved in the WHERE and ORDER BY statements. You can use EXPLAIN to check whether indexes or full table scans are used.

  - Fields with sparse value distribution, such as **gender** with only two or three values, cannot be indexed.

  - Do not use string fields as primary keys.

  - Do not use foreign keys. Programs can enforce the constraints.

  - When using multi-column indexes, arrange them in the same order as the query conditions and remove unnecessary single-column indexes (if any).

– Before removing an index, conduct a thorough analysis and back up the data.

4.1.4 SQL Usage

Database SQL Query

- Optimize the ORDER BY ... LIMIT statements by indexes to improve execution efficiency.

- If statements contain ORDER BY, GROUP BY, or DISTINCT, ensure that the result set filtered by the WHERE condition contains at most 1,000 lines. Otherwise, the SQL statements are executed slowly.

- For ORDER BY, GROUP BY, and DISTINCT statements, use indexes to directly retrieve sorted data. For example, use **key(a,b)** in **where a=1 order by b**.

- When using JOIN, use indexes on the same table in the WHERE condition.

  Example:

  select t1.a, t2.b from t1,t2 where t1.a=t2.a and t1.b=123 and t2.c= 4

  If the **t1.c** and **t2.c** fields have the same value, only **b** in the index **(b,c)** on **t1** is used.

  If you change **t2.c=4** in the WHERE condition to **t1.c=4**, you can use the complete index. This may occur during field redundancy design (denormalization).

- If deduplication is not required, use UNION ALL instead of UNION.

  As UNION ALL does not deduplicate and sort the data, it runs faster than UNION. If deduplication is not required, use UNION ALL preferentially.

- To implement pagination query in code, specify that if **count** is set to **0**, the subsequent pagination statements are not executed.

- Do not frequently execute COUNT on a table. It takes a long time to perform COUNT on a table with a large amount of data. Generally, the response speed is in seconds. If you need to frequently perform the COUNT operation on a table, introduce a special counting table.

- If only one record is returned, use LIMIT 1. If data is correct and the number of returned records in the result set can be determined, use LIMIT as soon as possible.

- When evaluating the efficiency of DELETE and UPDATE statements, change the statements to SELECT and then run EXPLAIN. A large number of SELECT statements will slow down the database, and write operations will lock tables.

- TRUNCATE TABLE is faster and uses fewer system and log resources than DELETE. If the table to be deleted does not have a trigger and the entire table needs to be deleted, TRUNCATE TABLE is recommended.

  – TRUNCATE TABLE does not write deleted data to log files.

  – A TRUNCATE TABLE statement has the same function as a DELETE statement without a WHERE clause.

  – TRUNCATE TABLE statements cannot be written with other DML statements in the same transaction.

- Do not use negative queries to avoid full table scanning. Negative queries indicate the following negative operators are used: NOT, !=, <>, NOT EXISTS, NOT IN, and NOT LIKE.

If a negative query is used, the index structure cannot be used for binary search. Instead, the entire table needs to be scanned.

- Avoid using JOIN to join more than three tables. The data types of the fields to be joined must be the same.

- During multi-table join query, ensure that the associated fields have indexes. When joining multiple tables, select the table with a smaller result set as the driving table to join other tables. Pay attention to table indexes and SQL performance even if two tables are joined.

- To query ultra-large tables, you also need to comply with the following rules:

  - To locate slow SQL statements, enable slow query logs.

  - Do not perform column operations, for example, **SELECT id WHERE age +1=10**. Any operation on a column, including database tutorial functions and calculation expressions, will cause table scans. Move operations to the right of the equal sign (=) during the query.

  - Split larger statements into smaller and simpler statements to reduce lock time and avoid blocking the entire database.

  - Do not use SELECT*.

  - Change OR to IN. The efficiency of OR is at the n level, while the efficiency of IN is at the log(n) level. Try to keep the number of INs below 200.

  - Avoid using stored procedures and triggers in applications.

  - Avoid using queries in the %xxx format.

  - Avoid using JOIN and try to query a single table whenever possible.

  - Use the same type for comparison, for example, **'123'** to **'123'** or **123** to **123**.

  - Avoid using the != or <> operators in the WHERE clause. Otherwise, the engine will not use indexes and instead scan the full table.

  - For consecutive values, use BETWEEN instead of IN: **SELECT id FROM t WHERE num BETWEEN1AND5**.

SQL Statement Development

- Split simple SQL statements.

  For example, in the OR condition **f_phone='10000' or f_mobile='10000'**, the two fields have their own indexes, but only one of them can be used.

  You can split the statement into two SQL statements or use UNION ALL.

- If possible, perform the complex SQL calculation or service logic at the service layer.

- Use a proper pagination method to improve pagination efficiency. Skipping paging is not recommended for large pages.

  - Negative example: **SELECT * FROM table1 ORDER BY ftime DESC LIMIT 10000,10;**

    It causes a large number of I/O operations because MySQL uses the read-ahead policy.

  - Positive example: **SELECT * FROM table1 WHERE ftime < last_time ORDER BY ftime DESC LIMIT 10;**

Recommended pagination method: Transfer the threshold value the last pagination.

- Execute UPDATE statements in transactions based on primary keys or unique keys. Otherwise, a gap lock is generated and the locked data range is expanded. As a result, the system performance deteriorates and a deadlock occurs.

- Do not use foreign keys and cascade operations. The problems of foreign keys can be solved at the application layer.

  Example:

  If **student_id** is a primary key in the student table, **student_id** is a foreign key in the score table. If **student_id** is updated in the student table, **student_id** in the score table is also updated. This is a cascade update.

  – Foreign keys and cascade updates are suitable for single-node clusters with low concurrency and are not suitable for distributed cluster with high concurrency.

  – Cascade updates may cause strong blocks and foreign keys affect the INSERT operations.

- If possible, do not use IN. If it is required, ensure that the number of set elements after IN should be at most 500.

- To reduce interactions with the database, use batches of SQL statements, for example, **INSERT INTO ... VALUES (*),(*),(*)....(*);**. Try to keep the number of ***** items below 100.

- Do not use stored procedures, which are difficult to debug, extend, and transplant.

- Do not use triggers, event schedulers, or views for service logic. The service logic must be processed at the service layer to avoid logical dependency on the database.

- Do not use implicit type conversion.

  ### NOTE

  The conversion rules are as follows:

  1. If at least one of the two parameters is NULL, the comparison result is also NULL. However, when <=> is used to compare two NULL values, 1 is returned.

  2. If both parameters are character strings, they are compared as character strings.

  3. If both parameters are integers, they are compared as integers.

  4. When one parameter is a hexadecimal value and the other parameter is a non-digit value, they are compared as binary strings.

  5. If one parameter is a TIMESTAMP or DATETIME value and the other parameter is a CONSTANT value, they are compared as TIMESTAMP values.

  6. If one parameter is a DECIMAL value and other parameter is a DECIMAL or INTEGER value, they are compared as DECIMAL values. If the other argument is a FLOATING POINT value, they are compared as FLOATING POINT values.

  7. In other cases, both parameters are compared as FLOATING POINT values.

  8. If one parameter is a string and the other parameter is an INT value, they are compared as FLOATING POINT values (by referring to item 7)

     For example, the type of **f_phone** is varchar. If **f_phone in (098890)** is used in the WHERE condition, two parameters are compared as FLOATING POINT values. In this case, the index cannot be used, affecting database performance.

     If **f_user_id = '1234567'**, the number is directly compared as a character string. For details, see item 2.

- If possible, ensure that the number of SQL statements in a transaction should be as small as possible, no more than 5. Long transactions will lock data for a long time, generate many caches in MySQL, and occupy many connections.

- Do not use NATURAL JOIN.

  NATURAL JOIN is used to implicitly join column, which is difficult to understand and may cause problems. The NATURAL JOIN statement cannot be transplanted.

- For tables with tens of millions or hundreds of millions of data records, you are advised to use the following methods to improve data write efficiency:

  a. Delete unnecessary indexes.

     When data is updated, the index data is also updated. For tables with large amounts of data, avoid creating too many indexes as this can slow down the update process. Delete unnecessary indexes.

  b. Insert multiple data records in batches.

     This is because batch insertion only requires a single remote request to the database.

     Example:

     ```
     insert into tb1 values(1,'value1');
     insert into tb2 values(2,'value2');
     insert into tb3 values(3,'value3');
     ```

     After optimization:

     ```
     insert into tb values(1,'value1'),(2,'value2'),(3,'value3');
     ```

  c. When inserting multiple data records, manually control transactions.

     By manually controlling the transaction, multiple execution units can be merged into a single transaction, avoiding the overhead of multiple transactions while ensuring data integrity and consistency.

     Example:

     ```
     insert into table1 values(1,'value1'),(2,'value2'),(3,'value3');
     insert into table2 values(4,'value1'),(5,'value2'),(6,'value3');
     insert into table3 values(7,'value1'),(8,'value2'),(9,'value3');
     ```

     After optimization:

     ```
     start transaction;
     insert into table1 values(1,'value1'),(2,'value2'),(3,'value3');
     insert into table2 values(4,'value1'),(5,'value2'),(6,'value3');
     insert into table3 values(7,'value1'),(8,'value2'),(9,'value3');
     commit;
     ```

     ⚠️ **CAUTION**

     Having too many merged statements can lead to large transactions, which will lock the table for a long time. Evaluate service needs and control the number of statements in a transaction accordingly.

  d. When inserting data with primary keys, try to insert them in a sequential order of the primary keys. You can use AUTO_INCREMENT.

     Inserting data in a random order of the primary keys can cause page splitting, which can negatively impact performance.

     Example:

     Inserting data in a random order of primary keys: 6 2 9 7 2

Inserting data in a sequential order of primary keys: 1 2 4 6 8

 e. Avoid using UUIDs or other natural keys, such as ID card numbers, as primary keys.

   UUIDs generated each time are unordered, and inserting them as primary keys can cause page splitting, which can negatively impact performance.

 f. Avoid modifying primary keys during service operations.

   Modifying primary keys requires modifying the index structure, which can be costly.

 g. Reduce the length of primary keys as much as possible.

 h. Do not use foreign keys to maintain foreign key relationships. Use programs instead.

 i. Separate read and write operations. Direct read requests to read replicas to avoid slow insertion caused by I/Os.

4.2 Database Management

4.2.1 Creating a Database

Scenarios

After a GaussDB(for MySQL) instance is created, you can create databases on it.

Constraints

- This operation is not allowed when another operation is being performed on your DB instance.
- After a database is created, the database name cannot be changed.

Method 1: Creating a Database on the GaussDB(for MySQL) Console

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

Step 2 Click 📍 in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Databases**.

Step 6 In the displayed dialog box, set the required parameters and click **OK**.

Figure 4-1 Creating a database

Table 4-2 Parameter description

Parameter	Description
Database Name	The database name can consist of 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The total number of hyphens (-) cannot exceed 10.
Character Set	Select a character set as required.
User	• You can select one or more unauthorized users. If there are no unauthorized users, you can **create one**. • If you require fine-grained permissions control, log in to the DAS console.
Remarks	The remarks can consist of up to 512 characters. It cannot contain carriage returns or any of the following special characters: !<"='>&

Step 7 After the database is created, authorize or delete it on the **Databases** page. You can search for the desired database by character set and database name.

----End

Method 2: Creating a Database Through DAS
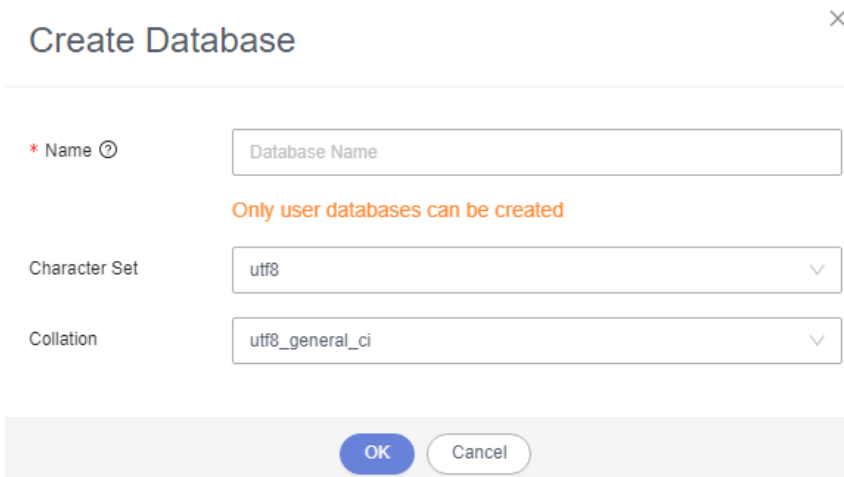
Step 1 **Log in to the management console.**

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate an instance and click **Log In** in the **Operation** column.

Step 5 On the displayed DAS login page, enter the username and password and click **Log In**.

Step 6 Create a database using either of the following methods:

- On the home page, click **Create Database**. In the displayed dialog box, set the database name, character set, and collation, and click **OK**.

Figure 4-2 Creating a database

- Choose **SQL Operations** > **SQL Query**. In the displayed SQL window, select the target database and run the following command:
  create database *database_name*;

----End

APIs

- **Creating a Database**
- **Querying Databases**
- **Querying Available Database Character Sets**
- **Modifying Database Remarks**

4.2.2 Deleting a Database

Scenarios

You can delete databases you have created.

Constraints

- Deleted databases cannot be recovered. Exercise caution when deleting a database.
- This operation is not allowed when another operation is being performed on your DB instance.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Databases**.

Step 6 On the displayed page, locate a database and click **Delete** in the **Operation** column.

Step 7 In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

Figure 4-3 Deleting a database
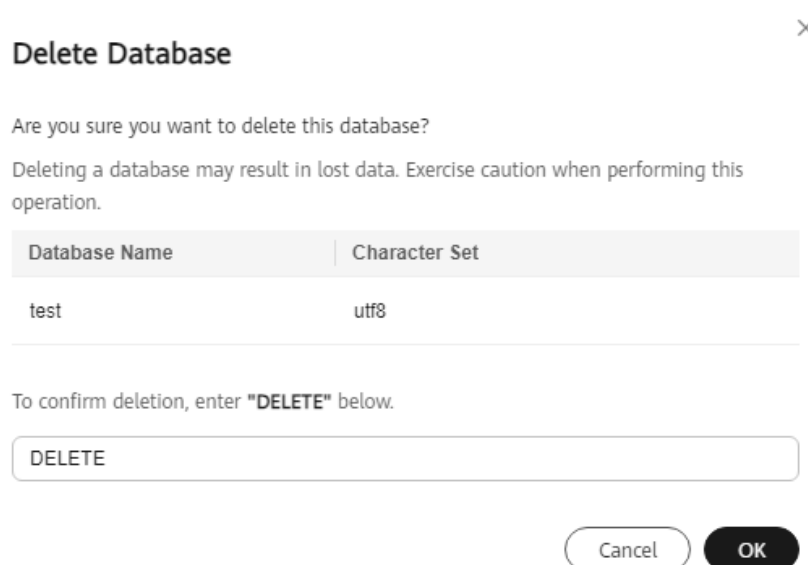
Step 8 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

----End

APIs

- **Deleting a Database**
- **Creating a Database**

4.2.3 Enabling or Disabling Event Scheduler

You can enable or disable event scheduler on the GaussDB(for MySQL) console. Read **Disclaimer** carefully before using it.

Disclaimer

Normal product functions on Huawei Cloud can meet the daily needs of most customers. For trigger-related functions, you are advised to implement them on the business program side. If you do need to enable event scheduler, be aware of the following issues due to known community risks:

● The actual time for triggering the event scheduler is inconsistent with the configured time.

● The event scheduler is not triggered.

● Due to the particularity of the event scheduler, the actual execution may be different from what you expected.

● The event scheduler may impact analysis and judgment for issues with database usage.

● Heterogeneous disaster recovery cannot be used.

● Other unknown issues.

If any of these issues occur, your workloads may be affected.

Constraints

When the instance is being rebooted or its specifications are being changed, event scheduler cannot be enabled or disabled.

Enabling Event Scheduler

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Instance Information** area, click ⬤ under **Event Scheduler**.

Figure 4-4 Enabling event scheduler
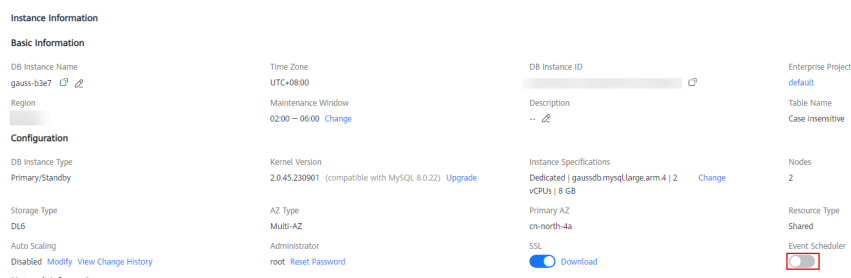
Step 6 In the displayed dialog box, read the disclaimer and click **Agree and Continue**.

Figure 4-5 Reading the disclaimer
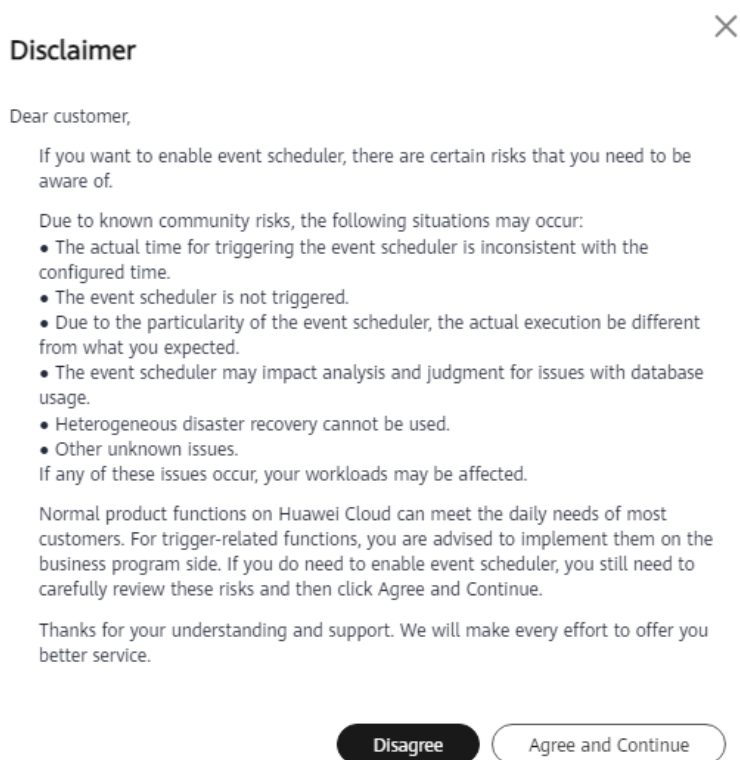
Step 7 In the displayed dialog box, confirm the instance information and click **OK**.

Figure 4-6 Confirming information
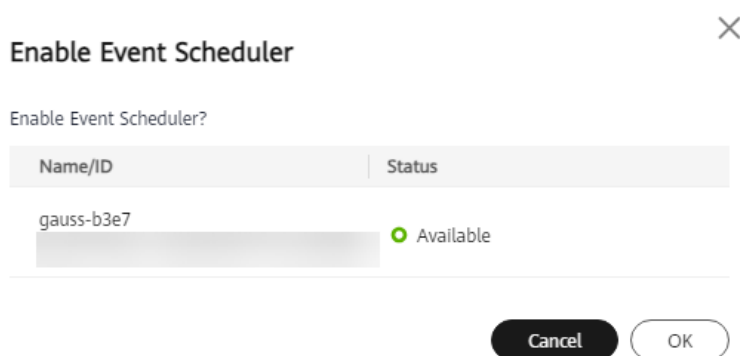
----End

Disabling Event Scheduler

Step 1 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 2 In the **Instance Information** area, click ⬤ under **Event Scheduler**.

Figure 4-7 Disabling event scheduler
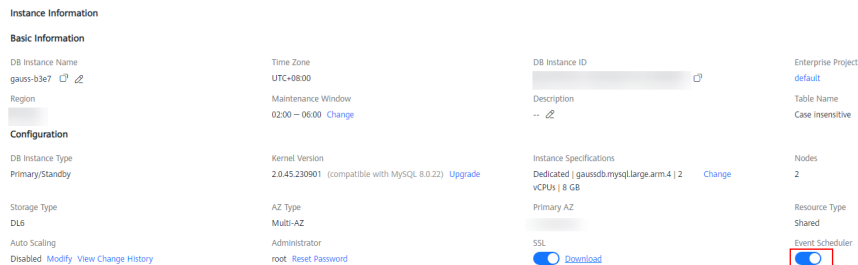
Step 3 In the displayed dialog box, click **OK**.

Figure 4-8 Confirming information
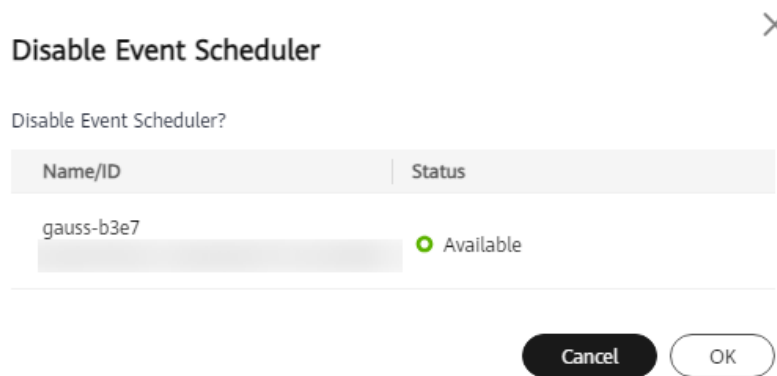
----End

4.3 Account Management (Non-Administrator)

4.3.1 Creating an Account

Scenarios

When you create a GaussDB(for MySQL) instance, account **root** is created by default. You can create other accounts as needed.

Constraints

This operation is not allowed when another operation is being performed on your DB instance.

Method 1: Creating an Account on the GaussDB(for MySQL) Console

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Accounts**.

Step 6 On the displayed page, click **Create Account**.

Step 7 In the displayed dialog box, set the required parameters.

Figure 4-9 Creating an account

Table 4-3 Parameter description

Parameter	Description
Username	The username can consist of 1 to 32 characters. Only letters, digits, and underscores (_) are allowed.
Host IP Address	• To enable all IP addresses to access your DB instance, set it to **%**. • To enable all IP addresses in the subnet 10.10.10.* to access your DB instance, set it to **10.10.10.%**. • To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.*,172.16.213.*** (no spaces before or after the comma).

Parameter	Description
Database	You can select one or more unauthorized databases and authorize their permissions to the account. If there are no unauthorized databases, you can **create ones**. You can also **modify the database permissions** after the account is created. **NOTE** ● If you do not delete a database on the GaussDB(for MySQL) console but delete a database in other ways, permissions granted specifically for the database are not automatically deleted. They must be deleted manually. This is an open-source MySQL behavior. For details, see **DROP DATABASE Statement**. ● If you require fine-grained permissions control, log in to the **DAS** console.
Password	The password must: ● Consist of 8 to 32 characters. ● Contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^&*-_=+?,()&|.). ● Comply with the values of **validate_password** parameters. To check the password-related parameter values, click an instance name, choose **Parameters** in the navigation pane, and search for **validate_password** in the upper right corner of the page. **Figure 4-10** Checking the password-related parameters ● Be different from the username or the username spelled backwards.
Confirm Password	The value must be the same as that of **Password**.
Remarks	The remarks can consist of up to 512 characters. It cannot contain carriage returns or any of the following special characters: !<"='>&

Step 8 Click **OK**.

Step 9 After the account is created, you can manage it on the **Accounts** page of the selected instance.

----End

Method 2: Creating an Account Through DAS

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.
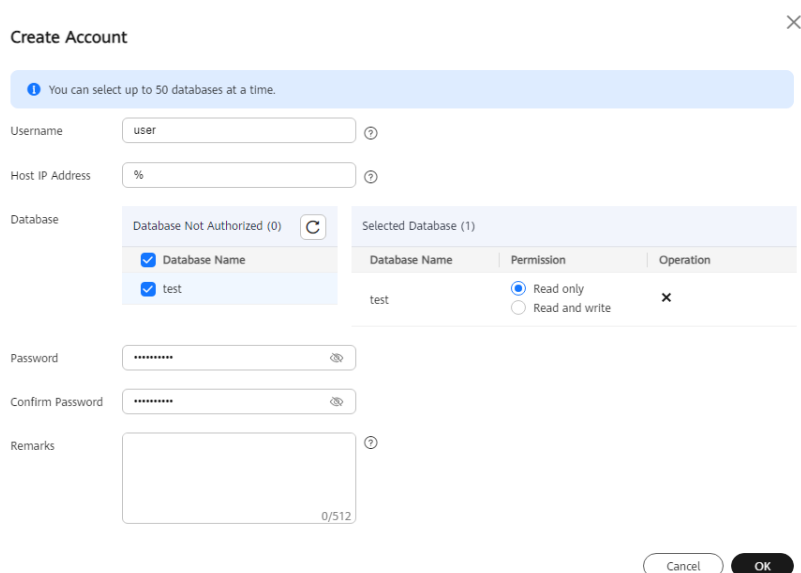
Step 4 On the **Instances** page, locate an instance and click **Log In** in the **Operation** column.

Step 5 On the displayed DAS login page, enter the username and password and click **Log In**.

Step 6 Create an account using either of the following methods:

- Choose **SQL Operations** > **SQL Query**. In the displayed SQL window, select the target database and run the following command:

  create user *username*;

- Choose **Account Management** > **User Management** and click **Create User**.

  For detailed operations and parameter settings, see **Creating a User**.

----End

APIs

- **Creating a Database Account**
- **Querying Database Users**
- **Modifying Remarks of a Database User**

4.3.2 Resetting a Password for an Account

Scenarios

You can reset passwords for the accounts you have created. To protect your DB instance against brute force cracking, change your password periodically, such as every three or six months.

Constraints

- This operation is not allowed when another operation is being performed on your DB instance.

- After the password is reset, the DB instance will not be rebooted and your permissions will not be changed.

- You can query password reset records on the CTS console. For details, see *Cloud Trace Service User Guide*.

Procedure

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

Step 2 Click 📍 in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Accounts**. On the displayed page, locate the target account and click **Reset Password** in the **Operation** column.

Step 6 In the displayed dialog box, enter a new password, confirm it, and click **OK**.

The password must meet the following requirements:

- It must consist of 8 to 32 characters.

- It must contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&|.).

- It must comply with the values of **validate_password** parameters.

  To check the password-related parameter values, click an instance name, choose **Parameters** in the navigation pane, and search for **validate_password** in the upper right corner of the page.

  Figure 4-11 Checking the password-related parameters

Parameter Name ↓↑	Effective upon Reboot ↓↑	Value	Allowed Values	Description
validate_password.check_user_name	No	ON	ON, OFF	Check whether the password is the same as the username or usernam...
validate_password.length	No	8	0–1,024	Controls the minimum number of characters in a password. Constraint...
validate_password.mixed_case_count	No	1	0–256	Controls the minimum number of letters in a password when validate_...
validate_password.number_count	No	1	0–256	Controls the minimum number of digits in a password when validate_p...
validate_password.policy	No	LOW	LOW, MEDIUM, STRONG	Value: LOW: The value of validate_password.length parameter is applie...
validate_password.special_char_count	No	1	0–256	Controls the minimum number of special characters in a password wh...

- The password you entered in the **Confirm Password** text box must be the same as that you entered in the **New Password** text box.

- It cannot be the username or the username spelled backwards.

Step 7 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

----End

APIs

- **Changing Password of a Database User**
- **Querying Database Users**

4.3.3 Changing Permissions for Accounts

Scenarios

You can authorize custom database users to specified databases and revoke permissions for authorized databases.

Constraints

This operation is not allowed when another operation is being performed on your DB instance.

Procedure
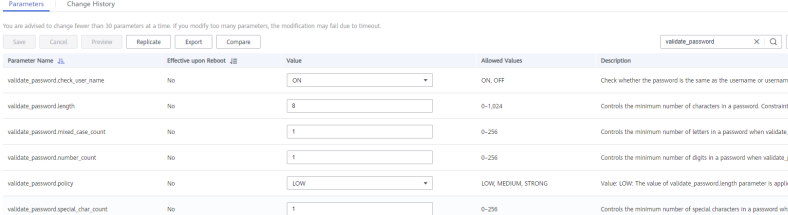
Step 1 **Log in to the management console.**

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Accounts**. On the displayed page, locate the target account and choose **More** > **Change Permission** in the **Operation** column.

Step 6 In the displayed dialog box, change account permissions.

Select one or more unauthorized databases and authorize their permissions to the account. To delete a selected database, locate the database and click **×** in the **Operation** column.

Figure 4-12 Changing permissions

----**End**

APIs

- **Authorizing Permissions to a Database User**
- **Deleting Permissions of a Database User**

4.3.4 Deleting an Account

Scenarios

You can delete accounts you have created.

Constraints

- Deleted accounts cannot be restored. Exercise caution when deleting an account.
- This operation is not allowed when another operation is being performed on your DB instance.

Procedure

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

Step 2 Click ⦿ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Accounts**.

Step 6 On the displayed page, locate an account and click **Delete** in the **Operation** column.

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

Step 8 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

----End

APIs

Deleting a Database User

5 Data Migration

5.1 Data Migration Schemes

You can migrate data from RDS for MySQL, self-managed MySQL, other cloud MySQL, and self-managed Oracle databases to GaussDB(for MySQL), or from one GaussDB(for MySQL) instance to another GaussDB(for MySQL) instance.

Migration Tools

Table 5-1 Migration tools

Tool	Description	Billing	Reference
DRS (recommended)	Data Replication Service (DRS) provides real-time data migration and synchronization between databases in various scenarios. It is easy to use and can complete a migration task in minutes. DRS facilitates data transfer between databases, helping you reduce DBA labor costs and hardware costs.	Pay-per-use For details, see **Billing**.	**What Is DRS?**
mysqldump	mysqldump is a command-line tool that comes with MySQL. It is used to back up and restore MySQL databases.	Free of charge	**What Is mysqldump?**

Tool	Description	Billing	Reference
DAS	During data backup or migration, Data Admin Service (DAS) can help you export data to a local PC or OBS bucket, and import the data to the target data table.	Free of charge	**What Is DAS?**

Migration Schemes

Table 5-2 Migration schemes

Source Database	Migration Tool	Characteristic	Operation Guide
RDS for MySQL	DRS	- Migration of all data, database-level data, or table-level data - Full and incremental data migration - Minimal downtime - Applicable to any data volume	**From MySQL to GaussDB(for MySQL)**
GaussDB(for MySQL)	mysqldump	- Full data migration - Long downtime - Applicable to small amounts of data	**Migrating Data to GaussDB(for MySQL) Using mysqldump**
	DAS	- Full data migration - Long downtime - Applicable to moderate amounts of data	**Migrating Data to GaussDB(for MySQL) Using the Export and Import Functions of DAS**
- On-premises MySQL databases - ECS-hosted MySQL databases	DRS	- Migration of all data, database-level data, or table-level data - Full and incremental data migration - Minimal downtime - Applicable to any data volume	**From ECS-hosted MySQL to GaussDB(for MySQL)**

Source Database	Migration Tool	Characteristic	Operation Guide
Other cloud MySQL databases	DRS	• Migration of all data, database-level data, or table-level data • Full and incremental data migration • Minimal downtime • Applicable to any data volume	**From Other Cloud MySQL to GaussDB(for MySQL)**

5.2 Migrating Data to GaussDB(for MySQL) Using mysqldump

You can use mysqldump to migrate data to GaussDB(for MySQL).

Precautions

Database migration is performed offline. Before the migration, you must stop any applications using the source database.

Preparing for the Migration

1. Prepare an ECS in the same VPC and subnet as the GaussDB(for MySQL) instance or bind an EIP to the GaussDB(for MySQL) instance.

   – To connect to an instance through a private network, an ECS has to be created first.

     Purchase an ECS and **log in to the ECS**.

   – To connect to an instance through an EIP, you must:

     i. Bind the EIP to the instance. For details, see **Procedure**.

     ii. Ensure that the local device can access the EIP that has been bound to the instance.

2. Install the MySQL client on the prepared ECS or device that can access the GaussDB(for MySQL) instance.

   📖 NOTE

   • Install the MySQL client by following the instructions provided in **How Can I Install the MySQL Client?**

   • Ensure that the MySQL client version is the same as or later than that installed on the GaussDB(for MySQL) instance. The MySQL database or client provides mysqldump and mysql by default.

Exporting Data

Before migrating data from the source database to the GaussDB(for MySQL) instance, you need to export data from the source database first.

ⵌ **NOTE**

> mysqldump must match the DB engine version.

Step 1 Log in to the prepared ECS or device that can access the GaussDB(for MySQL) instance.

Step 2 Use mysqldump to export metadata into an SQL file.

NOTICE

MySQL databases are required for GaussDB(for MySQL) management. When exporting metadata, do not specify **--all-database**, or the databases will be unavailable.

---

mysqldump --databases *<DB_NAME>* **--single-transaction --order-by-primary --hex-blob --no-data --routines --events --set-gtid-purged=OFF** -u *<DB_USER>* **-p -h** *<DB_ADDRESS>* **-P** *<DB_PORT>* **|sed -e 's/DEFINER[ ]*=[ ]*[^*]**/*/' -e 's/ DEFINER[ ]*=.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*=.*PROCEDURE/ PROCEDURE/' -e 's/DEFINER[ ]*=.*TRIGGER/TRIGGER/' -e 's/ DEFINER[ ]*=.*EVENT/EVENT/' >** *<BACKUP_FILE>*

- *<DB_NAME>* indicates the name of the database to be migrated.
- *<DB_USER>* indicates the database username.
- *<DB_ADDRESS>* indicates the database address.
- *<DB_PORT>* indicates the database port.
- *<BACKUP_FILE>* indicates the name of the file to which the data will be exported.

Enter the database password when prompted.

Example:

mysqldump --databases gaussdb --single-transaction --order-by-primary -- hex-blob --no-data --routines --events --set-gtid-purged=OFF -u root -p -h 192.*.*.* -P 3306 |sed -e 's/DEFINER[ ]*=[ ]*[^*]**/*/' -e 's/ DEFINER[ ]*=.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*=.*PROCEDURE/ PROCEDURE/' -e 's/DEFINER[ ]*=.*TRIGGER/TRIGGER/' -e 's/ DEFINER[ ]*=.*EVENT/EVENT/' > dump-defs.sql

Enter password:

After this command is executed, the **dump-defs.sql** file will be generated.

Step 3 Use mysqldump to export data into an SQL file.

NOTICE

MySQL databases are required for GaussDB(for MySQL) management. When exporting metadata, do not specify **--all-database**, or the databases will be unavailable.

---

mysqldump --databases *<DB_NAME>* **--single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u** *<DB_USER>* **-p -h** *<DB_ADDRESS>* **-P** *<DB_PORT>* **-r** *<BACKUP_FILE>*

For details on the parameters in the preceding command, see **Step 2**.

Enter the database password when prompted.

Example:

mysqldump --databases gaussdb --single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u root -p -h 192.*.*.* -P 3306 -r dump-data.sql

After this command is executed, the **dump-data.sql** file will be generated.

----End

Importing Data

You can use a client to connect to the GaussDB(for MySQL) instance through an ECS or device that can access the GaussDB(for MySQL) instance and then import the exported SQL files into that instance.

> **NOTICE**
>
> If the source database calls triggers, stored procedures, functions, or events, you must set **log_bin_trust_function_creators** to **ON** for the destination database before importing data.

Step 1 Import metadata into the GaussDB(for MySQL) instance.

mysql -f -h *<DB_ADDRESS>* **-P** *<DB_PORT>* **-u** root **-p <** *<BACKUP_DIR>***/dump-defs.sql**

- *<DB_ADDRESS>* indicates the IP address of the GaussDB(for MySQL) instance.
- *<DB_PORT>* indicates the port of the GaussDB(for MySQL) instance.
- *<BACKUP_DIR>* indicates the directory where **dump-defs.sql** will be stored.

Example:

mysql -f -h 172.*.*.* -P 3306 -u root -p < dump-defs.sql

Enter password:

Step 2 Import data into the GaussDB(for MySQL) instance.

mysql -f -h *<DB_ADDRESS>* **-P** *<DB_PORT>* **-u** root **-p <** *<BACKUP_DIR>***/dump-data.sql**

- *<DB_ADDRESS>* indicates the IP address of the GaussDB(for MySQL) instance.
- *<DB_PORT>* indicates the port of the GaussDB(for MySQL) instance.
- *<BACKUP_DIR>* indicates the directory where **dump-data.sql** will be stored.

Example:

mysql -f -h 172.*.*.* -P 3306 -u root -p < dump-data.sql

Enter password:

Step 3   Use the MySQL tool to connect to the GaussDB(for MySQL) instance and view the results.

mysql> show databases;

In this example, the database named **my_db** has been imported.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| my_db              |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)
```

----End

5.3 Migrating Data to GaussDB(for MySQL) Using the Export and Import Functions of DAS

Scenarios

Data Admin Service (DAS) is a one-stop management platform that allows you to manage Huawei Cloud databases on a web console. It offers database development, O&M, and intelligent diagnosis, making it easy to use and maintain databases.

During data backup or migration, Data Admin Service (DAS) can help you export data to a local PC or OBS bucket, and import the data to the target data table. DAS allows you to export an entire database, some data tables, or SQL result sets.

Constraints

- Only one file that is no larger than 1 GB can be imported at a time.
- Only data files in .sql, .csv, or .xlsx format can be imported.
- If data files are exported as a .zip package, they cannot be directly imported. You need to extract the files first.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB cannot be imported.
- Data cannot be exported from or imported to cross-region OBS buckets.

Exporting Data

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

Step 2   Click   in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate an instance and click **Log In** in the **Operation** column.

Step 5 On the displayed DAS login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.
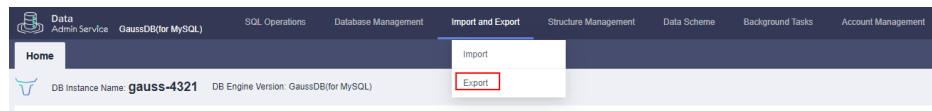
Step 7 Export **an entire database**, **some data tables**, or **SQL result sets**.

Step 8 Export an entire database.

Method 1: Use the quick export function.

1. Click **Quick Export** and select the database to be exported.
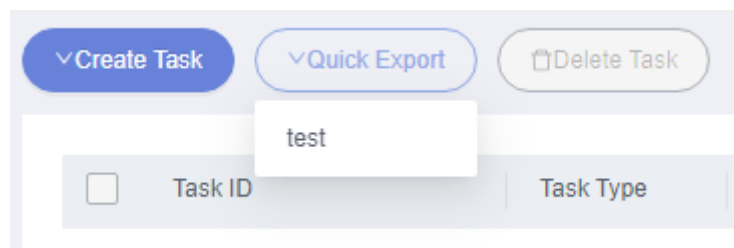
2. In the displayed dialog box, select a storage path and click **OK**.

☐ **NOTE**

– DAS does not store any data. The exported data files are stored in the OBS bucket that you have created.

– Creating an OBS bucket is free, but you will be billed for storing data in the bucket.
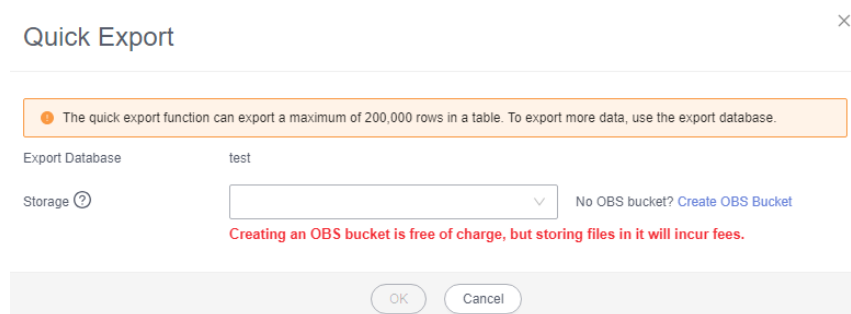
Method 2: Create an export task.

1. Click **Create Task** > **Export Database**.

2. In the displayed dialog box, configure task information.
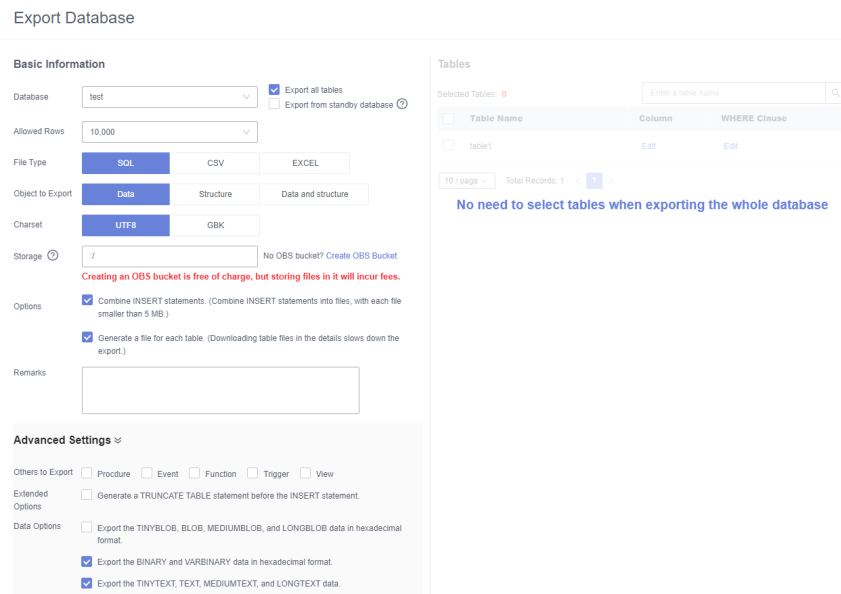
Table 5-3 Parameter description

Category	Parameter	Description
Basic Information	Database	Select the database to be exported and select **Export all tables**. – You can also select **Export from standby database** as required. If this option is selected, DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest. – Databases are classified into user databases and system databases. System databases cannot be exported. If system database data is required, deploy system database services in a user database, so that you can export the system database data from the user database.
	Allowed Rows	Select the maximum number of rows in a single table.
	File Type	Select **SQL**, **CSV**, or **EXCEL**.
	Object to Export	Select **Data**, **Structure**, or **Data and structure**.
	Charset	Select **UTF8** or **GBK**.
	Storage	Select an OBS bucket for storing data files.

Categor y	Paramete r	Description
	Options	– **Combine INSERT statements.** If you select this option, INSERT statements will be combined into files, with each file no greater than 5 MB. – **Generate a file for each table.** If you do not select this option, all data files will be exported as a .zip package, which cannot be directly imported. You need to extract the files first. If you select this option, the data file (in .sql, .csv, or .xlsx format) of each table will be exported and can be directly imported again.
	Remarks	-
Advance d Settings	You can configure advanced options as required.	

3. Click **OK**.

Step 9 Export some data tables.

1. Click **Create Task** > **Export Database**.

2. In the displayed dialog box, configure task information.
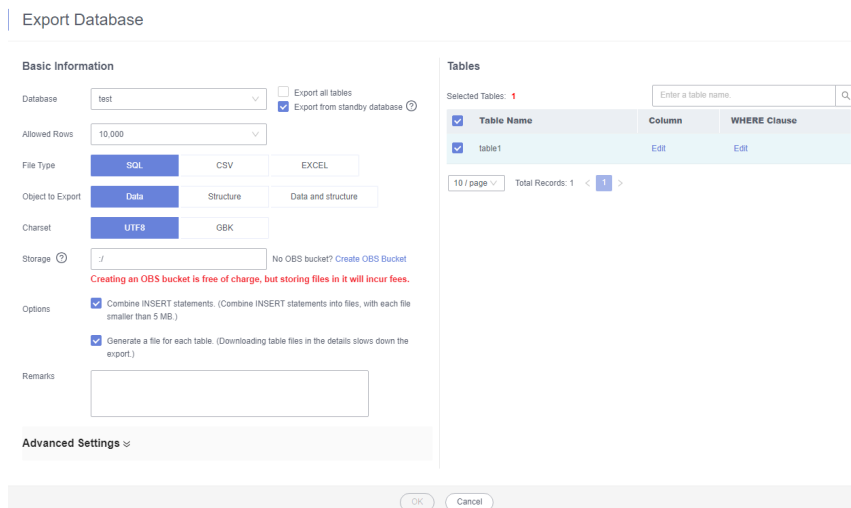
Table 5-4 Parameter description

Category	Parameter	Description
Basic Information	Database	Select the database to be exported and select the tables to be exported in the **Tables** area on the right. You can also select **Export from standby database** as required. If this option is selected, DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.
	Allowed Rows	Select the maximum number of rows in a single table.
	File Type	Select **SQL**, **CSV**, or **EXCEL**.
	Object to Export	Select **Data**, **Structure**, or **Data and structure**.
	Charset	Select **UTF8** or **GBK**.
	Storage	Select an OBS bucket for storing data files.
	Options	– **Combine INSERT statements.** If you select this option, INSERT statements will be combined into files, with each file no greater than 5 MB. – **Generate a file for each table.** If you do not select this option, all data files will be exported as a .zip package, which cannot be directly imported. You need to extract the files first. If you select this option, the data file (in .sql, .csv, or .xlsx format) of each table will be exported and can be directly imported again.
	Remarks	-
Advanced Settings		You can configure advanced options as required.

3. Click **OK**.

Step 10 Export SQL result sets.

1. Click **Create Task** > **Export SQL Result**.
2. In the displayed dialog box, configure task information.

Export SQL Result   ×

Basic Information

Database	test	☐ Export from standby database ⑦
Allowed Rows	10,000	
File Type	**SQL-Insert**	CSV	EXCEL
Charset	**UTF8**	GBK
Storage ⑦	:/	No OBS bucket? Create Bucket

Creating an OBS bucket is free of charge, but storing files in it will incur fees.

Options	☑ Combine INSERT statements. (Combine INSERT statements into files, with each file smaller than 5 MB.)
	☐ Generate one file for each result.
SQL to Execute	SELECT * FROM
Remarks	

Advanced Settings ⌄

OK   Cancel

Table 5-5 Parameter description

Categor y	Paramete r	Description
Basic Informat ion	Database	Select the database to be exported. You can also select **Export from standby database** as required. If this option is selected, DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.
	Allowed Rows	Select the maximum number of rows in a single table.
	File Type	Select **SQL-Insert**, **CSV**, or **EXCEL**.
	Charset	Select **UTF8** or **GBK**.
	Storage	Select an OBS bucket for storing data files.

Categor y	Paramete r	Description
	Options	– **Combine INSERT statements.** If you select this option, INSERT statements will be combined into files, with each file no greater than 5 MB. – **Generate one file for each result.** If you do not select this option, all data files will be exported as a .zip package, which cannot be directly imported. You need to extract the files first. If you select this option, the data file (in .sql, .csv, or .xlsx format) of each result set will be exported and can be directly imported again.
	SQL to Execute	Enter an SQL statement. To export multiple SQL result sets at a time, enter multiple SQL statements, with each on a separate line and ending with a semicolon (;). After the export task is complete, SQL files are generated. One SQL statement corresponds to one file.
	Remarks	-
Advance d Settings		You can configure advanced options as required.

3. Click **OK**.

----End

Importing Data

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate an instance and click **Log In** in the **Operation** column.

Step 5 On the displayed DAS login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **Import and Export** > **Import**.

Step 7 Click **Create Task**.

Step 8 In the displayed dialog box, configure task information.

Table 5-6 Parameter description

Parameter	Description
Import Type	Set this parameter based on the type of an exported file. Currently, only SQL and CVS files are supported.
File Source	Import a file from your local PC or an OBS bucket. ● **Upload file** If you select **Upload file** for **File Source**, you need to set **Attachment Storage** and upload the required file. To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading. Creating an OBS bucket is free, but you will be billed for storing data in the bucket. ● **Choose from OBS** If you select **Choose from OBS** for **File Source**, you need to select a file from the bucket. The file uploaded from an OBS bucket will not be deleted upon an import success.
Database	Select the destination database.
Charset	Select **UTF8**, **GBK**, or **Auto Detect**.

Parameter	Description
Options	• **Ignore errors, skip the step when the SQL statement fails to be executed.** If you select this option, the system will skip any errors detected when SQL statements are being executed. • **Delete the uploaded file upon an import success.** If you select this option, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database. This option is only available to the files uploaded from your local PC.
Remarks	-

Step 9 Click **Create**.

Step 10 In the displayed dialog box, confirm the information and click **OK**.

> ⚠️ An import task will be created for you. The import task may overwrite your original data. Please confirm and click OK to continue.
>
> Target database: **test**
>
> [OK]   [Cancel]

Step 11 After the data is imported successfully, log in to the destination database to query the imported data.

----End

6 Instance Management

6.1 Viewing the Overall Status of DB Instances

The **Overview** page gives you a bird's eye view of GaussDB(for MySQL) instances, including instances by status, alarms, and intelligent diagnosis.

Functions

Table 6-1 lists the functions of the **Overview** page.

Table 6-1 Function description

Function	Description	Related Operation
Instances by Status	Shows the number of instances in different states.	For details, see **Instances by Status**.
Alarms	Shows the active alarms of all instances, including alarms in the **Alarm (metric)** and **Triggered (event)** states.	For details, see **Alarms**.
Intelligent Diagnosis	Diagnoses instance health using operational data analytics and intelligent algorithms.	For details, see **Intelligent Diagnosis**.

Instances by Status

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

Step 2 Click ⊚ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, click **Overview**.

Step 5 In the **Instances by Status** area, check the status of all GaussDB(for MySQL) instances under the current account.
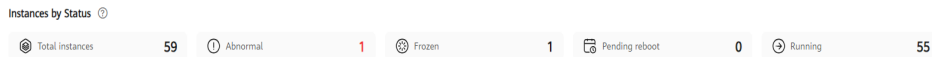
Figure 6-1 Checking instances by status

Table 6-2 Status description

Status	Description	Handling Suggestion
Total instances	Total number of GaussDB(for MySQL) instances in all states	-
Abnormal	Total number of GaussDB(for MySQL) instances in the **Abnormal** state	Contact customer service.
Frozen	Total number of GaussDB(for MySQL) instances in the **Frozen** state	See **Resource Freezing, Unfreezing, Release, Deletion, and Unsubscription**.
Pending reboot	Total number of GaussDB(for MySQL) instances in the **Pending reboot** state **NOTE** Modifications to some parameters require an instance reboot before they can be applied.	Reboot instances.
Available	Total number of GaussDB(for MySQL) instances in the **Available** state	-

----End

Alarms

Based on the configured alarm rules, you can view active alarms of all GaussDB(for MySQL) instances under the current account, including alarms in the **Alarm** (metric) and **Triggered** (event) states.

1. In the upper right corner of the **Alarms** area, click **Create Alarm Rule** to access the Cloud Eye console.
   - By default, the system has a built-in alarm rule, which can be modified, disabled, and deleted. For details, see **Modifying an Alarm Rule**.
   - Click **Create Alarm Rule** to create an alarm rule to monitor a metric or event for instances. For details, see **Creating an Alarm Rule**.

2. In the upper right corner of the **Alarms** area, select a time window and view alarm details.
   - The time window can be **Last 1 hour**, **Last 6 hours**, **Last 12 hours**, **Last day**, **Last week**, or **Last month**.

- The **Alarm Severity** area displays the total number of alarms and the number of alarms of each severity. Alarm severities include **Critical**, **Major**, **Minor**, and **Warning**.

- The **Top 5 Instances by Total Number of Alarms** area displays alarm statistics of the top 5 instances with the largest number of alarms. You can hover over an instance to view the number of alarms of each severity.

- For details about critical alarms, see **Table 6-3**.

**Table 6-3** Critical alarm description

Parameter	Description
Instance Name	Name of the instance where the alarm was reported. After the page is refreshed, the latest alarm details will be displayed in real time.
Status	You can view active alarms of all instances in the current region, including alarms in the **Alarm** (metric) and **Triggered** (event) states. ● **Alarm**: The metric value reached the alarm threshold, and an alarm has been triggered but not cleared for the resource. ● **Triggered**: An event configured in the alarm policy triggered an alarm.
Alarm Type	Alarm type to which the alarm rule applies. ● Alarm (metric) ● Triggered (event)
Alarm Policy	Policy for triggering an alarm. ● If you set **Alarm Type** to **Metric**, whether to trigger an alarm depends on whether the data in consecutive periods reaches the threshold. For example, Cloud Eye triggers an alarm if the average CPU usage of the monitored object is 80% or more for three consecutive 5-minute periods. For handling suggestions for high CPU usage, see **What Should I Do If the CPU Usage of My GaussDB(for MySQL) Instance Is High?** ● If you set **Alarm Type** to **Event**, the event that triggers the alarm is an instant operation. For example, if an instance fails to be created, an alarm is triggered. For details about supported events and handling suggestions for exceptions, see **Events Supported by Event Monitoring**.
Alarm Rule	Name or ID of the alarm rule
Last Updated	Latest time when the alarm was triggered

Parameter	Description
Operation	Click **Metrics**. In the displayed dialog box, check the metric monitoring views in the selected time window.

Intelligent Diagnosis

Intelligent Diagnosis checks instance health using operational data analytics and intelligent algorithms and provides diagnosis results and suggestions.

Figure 6-2 Health diagnosis
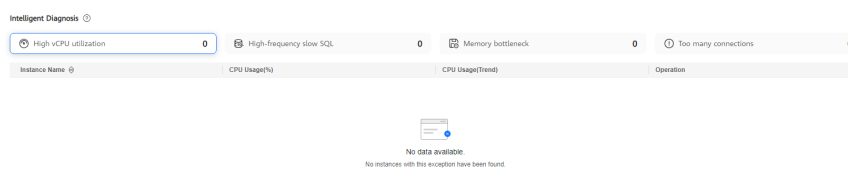
Click an abnormal diagnosis item to view the abnormal instances and related metric data.

For example, if the vCPU utilization is high, you can click **High vCPU utilization** to view the abnormal instances, CPU usage, and CPU usage trend. You can also click **Diagnosis Details** in the **Operation** column to view the detailed diagnosis results.

For details about supported diagnosis items and their handling suggestions, see **Table 6-4**.

Table 6-4 Intelligent diagnosis details

Diagnosis Item	Metric	Metric Description	Handling Suggestion	Reference
High vCPU utilization	CPU Usage	CPU usage of the monitored object	• Evaluate the SQL execution plan and add indexes to avoid full table scanning. • Upgrade vCPUs for compute-intensive workloads.	**What Should I Do If the CPU Usage of My GaussDB(for MySQL) Instance Is High?**

Diagn osis Item	Metric	Metric Description	Handling Suggestion	Reference
Memor y bottlen eck	Memory Usage	Memory usage of the monitored object	• Upgrade instance specificati ons. • Optimize SQL statement s to reduce the use of temporary tables. • Reconnect sessions at a specific interval to release memory of the sessions.	**How Do I Handle a Large Number of Temporary Tables Being Generated for Long Transaction s and High Memory Usage?**
High-freque ncy slow SQL	Slow Query Logs(Count/ min)	Number of GaussDB(for MySQL) slow query logs generated per minute	• Optimize slow SQL statement s based on the execution plan. • Upgrade vCPUs.	**How Do I Handle Slow SQL Statements Caused by Inappropria te Composite Index Settings?**

Diagn osis Item	Metric	Metric Description	Handling Suggestion	Reference
Too many connec tions	Total Connections(Count)	Total number of connections that connect to the GaussDB(for MySQL) server	• Check whether applicatio ns are connected , optimize the connectio ns, and release unnecessa ry connectio ns. • Check the instance specificati ons and upgrade them if needed.	**What Do I Do If the Number of GaussDB(fo r MySQL) Database Connections Reaches the Upper Limit?**
	Current Active Connections(Count)	Number of active connections		
	Connection Usage	Percent of used GaussDB(for MySQL) connections to the total number of connections		

6.2 Viewing Metrics

The **Metrics** page allows you to monitor GaussDB(for MySQL) instances.

- You can view the real-time performance metrics and trends of all instances in your account. This allows you to quickly identify and address any abnormal instances.

- You can also view the historical performance metrics.

Viewing Real-Time Metrics

1. **Log in to the management console.**

2. Click ⊙ in the upper left corner and select a region and project.

3. Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

4. In the navigation pane, choose **Metrics**.

5. View the real-time performance metrics of a created GaussDB(for MySQL) instance under the current account.

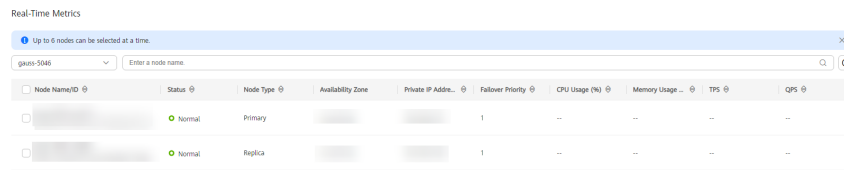**Figure 6-3** Viewing real-time metrics

Table 6-5 Parameter description

| Parameter	Description
Node Name/ID	Only monitoring data for the nodes of a created GaussDB(for MySQL) instance is displayed.

Status	The value can be:
	• **Normal**: Real-time monitoring data is displayed.
	  NOTE
	  The monitoring data and charts are available for a new instance after the instance runs for about 10 minutes.
	• **Abnormal**: There is no monitoring data. The default values for all metrics are 0. The monitoring data is available only after the instance becomes normal.
	• **Stopped**: There is no monitoring data. The default values for all metrics are 0. The monitoring data is available only after the instance is started.

Node Type	The value can be: • Primary • Replica
Availability Zone	AZ where a node is located
Private IP Address	Private IP address of a node
Failover Priority	Failover priority of a node
Metrics	For details about metric description and handling suggestions for abnormal metrics, see **Table 6-6**. The following metrics are available: • CPU Usage • Memory Usage • TPS • QPS

Table 6-6 Monitoring items

| Item	Description	Handling Suggestion	Reference
CPU Usage	CPU usage of the monitored object	• Evaluate the SQL execution plan and add indexes to avoid full table scanning. • Upgrade vCPUs for compute-intensive workloads.	**What Should I Do If the CPU Usage of My GaussDB(for MySQL) Instance Is High?**
Memory Usage	Memory usage of the monitored object	• Upgrade instance specifications. • Optimize SQL statements to reduce the use of temporary tables. • Reconnect sessions at a specific interval to release memory of the sessions.	**How Do I Handle a Large Number of Temporary Tables Being Generated for Long Transactions and High Memory Usage?**
TPS	Execution times of submitted and rollback transactions per second	• Evaluate the SQL execution plan and add indexes to avoid full table scanning. • Upgrade vCPUs for compute-intensive workloads.	**What Should I Do If the CPU Usage of My GaussDB(for MySQL) Instance Is High?**
QPS	Query times of SQL statements (including stored procedures) per second		

Viewing Historical Metrics

Select one or more nodes in the real-time metric list and then view their historical metrics in the **Historical Metrics** area.

Figure 6-4 Viewing historical metrics
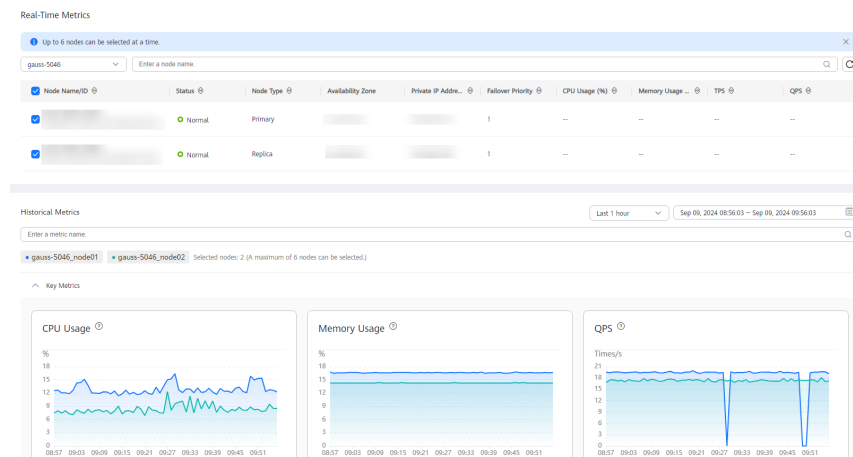
- You can view the metrics of up to six nodes at a time.

- You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 24 hours, or last 7 days. You can also configure a time period.

- You can move the cursor to a point in time of a chart to view the performance metric at that point in time.

Figure 6-5 Viewing a performance metric at a point in time
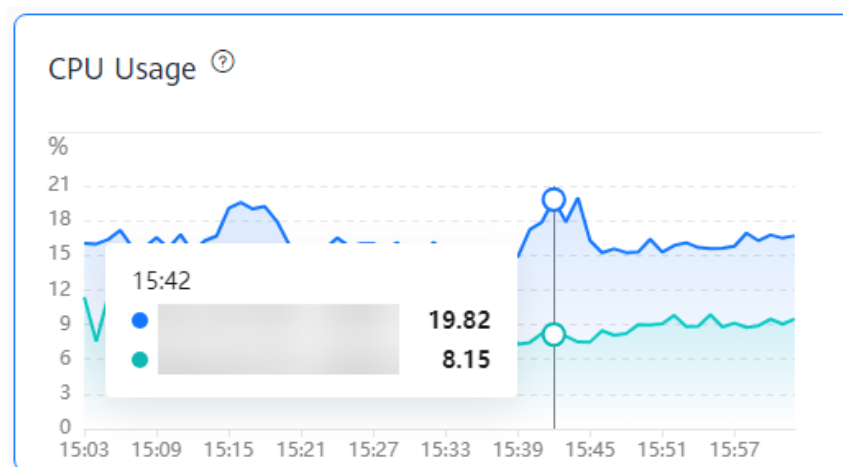
6.3 Instance Lifecycle Management

6.3.1 Changing a DB Instance or Node Name

Scenarios

You can change the name of a GaussDB(for MySQL) instance or its node for easy identification.

Changing a DB Instance Name

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

Step 2 Click [icon] in the upper left corner and select a region and project.

Step 3 Click [icon] in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.
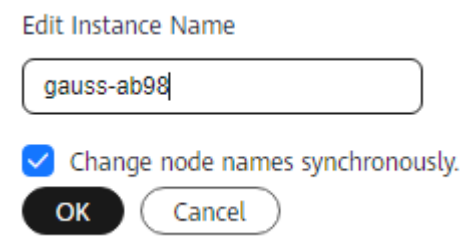
Step 4 On the **Instances** page, locate a DB instance and click [icon] in the **Name/ID** column to edit the DB instance name.

Figure 6-6 Changing a DB instance name on the **Instances** page

Edit Instance Name

gauss-ab98

☑ Change node names synchronously.
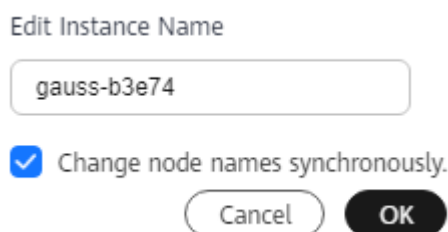
OK   Cancel

Alternatively, click the instance name to go to the **Basic Information** page. Locate **DB Instance Name** in the **Instance Information** area, and click [icon] to edit the instance name.

Figure 6-7 Changing a DB instance name on the **Basic Information** page

Edit Instance Name

gauss-b3e74

☑ Change node names synchronously.

Cancel   OK

- The instance name must start with a letter and consist of 4 to 64 characters. Only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_) are allowed.

- When changing the instance name, you can determine whether to select **Change node names synchronously** as required. If this option is selected, the names of the corresponding nodes are changed when the instance name is changed. If this option is not selected, only the instance name is changed, and the corresponding node names are not changed.

- If you want to submit the change, click **OK**. If you want to cancel the change, click **Cancel**.

Step 5 Check that the instance name has been changed. It takes less than 1 minute to change a DB instance name.

----End

Changing a Node Name

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, select one or more nodes, click **Change Node Name**.

Figure 6-8 Changing node names
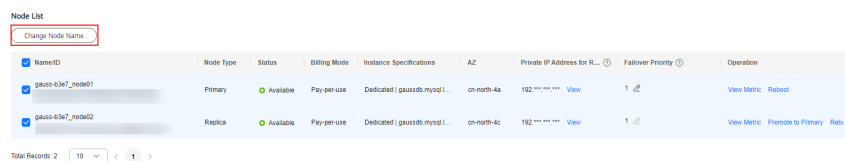
Alternatively, click ✏ next to a node name to edit the node name.

- The node name must start with a letter and consist of 4 to 128 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
- The node name must be unique.

Step 6 Click **OK** to submit the change.

Step 7 Check that the node name has been changed.

----End

APIs

- **Changing a DB Instance Name**
- **Querying DB Instances**
- **Querying Details of a DB Instance**
- **Querying Details of DB Instances in Batches**

6.3.2 Modifying a DB Instance Description

Scenarios

After a GaussDB(for MySQL) instance is created, you can add a description for it.

Procedure

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

Step 2   Click ⦿ in the upper left corner and select a region and project.

Step 3   Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4   On the **Instances** page, locate an instance and click ✐ in the **Description** column to edit the instance description.

- The instance description can contain up to 64 characters, and cannot start with or end with a space. Only letters, digits, hyphens (-), underscores (_), periods (.), and spaces are allowed.

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

Alternatively, click the instance name to go to the **Basic Information** page. Locate

Description in the **Instance Information** area, and click ✐ to edit the instance description.

----End

APIs

- **Changing a DB Instance Description**
- **Querying DB Instances**
- **Querying Details of a DB Instance**
- **Querying Details of DB Instances in Batches**

6.3.3 Rebooting a DB Instance or Node

Scenarios

You may need to reboot a GaussDB(for MySQL) instance or node for maintenance purposes. For example, after modifying some parameters, you may need to reboot your instance to apply the modifications. You may need to reboot a node to resolve database connection issues.

Rebooting a DB Instance

> **NOTICE**
>
> - If the DB instance status is **Abnormal**, the reboot may fail.
> - To shorten the time required, reduce database activities during the reboot to reduce rollback of transit transactions.
> - Rebooting a DB instance will interrupt services briefly. During this period, the instance status is **Rebooting**.
> - A DB instance will be unavailable when it is being rebooted. Rebooting a DB instance will clear the cached memory in it. To prevent traffic congestion during peak hours, you are advised to reboot the DB instance during off-peak hours.

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to reboot and choose **More** > **Reboot** in the **Operation** column.

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

The read replicas are also rebooted.

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 obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 6 In the **Reboot DB Instance** dialog box, set **Scheduled Time**.

Figure 6-9 Rebooting a DB instance

Table 6-7 Rebooting a DB instance

Parameter	Description
Scheduled Time	You can reboot a DB instance immediately or during the maintenance window. ● **Immediate**: The DB instance will be rebooted immediately. ● **During maintenance window**: The DB instance will be rebooted during a maintenance window. The maintenance window is 02:00–06:00 by default and you can change it as required. Changing the maintenance window will not affect the timing that has already been scheduled. A reboot task configured during a current maintenance window will not be executed until the next maintenance window.

Step 7 Click **OK**.

Step 8 View the reboot progress on the **Task Center** page. If the task status becomes **Completed** and the instance status becomes **Available**, the DB instance has been rebooted successfully.

----End

Rebooting a Node

> **NOTICE**
>
> - Nodes in the **Abnormal** state can be rebooted.
> - To shorten the time required, reduce database activities during the reboot to reduce rollback of transit transactions.
> - Rebooting a node will interrupt services briefly. During this period, the node status is **Rebooting node**.
> - A node will be unavailable when it is being rebooted. You are advised to reboot the node during off-peak hours and ensure that your applications support automatic reconnection.
> - If a parameter of your DB instance is modified, you need to first reboot the DB instance to apply the modification, and then reboot a node of the DB instance.

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, select the target node and click **Reboot** in the **Operation** column.

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

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 7 In the displayed dialog box, set **Scheduled Time**.

Table 6-8 Rebooting a node

Parameter	Description
Scheduled Time	You can reboot a node immediately or during the maintenance window. ● **Immediate**: The node will be rebooted immediately. ● **During maintenance window**: The node will be rebooted during a maintenance window. The maintenance window is 02:00–06:00 by default and you can change it as required. Changing the maintenance window will not affect the timing that has already been scheduled. A reboot task configured during a current maintenance window will not be executed until the next maintenance window.

Step 8 Click **Yes**.

Step 9 View the reboot progress on the **Task Center** page. If the task status becomes **Completed** and the node status becomes **Available**, the node has been rebooted successfully.

----End

APIs

● **Rebooting a DB Instance**
● **Rebooting a Node**

6.3.4 Exporting DB Instance Information

Scenarios

You can export DB instance information in the instance list for further analysis.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click **Export Instance Info** above the instance list. In the displayed dialog box, select the items to be exported and click **OK**.

Step 5 Check the .csv file locally after the export task is complete.

----End

6.3.5 Deleting a DB Instance

Scenarios

You can manually delete a DB instance billed on a pay-per-use or serverless basis on the **Instances** page.

Constraints

- Instances cannot be deleted when operations are being performed on them.
- If you delete a DB instance, its automated backups are also deleted and you are no longer billed for them. Manual backups are still retained and will incur additional costs.
- If you delete a DB instance, its read replicas are also deleted.
- If a backup of a DB instance is being used to restore data, the DB instance cannot be deleted.
- Deleted DB instances cannot be recovered and their resources will be released immediately. To retain data, back up the data first and then delete the DB instances.
- Deleted DB instances will be moved to the recycle bin, but will be permanently deleted after a length of time determined by the recycling policy.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to delete and click **More** > **Delete** in the **Operation** column.

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 obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 6 In the displayed dialog box, enter **DELETE** and click **OK**. Refresh the **Instances** page later to check that the deletion is successful.

----End

APIs

- **Deleting a DB Instance**
- **Deleting a Read Replica**
- **Querying DB Instances**

6.3.6 Rebuilding a DB Instance in the Recycle Bin

You can rebuild unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances in the recycle bin.

The recycle bin is enabled by default and cannot be disabled.

Modifying the Recycling Policy

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

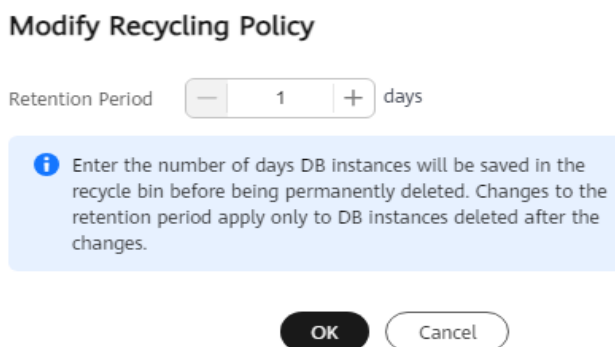**Step 2** Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set **Retention Period** (value range: 1 to 7, in days).

> 📖 **NOTE**
>
> The new recycling policy applies only to DB instances deleted after the changes.

Figure 6-10 Modifying the recycling policy

Modify Recycling Policy

Retention Period   [—   1   +] days

ⓘ Enter the number of days DB instances will be saved in the recycle bin before being permanently deleted. Changes to the retention period apply only to DB instances deleted after the changes.

(OK)   (Cancel)

Step 5 Click **OK**.

----End

Rebuilding a DB Instance

You can rebuild DB instances in the recycle bin within the retention period.

Step 1 On the **Recycle Bin** page, locate the DB instance you want to rebuild and click **Rebuild** in the **Operation** column.

Step 2 On the displayed **Rebuild DB Instance** page, set required parameters by referring to section "Buying a DB Instance".

Step 3 Click **Next**.

**Step 4** Confirm the information and click **Submit**.

----End

6.3.7 Introducing Heterogeneous DR Instances

GaussDB(for MySQL) supports heterogeneous DR instances to ensure high availability. If your DB instance is unavailable due to unknown community bugs, misoperations, or AZ faults, the heterogeneous DR instance can take over the services.

The DR instance creation, maintenance, and switchover are all completed by Huawei Cloud and you do not need to do anything.

The free trial of heterogeneous DR instances has ended on January 1, 2024.

Figure 6-11 Heterogeneous DR instance
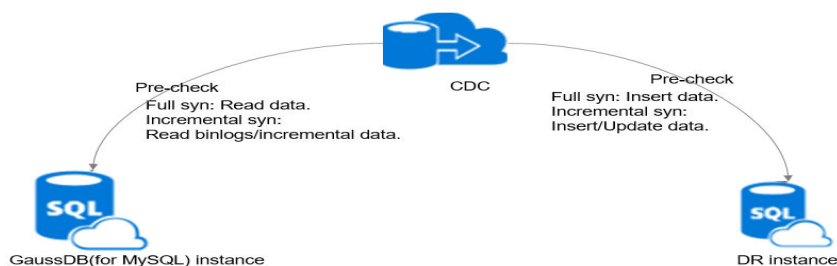
Constraints

Heterogeneous DR instances cannot be created in the scenarios listed in the following table.

Table 6-9 Constraints

Scenario	Reason
Binlog is not enabled.	Binlogs are used to synchronize data between the GaussDB(for MySQL) instance and DR instance.
binlog_format is not set to **ROW**.	The DR link may be disconnected.
There are triggers.	Data may be inconsistent between the GaussDB(for MySQL) instance and the DR instance.
There are events.	Data may be inconsistent between the GaussDB(for MySQL) instance and the DR instance.

Scenario	Reason
Database names, table names, field names, or indexes contain special characters.	The DR link may be disconnected.
The data volume exceeds 4 TB.	The capacity of the DR instance has an upper limit.
The database access is restricted by rules in the security group.	The DR instance cannot connect to the GaussDB(for MySQL) instance to synchronize data.
The GaussDB(for MySQL) instance is frozen.	The DR instance cannot be created.
There are no sufficient IP addresses in the VPC subnet.	The DR instance and GaussDB(for MySQL) instance need to use IP addresses in the same subnet.
There are tables without primary keys or indexes and a large number of operations.	The DR link may be disconnected.
Cascade operations are performed on tables with foreign keys.	Data may be inconsistent between the GaussDB(for MySQL) instance and the DR instance.

6.4 Configuration Changes

6.4.1 Changing the vCPUs or Memory of a DB Instance or Node

Scenarios

You can change the the specifications (vCPUs and memory) of a yearly/monthly or pay-per-use DB instance or node as needed. If the status of a DB instance or node changes from **Changing instance specifications** to **Available**, the change is successful.

You can scale up or down the specifications of your DB instance or node.

Constraints

- A DB instance or node cannot be deleted when its specifications are being changed.
- You can change the specifications of a DB instance or just a single node within the instance. To change the specifications of a single node, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

- Instance specifications can only be changed from the general-purpose edition to the dedicated edition.

- You can change the specifications of yearly/monthly or pay-per-use DB instances immediately or during a maintenance window. Serverless DB instances do not support specification changes.

- If you want to change instance specifications during a maintenance window, you can cancel the task before it starts. Once started, the task cannot be canceled.

- During an instance specification change, a read replica will be promoted to primary. To prevent service interruptions, perform the operation during off-peak hours.

- The time required for modifying specifications depends on factors such as the number of nodes, database load, and number of database tables.

- Changing instance specifications will change the private IP addresses for read of the primary node and read replicas. The connection addresses in your application need to be changed to prevent your services from being affected. You are advised to use the private IP address of a DB instance to connect your application.

- The specifications of the primary node and read replicas can be changed separately. When the specifications of the primary node are changed separately, the specifications of **synchronous nodes** are also changed.

Procedure

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

Step 2 Click ⬙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 Change specifications.

- **Changing the specifications of a DB instance**

  On the **Instances** page, locate the DB instance whose specifications you want to change and choose **More** > **Change Instance Specifications** in the **Operation** column.

  Alternatively, click the instance name to go to the **Basic Information** page. In the **Configuration** area, click **Change** under **Instance Specifications**.

- **Changing the specifications of the primary node**

  On the **Instances** page, click the instance name to go to the **Basic Information** page. In the **Node List** area, locate the primary node and click **Change** in the **Operation** column.

- **Changing the specifications of a read replica**

  On the **Instances** page, click the instance name to go to the **Basic Information** page. In the **Node List** area, locate a read replica and click **Change** in the **Operation** column.

Step 5 On the displayed page, select the desired specifications. You can scale up or down the specifications as required.

You can change the specifications immediately or during the maintenance window.

- **Upon submission**: The specifications will be changed immediately after the task is submitted.
- **In maintenance window**: The specifications will be changed during the maintenance window you specify.

Step 6 Click **Next**. On the displayed page, confirm the specifications.

- If you need to modify your settings, click **Previous** to go back to the page where you specify details.
- For pay-per-use instances, click **Submit**.

  To view the cost incurred by the instance specifications change, choose **Billing Center** > **Billing Dashboard** in the upper right corner.
- For yearly/monthly instances:
  - Scaling down the specifications: click **Submit**.

    The refund is automatically returned to your account. You can click **Billing Center** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - Scaling up the specifications: click **Submit**. The scaling starts only after the payment is successful.

Step 7 View the results.

Changing the instance specifications takes 5–15 minutes. During this period, the status of the instance on the **Instances** page is **Changing instance specifications**. After a few minutes, you can click the instance name to view the new instance specifications on the displayed **Basic Information** page.

> **NOTICE**
>
> - After the instance specifications of GaussDB(for MySQL) 8.0 are changed, the system will change the values of the following parameters accordingly: **innodb_buffer_pool_size**, **innodb_log_buffer_size**, **max_connections**, **innodb_buffer_pool_instances**, **innodb_page_cleaners**, **innodb_parallel_read_threads**, **innodb_read_io_threads**, **innodb_write_io_threads**, and **threadpool_size**.
> - The default value of **innodb_parallel_select_count** is **OFF** for instance with 16 vCPUs or less and **ON** for instances with more than 16 vCPUs.
>
>   If you have modified value of the parameter, the parameter value remains unchanged after the specifications are changed, or the default value is used.

----End

APIs

- **Changing DB Instance Specifications**
- **Promoting a Read Replica to Primary**
- **Querying Database Specifications**

- **Querying Details of a DB Instance**

6.4.2 Changing the Storage Space of a DB Instance

Scenarios

If the original storage space of your yearly/monthly DB instance is insufficient or redundant as your services change, you can scale up or down the storage.

Constraints

- The storage of pay-per-use DB instances grows as needed, so you cannot manually scale up their storage. The storage of pay-per-use DB instances is not limited.

- When you purchase a yearly/monthly DB instance, you need to select storage for it as needed. If your purchased storage cannot meet service requirements, the system will automatically scale up the storage as needed and you will be billed on a pay-per-use basis for additional storage. If services requirements decrease later, the system preferentially scales down the storage that was automatically scaled up.

  For example, you purchased 10 GB of storage when purchasing a DB instance. Later, as services increased, GaussDB(for MySQL) automatically scaled the storage to 18 GB as needed and you would be billed on a pay-per-use basis for the additional 8 GB of storage. Then, you manually scaled up the storage to 20 GB, which could meet service requirements. The 8 GB of storage that was scaled up by the system will be scaled down. You would only pay for the 20 GB of storage at yearly/monthly rates.

- The system changes storage of your DB instance as your services change, but you can change storage only by a multiple of 10 GB.

- During a storage change, services including backup service are not interrupted.

- You can change the storage space of a DB instance numerous times.

- If the storage space of a DB instance is being changed, you cannot reboot or delete the DB instance.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance and choose **More** > **Change Storage** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Storage and Backup** area, locate **Storage Space** and click **Change Storage**.

Figure 6-12 Scaling storage space
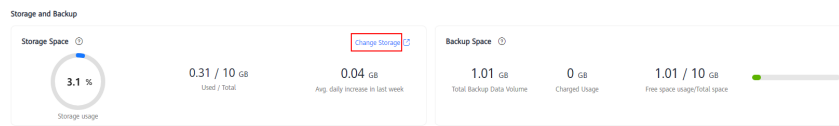
Step 5 Select the new storage space and click **Next**.

Storage space can be scaled up to 128,000 GB only by a multiple of 10 GB. Price after scaling is displayed in the lower left corner of the page.

Storage space can be scaled down to 40 GB only by a multiple of 10 GB. Refund price is displayed in the lower left corner of the page.

□ **NOTE**

To reduce the storage of a DB instance to 10 GB, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Step 6 Confirm your settings.

● If you need to modify your settings, click **Previous**.

● If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.

Step 7 View the new storage.

On the **Instances** page, click the instance name to go to the **Basic Information** page and view the new storage.

----End

APIs

● **Scaling up Storage of a Yearly/Monthly DB Instance**

● **Querying Database Specifications**

● **Querying Details of a DB Instance**

6.4.3 Configuring Auto Scaling for a DB Instance

Scenarios

You can configure auto scaling policies for your pay-per-use and yearly/monthly DB instances on the **Basic Information** page. When configuring auto scaling policies, you can enable or disable **Auto Scale-up** or **Auto Scale-down**. The scaling types include changing instance specifications and the number of read replicas.

Constraints

● This function is only available for pay-per-use and yearly/monthly DB instances.

● To set **Scaling Type** to **Number of read replicas** for a yearly/monthly DB instance, submit an application by choosing **Service Tickets > Create Service**

Ticket in the upper right corner of the management console. The read replicas that are automatically added or deleted will be billed based on a pay-per-use basis.

● To configure auto scaling policies, you must have the iam:agencies:listAgencies permission. If you do not have this permission, **create a custom policy**.

● Changing DB instance specifications will briefly interrupt services.

● If you want to set **Scaling Type** to **Number of read replicas**, there must be only one proxy instance. For details, see **How to Use a Proxy Instance to Enable Read/Write Splitting**.

● The system will delete or add read replicas. To prevent your services from being affected, you are advised not to use an IP address for read to connect to your applications.

● The pricing standard for auto scaling is the same as that for manual scaling. For details, see **Billing**.

Billing

● Pay-per-Use Instances

The instance specifications and number of read replicas can be automatically changed.

Pricing is listed on a per-hour basis, but bills are calculated down to the second. The old order automatically becomes invalid.

To view the cost incurred by auto scaling, choose **Billing Center** > **Billing Dashboard** in the upper right corner of the management console.

● Yearly/Monthly Instances

The instance specifications and number of read replicas can be automatically changed.

You will be billed for the new specifications. For details, see **GaussDB(for MySQL) Pricing Details**.

If the new specifications are less than the specifications that you purchased, the refund is automatically returned to your account. You can click **Billing Center** in the upper right corner of the management console and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.

Table 6-10 Pricing description for yearly/monthly instances

Billing Item	Description
Specifications after scale-up	You need to pay the following fee:
	Price of new specifications x Remaining duration x Number of nodes – Price of old specifications x Remaining duration x Number of nodes
	Note: Remaining duration = Number of remaining days in a calendar month/Total number of days in the calendar month
	Example:
	A customer placed and paid a monthly order for a GaussDB(for MySQL) instance on April 1, 2023. The instance contains 2 nodes and its specifications are 2 vCPUs and 8 GB of memory. The total subscription period would be 30 days and the instance would expire on April 30, 2023. On April 18, 2023, the instance specifications were automatically expanded to 4 vCPUs and 16 GB memory. The remaining duration is 0.4 (12/30).
	The monthly price of the new specifications is $290 USD and that of the old specifications is $145 USD, so you need to pay for $116 USD (290 x 0.4 x 2 – 145 x 0.4 x 2).
Specifications after scale-down	For pricing details, see **Unsubscriptions**.
Added read replicas	New read replicas are billed based on the actual usage duration.
Deleted read replicas	Deleted read replicas are no longer billed.

Modifying Auto Scaling Policies

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

Step 2 Click ⦿ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Configuration** area, click **Modify** under **Auto Scaling**.

Figure 6-13 Modifying auto scaling policies
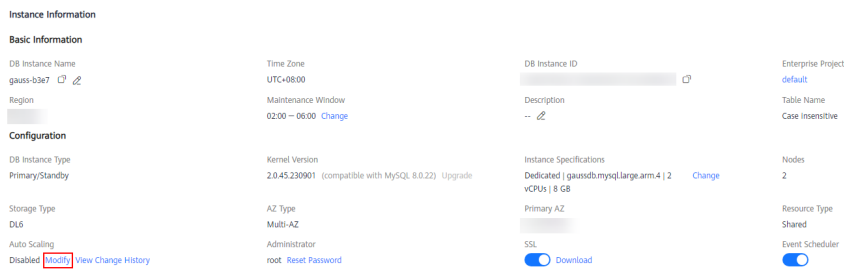
Step 6 In the displayed dialog box, set the required parameters.
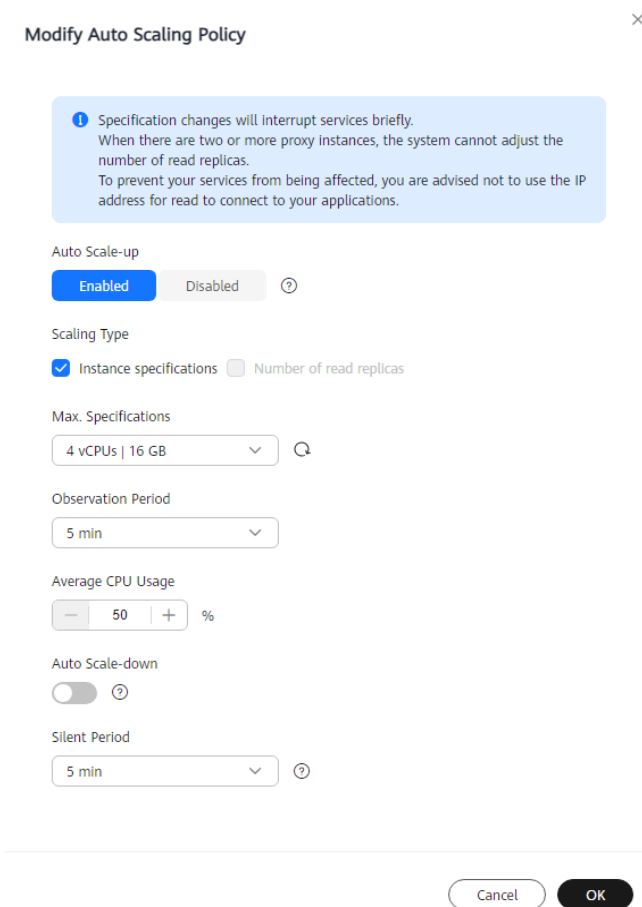
Figure 6-14 Modifying the auto scaling policy

Table 6-11 Parameter configuration

Parameter	Parameter description
Auto Scale-up	You can enable or disable it as needed.

Parameter	Parameter description
Scaling Type	● Instance specifications ● Number of read replicas **NOTE** 　● You can select one or more scaling types. 　● The read replicas that are automatically added or deleted will be billed based on a pay-per-use basis. 　● If you deselect **Number of read replicas** for **Scaling Type**, pay-per-use nodes created in the current instance will be automatically deleted. Exercise caution when performing this operation. 　● The account balance must be sufficient, or scaling up the specifications or adding read replicas may fail. 　● After **Auto Scale-up** is enabled, read replicas that are automatically added cannot be promoted to primary.
Observation Period	● Once auto scale-up is enabled, if the system observes any increases in the average CPU usage over the preset value, it upgrades the specifications or adds read replicas based on the read and write traffic. The system enters a silent period after each scale-up. ● The minimum observation period is 5 minutes.
Average CPU Usage	Indicates threshold for triggering an auto scale-up. Allowed range: 50%–100%
Max. Specifications	Indicates the maximum specifications after the final auto scale-up. The specifications can only be scaled up gradually and the system enters the silent period after each scale-up.
Max. Read Replicas	Only one read replica can be added at a time.
Replica Read Weight	If you have enabled read/write splitting, the new read replicas are automatically associated with the proxy instance.
Auto Scale-down	You can enable or disable it as needed. **NOTE** Once auto scale-down is enabled, if the system observes an average CPU usage of 99% drops below 30% within the observation period, it gradually restores the original configuration. The system enters a silent period after each scale-down.
Silent Period	The silent period is the minimum interval between two changes (triggered automatically or manually), where no more changes can happen.

Step 7 Click **OK**.

----End

Viewing Change History

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Configuration** area, click **View Change History** under **Auto Scaling**.

Step 6 In the displayed dialog box, view the change time, change type, status, original specifications, and new specifications.

----End

6.4.4 Changing the Maintenance Window of a DB Instance

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required. To prevent service interruption, set the maintenance window to off-peak hours.

Constraints

Before maintenance is performed, GaussDB(for MySQL) will send SMS messages and emails to the contact person that has been set in the Huawei ID.
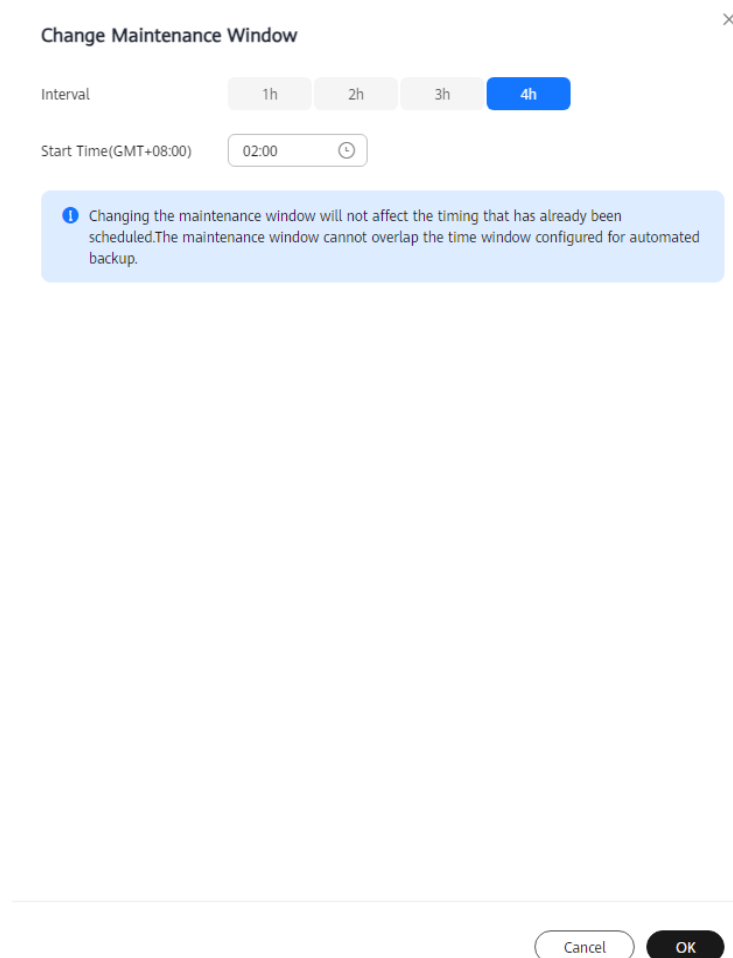
Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 On the **Instance Information** page, click **Change** under **Maintenance Window**.

Figure 6-15 Changing a maintenance window (1)

Step 6 In the displayed dialog box, select a maintenance window and click **OK**.

Figure 6-16 Changing a maintenance window (2)

----**End**

APIs

Changing a Maintenance Window

6.4.5 Customizing Displayed Items of the Instance List

Scenarios

You can customize instance information items displayed on the **Instances** page based on your requirements.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click ⊙ and select items displayed in the instance list.

- The following items are displayed by default and cannot be hidden: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine**, **Status**, **Billing Mode**, **Private IP Address**, **Storage Type**, and **Operation**.

- The following items can be displayed or hidden: **Private IP Address for Read**, **Proxy Address**, **Private Domain Name**, **Enterprise Project**, **Created**, and **Database Port**.

----End

6.4.6 Upgrading the Minor Kernel Version of a DB Instance

Scenarios

You can upgrade the minor kernel version of your DB instance to improve performance, optimize functions, and fix bugs.

Constraints

- For details about the minor versions, see **Kernel Version Release History**.

- When any new minor kernel version is released for addressing issues and vulnerabilities from the open source community, upgrade the minor kernel version of your DB instance **immediately** or **during the maintenance window**.

- An upgrade will reboot your DB instance and interrupt services intermittently. To minimize the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.

- If a DB instance contains a large number of table partitions (more than 1 million), it may take more than 2 hours to reboot the instance.

- If you want to upgrade the minor kernel version of your DB instance from 8.0.18 to 8.0.22 and there are more than 1,000 partitions, the upgrade may fail. Contact Huawei Cloud engineers to check the version compatibility before the upgrade.

- If the primary node and read replicas of a DB instance are deployed in the same AZ, a minor kernel version upgrade will trigger a failover. If they are in different AZs, a minor kernel version upgrade will trigger two failovers. A failover means that the system fails over to a read replica in case the primary node is unavailable.

- When you upgrade a minor kernel version of a DB instance, minor versions of read replicas (if any) will also be upgraded automatically. Minor versions of read replicas cannot be upgraded separately. A minor kernel version upgrade cannot be rolled back after the upgrade is complete.

- DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a minor kernel version upgrade.

- If the replication delay between the primary node and read replicas is longer than 300 seconds, the minor kernel version cannot be upgraded.

- If the kernel version is earlier than 2.0.51.240305, it will be upgraded to 2.0.51.240305 first.

- To upgrade the kernel version to 2.0.54.240600 or later, ensure that **rds_global_sql_log_bin** is **ON** and **binlog_expire_logs_seconds** is greater than or equal to **86400**. For details about parameter settings, see **Modifying Parameters of a DB Instance**.

Upgrading the Minor Kernel Version of a Single DB Instance

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

Step 2 Click   in the upper left corner and select a region and project.

Step 3 Click   in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.
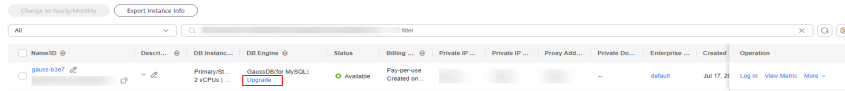
Step 5 In the **Instance Information** area, click **Upgrade** under **Kernel Version**.

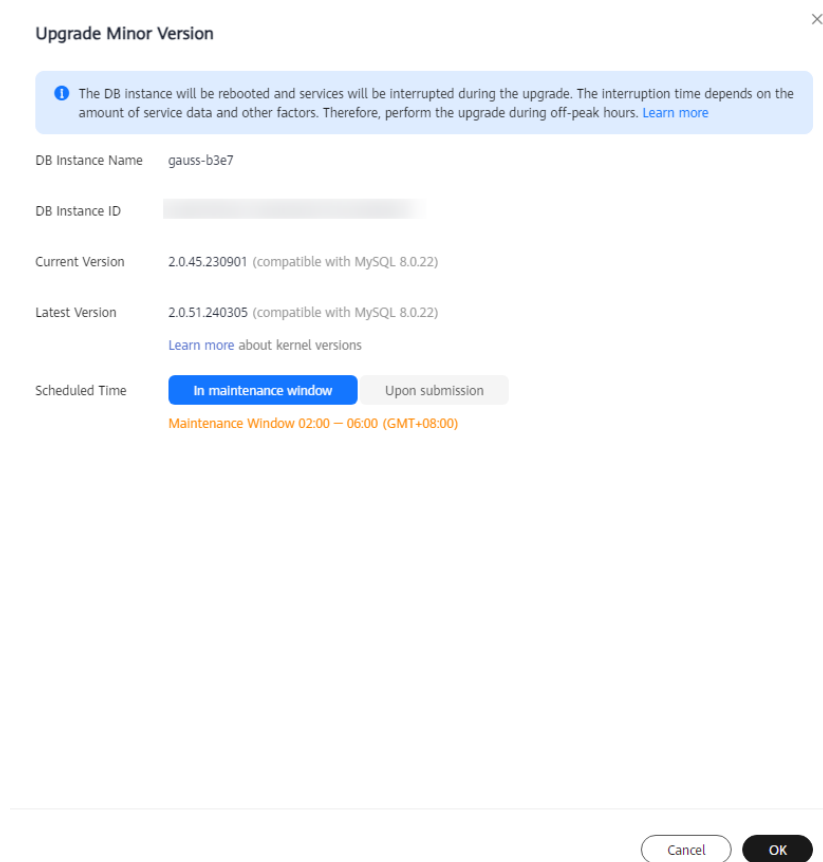**Figure 6-17** Upgrading the minor kernel version on the **Basic Information** page
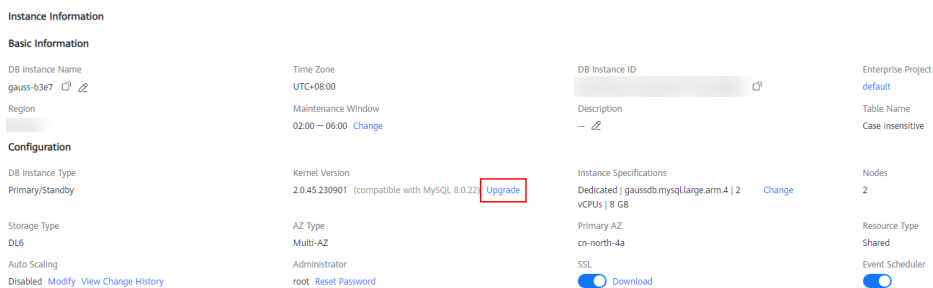
Alternatively, go to the **Instances** page, click **Upgrade** in the **DB Engine** column.

Figure 6-18 Upgrading the minor kernel version on the **Instances** page

Step 6 In the displayed dialog box, set **Scheduled Time** and click **OK**.

**Figure 6-19** Upgrading the minor kernel version of a DB instance

- Upon submission: The system upgrades the minor kernel version immediately after your submission of the upgrade request. In the **Task Center** page, click **Instant Tasks** and view the task progress.
- **In maintenance window**: The system will upgrade the minor kernel version during a maintenance window you specify. After the operation is complete, on the **Task Center** page, click **Scheduled Tasks** and view the information about the upgrade task.

   ----End

Upgrading Minor Kernel Versions of Multiple DB Instances at a Time

Step 1 On the **Instances** page, select the desired DB instances and click **Upgrade** in the upper left corner of the list.

Figure 6-20 Upgrading minor kernel versions of multiple DB instances at a time
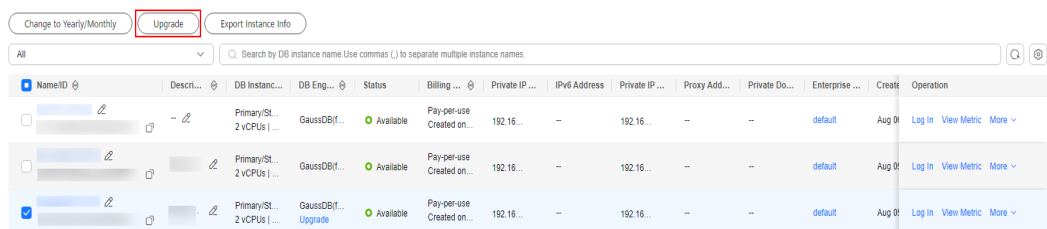
> **NOTICE**
>
> A maximum of 100 DB instances can be selected at a time.

Step 2 In the displayed dialog box, confirm the information about the DB instances to be upgraded and set **Scheduled Time**.

Figure 6-21 Selecting a scheduled time

- **Upon submission**: The system upgrades the minor kernel version immediately after your submission of the upgrade request. In the **Task Center** page, click **Instant Tasks** and view the task progress.

- **In maintenance window**: The system will upgrade the minor kernel version during a maintenance window you specify. After the operation is complete, on the **Task Center** page, click **Scheduled Tasks** and view the information about the upgrade task.

Step 3 Confirm the information, enter **YES** in the text box as prompted, and click **OK**.

> ⚠ CAUTION
>
> - Wait for 2 to 5 minutes and check whether the upgrade has been started for the DB instance. If the upgrade has not been started, check whether the value of **rds_global_sql_log_bin** is **ON** and the value of **binlog_expire_logs_seconds** is at least **86400**. If the parameters are not correctly configured, the upgrade cannot be performed.
> - If the parameters are correctly configured but the upgrade has not started, it could be due to that the value of **rds_sql_log_bin_inconsistent_count** is not **0**. Wait until this value becomes **0** before proceeding with the upgrade.

----End

APIs

Upgrading the Kernel Version of a DB Instance

6.4.7 Updating the OS of a DB Instance

To improve database performance and security, the OS of your GaussDB(for MySQL) instance needs to be updated timely.

Every time you upgrade the kernel version of your instance, GaussDB(for MySQL) determines whether to update the OS and selects the right cold patch to upgrade the OS if necessary.

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

In addition, GaussDB(for MySQL) installs hot patches as required to fix major OS vulnerabilities within the maintenance window you specified.

7 Data Backups

7.1 Backup Principles

GaussDB(for MySQL) uses Huawei's DFV storage, which decouples storage from compute. The compute layer provides services for external systems and manages logs, and the storage layer stores data. The storage layer consists of Common Log nodes and Slice Store nodes. Common Log nodes store logs, while Slice Store nodes store data.

Figure 7-1 Backup principles
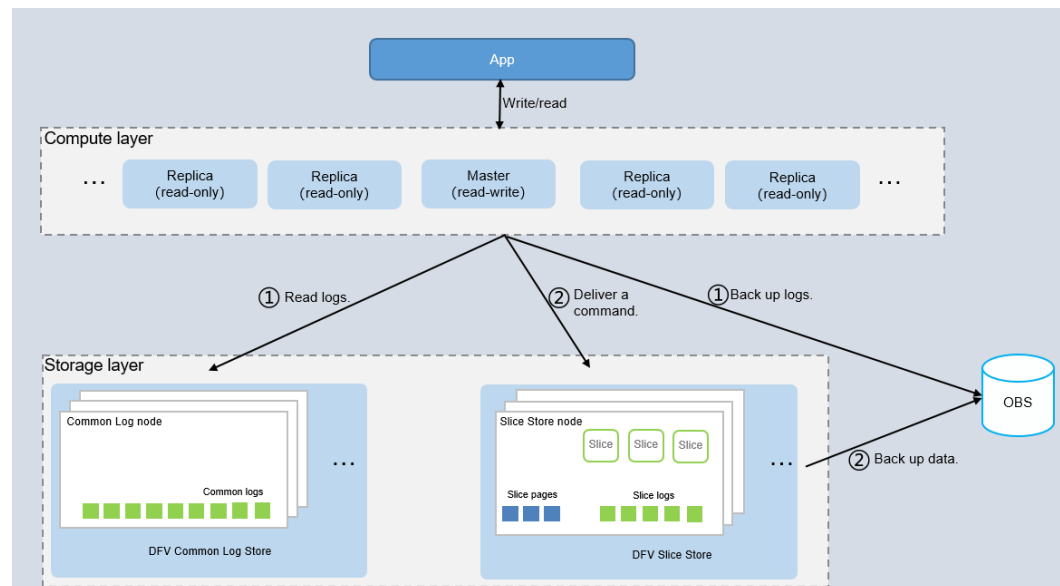
As shown in **Figure 7-1**, the creation of backups involves in the compute layer and storage layer.

- The primary node at the compute layer reads the logs of the Common Log nodes at the storage layer and backs them up to OBS.

- The primary node at the compute layer delivers a command for backing up data to the Slice Store nodes at the storage layer. The Slice Store nodes back up data to OBS.

During the creation of a backup, the CPU usage and memory usage of the primary node of your instance increase slightly, but you will not notice anything at the storage layer. The final backup is stored in OBS as multiple data files and does not use up any of the disk space of the instance.
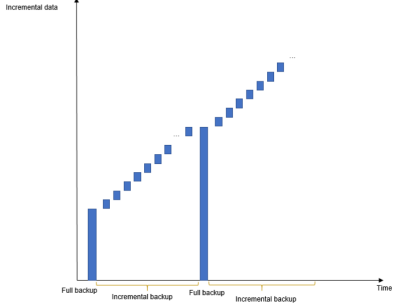
7.2 Backup Types

Based on different dimensions, there are the following backup types in GaussDB(for MySQL).

- Full backups and incremental backups based on data volume

Table 7-1 Comparison between full backups and incremental backups

Backup Type	Full backups	Incremental backups
Description	All data in your DB instance is backed up.	Only data that has changed within a certain period is backed up.
Enabled by Default	Yes	Yes
Retention Period	Full backups are retained till the retention period expires.	Incremental backups are retained till the retention period expires.

Character istic	• A full backup is to back up all data of your DB instance in the current point of time. • You can use a full backup to restore the complete data generated when the backup was created. • Full backups include automated backups and manual backups.	• GaussDB(for MySQL) automatically backs up the updated data since the last automated or incremental backup was made every five minutes or after a certain amount of data is reached. • Incremental backups are automated backups. • Incremental backups are created based on the most recent full backup, as shown in **Figure 7-2**, so the most recent full backup that exceeds the retention period is still retained. For details, see the following example. **Figure 7-2** Incremental data restoration
How to View	Click the instance name. On the **Backups** page, click the **Full Backups** tab and view the backup size.	Click the instance name. On the **Backups** page, click the **Incremental Backups** tab and view the backup size.

- Automated backups and manual backups based on backup methods

Table 7-2 Comparison between automated backups and manual backups

Backup Type	Automated backups	Manual backups

Description	● You can set an automated backup policy on the management console, and the system will automatically back up your instance data based on the time window and backup cycle you set in the backup policy and will store the data for the retention period you specified. ● Automated backups cannot be manually deleted. To delete them, you adjust the retention period specified in your backup policy. Retained backups (including full and incremental backups) will be automatically deleted at the end of the retention period.	● Manual backups are user-initiated full backups of your DB instance. They are retained until you delete them manually. ● Regularly backing up your DB instance is recommended, so if your DB instance becomes faulty or data is corrupted, you can restore it using backups.
Enabled by Default	Yes	Yes
Retention Period	Automated backups are retained for the retention period you specified. The backup retention period ranges from 1 to 732 days.	Manual backups are always retained until you delete them manually.
Configuration	**Configuring a Same-Region Backup Policy**	**Creating a Manual Backup**

● Same-region backups and cross-region backups based on backup regions

☐ **NOTE**

    To configure cross-region backup policies, contact customer service.

Table 7-3 Comparison between same-region backups and cross-region backups

Backup Type	Same-region backups	Cross-region backups

Description	Backups are stored in the same region as your DB instance.	Backups are stored in a different region from that of your DB instance.
Enabled by Default	Yes	No
Retention Period	Same-region backups are retained for the retention period you specified. The backup retention period ranges from 1 to 732 days. **NOTE** You can contact customer service to extend the backup retention period to up to 3,660 days.	Cross-region backups are retained for the retention period you specified. The backup retention period ranges from 1 to 1,825 days.
Characteristic	Backups are stored in the same region as your DB instance. Same-region backup (automated backup) is enabled by default and cannot be disabled.	Backups are stored in a region different from the region where your DB instance is located. After you enable cross-region backup, the backups are automatically stored in the region you specify.
Configuration	**Configuring a Same-Region Backup Policy**	**Configuring a Cross-Region Backup Policy**
How to View	• If cross-region backup is enabled: Click **Backups** in the navigation pane. On the **Same-Region Backups** tab, view the backup size. • If cross-region backup is not enabled: Click **Backups** in the navigation pane and view the backup size.	Click **Backups** in the navigation pane. On the **Cross-Region Backups** tab, locate a DB instance and click **View Cross-Region Backup** in the **Operation** column.

7.3 Backup Space and Billing

Concepts

- Full backup: All data is backed up even if no data has changed since the last backup.

- Incremental backup: The system automatically backs up data that has changed since the last automated backup or incremental backup in binlogs every 5 minutes. The binlogs can be used to restore data to a specified point in time.

- Differential backup: The system backs up data that has changed since the most recent full backup or differential backup in to physical files. Physical files cannot be used for log replay.

- Billed space: backup space that you are billed for

- Logical space: space occupied by full backups

- Physical space: the amount of data that is backed up to OBS

  📖 NOTE

  After you purchase a DB instance, the logical space is the same as the physical space. When a backup starts in a backup chain, the physical space stores the data of the first full backup and subsequent differential backups.

Backup Space Calculation Methods

There is a default backup chain (where there are seven backups). The first automated backup is a full backup, and subsequent automated backups are differential backups.

In a backup chain, the backup space is released only after all full backups and differential backups are deleted.

Billed space is calculated as follows:

Billed space = Min(Logical space, Physical space) – Free space = Min(Logical space, Physical space) – Storage space x 100%

- Logical space: Total size of the logical space – Logical size of the expired backup

- Physical space: Size of the first full backup + Total size of subsequent differential backups

- Free space: There is free backup storage up to 100% of your purchased storage space.

Example

A backup chain contains seven backups by default. There are 11 backups shown in the following figure. The 1st backup to the 7th belong to one backup chain and the 8th to the 11th belong to another.

Figure 7-3 Backup example
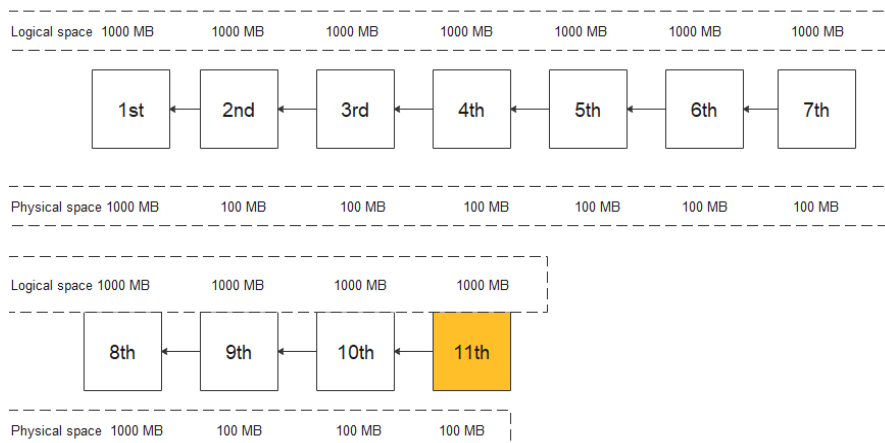
If there is 1,000 MB of backup space and the logical space is 1,000 MB each time, the physical space for the 1st backup is 1,000 MB. If the incremental data size is 100 MB each time, the physical space for the 2nd backup to 7th is 100 MB.

A backup chain contains seven backups by default. The physical space for the 8th backup is 1,000 MB because it represents a new backup chain.

Billed space includes the space of the two chains in the example.

Suppose that after the 11th backup was created, and the 1st, 2nd and 3rd backups expired and were automatically deleted. The size of each space is calculated as follows:

- Total logical space size = Total logical space size – Logical size of the expired backup (1,000 MB x 11 - 3,000 MB = 8,000 MB in this example)

- Physical space: size of data backed up to OBS. In this example, physical space includes the sum of physical space on the two backup links: 1,000 MB + (100 MB x 6) +1,000 MB+ (100 MB X 3) =2,900 MB

- Total billed space = Min (Total size of logical space, Total size of physical space) – free space, so the total billed space in this example = Min (8,000 MB, 2,900 MB) – 1,000 MB = 1,900 MB

7.4 Creating an Automated Backup

7.4.1 Configuring a Same-Region Backup Policy

Scenarios

When you create a GaussDB(for MySQL) instance, the automated backup policy is enabled by default. However, it can be modified after instance creation is complete. GaussDB(for MySQL) backs up data based on the automated backup policy you specify.

GaussDB(for MySQL) backs up data at the DB instance level. If a DB instance is faulty or data is damaged, you can still restore it using backups to ensure data

reliability. 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.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

You do not need to configure incremental backup policies because the system automatically performs an incremental backup every 5 minutes. The generated incremental backups can be used to restore the database and table data to a specified point in time.

Constraints

- The automated backup policy is enabled by default and cannot be disabled.
- Rebooting instances is not allowed during the creation of a full backup. Exercise caution when selecting a backup time window.
- When starting a full backup task, GaussDB(for MySQL) first tests connectivity to your instance. If the backup lock failed to be obtained from the DB instance, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
- Performing a full backup may decrease instance throughput because it occupies node resources, especially disk bandwidth.

Backup Clearing

To ensure data integrity, even after the retention period expires, the most recent full backup will be retained, for example, if **Backup Cycle** was set to **Monday** and **Tuesday** and **Retention Period** was set to **2**:

- The full backup generated on Monday will be automatically deleted on Thursday because:

  The backup generated on Monday expires on Wednesday, but it was the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday.

- The full backup generated on Tuesday will be automatically deleted on the following Wednesday because:

  The backup generated on Tuesday will expire on Thursday, but as it is the last backup, it will be retained until a new backup expires. The next backup will be generated on the following Monday and will expire on the following Wednesday, so the full backup generated on Tuesday will not be automatically deleted until the following Wednesday.

Viewing or Modifying a Same-Region Backup Policy

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

Step 2 Click ⊙ in the upper left corner and select a region and project.
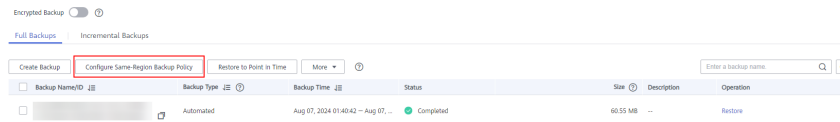
Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Backups**.

Step 6 On the displayed **Full Backups** tab, click **Configure Same-Region Backup Policy**.

Figure 7-4 Configuring a same-region backup policy

Step 7 In the displayed dialog box, view the current backup policy. To modify the backup policy, adjust the parameter values by referring to **Table 7-4**.
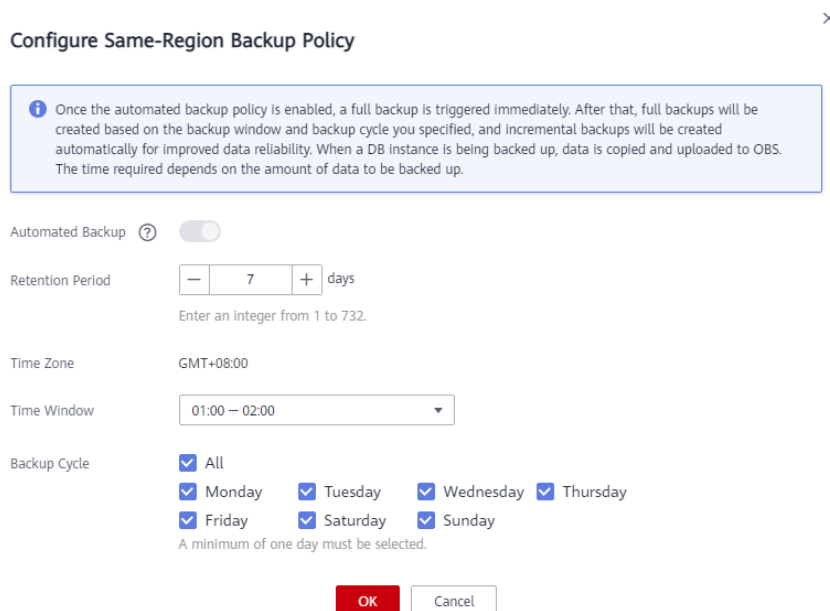
Figure 7-5 Modifying backup policies

Table 7-4 Parameter description

Parameter	Description
Retention Period	Number of days that your automated backups can be retained. The retention period is from 1 to 732 days and the default value is **7**. ● Extending the retention period improves data reliability. You can configure the retention period if needed. ● If you shorten the retention period, the new backup policy takes effect for existing backups. Any backups (including full and automated backups) that have expired will be automatically deleted. Manual backups will not be automatically deleted but you can delete them manually. **NOTE** To extend the retention period to 3,660 days, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
Time Zone	The backup time is in UTC format. The backup time segment changes with the time zone during the switch between the DST and standard time.
Time Window	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00.
Backup Cycle	By default, each day of the week is selected. You can change the backup cycle and must select at least one day of the week.

Step 8 Click **OK**.

----End

APIs

- **Configuring a Same-Region Backup Policy**
- **Querying an Automated Backup Policy**

7.4.2 Configuring a Cross-Region Backup Policy

Scenarios

GaussDB(for MySQL) can store backups in a different region from your DB instance for disaster recovery. If your DB instance in a region is faulty, you can use the backups in another region to restore data to a new DB instance.

After you enable cross-region backup, the backups are automatically stored in the region you specify.

Precautions

- To apply for the permissions to configure cross-region backup policies, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

- Cross-region backup can be enabled for up to 150 DB instances in a single region under a tenant. It is recommended that the data volume of a single DB instance be at most 2 TB. If the data volume is too large, the synchronization progress may be delayed.

Supported Regions

Table 7-5 Supported regions

Instance Region	Backup Region
CN North-Beijing4	CN East-Shanghai1, CN North-Ulanqab1, CN Southwest-Guiyang1, and CN South-Guangzhou
CN East-Shanghai1	CN North-Beijing4, CN North-Ulanqab1, CN Southwest-Guiyang1, and CN South-Guangzhou
CN North-Ulanqab1	CN North-Beijing4, CN East-Shanghai1, CN Southwest-Guiyang1, and CN South-Guangzhou
CN Southwest-Guiyang1	CN North-Beijing4, CN East-Shanghai1, CN North-Ulanqab1, and CN South-Guangzhou
CN South-Guangzhou	CN North-Beijing4, CN East-Shanghai1, CN North-Ulanqab1, and CN Southwest-Guiyang1

Billing

For details, see **GaussDB(for MySQL) Cross-Region Backup Billed Items**.

Enabling or Modifying a Cross-Region Backup Policy

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Configure Cross-Region Backup Policy**.

Figure 7-6 Configuring a cross-region backup policy
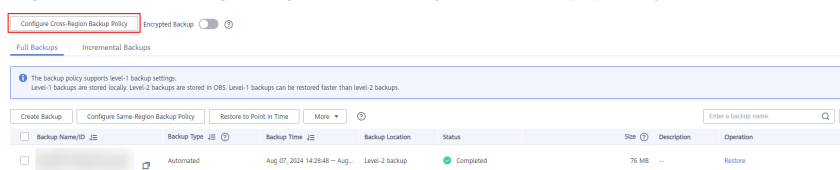
Step 6 In the displayed dialog box, set required parameters.
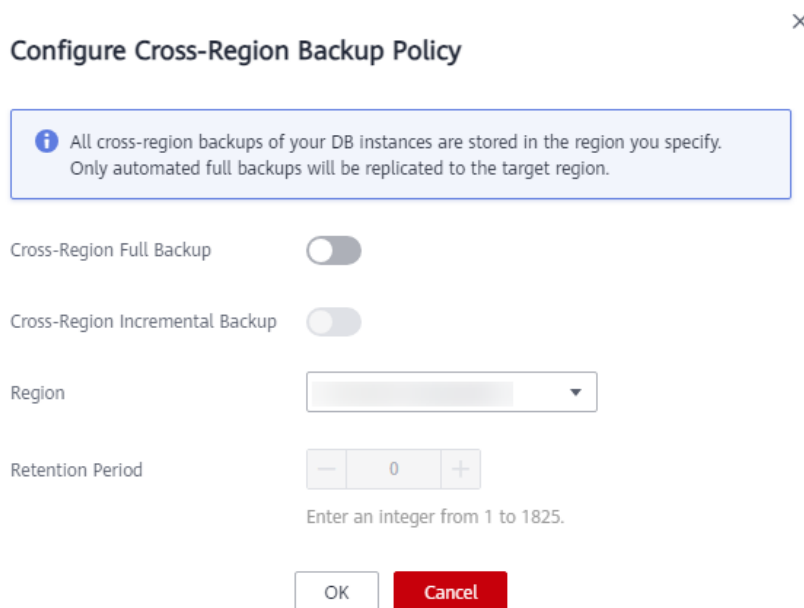
Figure 7-7 Configuring a backup policy

Table 7-6 Parameter description

Parameter	Description
Cross-Region Full Backup	If you enable this option, automated full backups of the DB instance are stored in the region you specify.
Cross-Region Incremental Backup	If you enable this option, incremental backups of the DB instance are stored in the region you specify. ● To enable cross-region incremental backup, enable cross-region full backup first. ● After cross-region incremental backup is enabled, you can restore an instance to a specified point in time only after the next automated full backup replication is complete. The specified point in time must be later than the time when the automated full backup is complete.
Region	Select the region for storing backups.
Retention Period	Cross-region backup files can be retained from 1 to 1,825 days.

Step 7 Click **OK**.

Step 8 On the **Cross-Region Backups** tab of the **Backups** page, manage cross-region backups.

Figure 7-8 Cross-region backups

- By default, all instances with cross-region backups are displayed.

  To modify the cross-region backup policy, click **Set Cross-Region Backup** in the **Operation** column.

  To view generated cross-region backup files, click **View Cross-Region Backup** in the **Operation** column. If a DB instance fails, you can use the cross-region backups to restore data to a new DB instance.

  Full or incremental backups can be resorted to a new DB instance. Select the backup you want to restore and click **Restore** in the **Operation** column.

  Figure 7-9 Full backups

  Figure 7-10 Incremental backups

  Incremental backups can be restored to a point in time. You need to select a time range, select or enter a time point within the acceptable range.

  Figure 7-11 Restoring an incremental backup to a point in time

- To view all cross-region backups, click **View All Backups**.

  To restore a backup, locate the backup and click **Restore** in the **Operation**. For details, see **Restoring a DB Instance from a Cross-region Backup**.

- To return to the instance list, click **View Instances**.

----End

Disabling a Cross-Region Backup Policy

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

Step 2 Click ⦿ in the upper left corner and select a region and project.

**Step 3** Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the upper left corner of the page, select the region where the original DB instance is located.

Step 5 Disable the cross-region backup policy using either of the following methods.

Method 1:

1. Choose **Backups** in the navigation pane and click the **Cross-Region Backups** tab.

2. Locate the target DB instance and click **Set Cross-Region Backup** in the **Operation** column.

   Figure 7-12 Setting cross-region backup
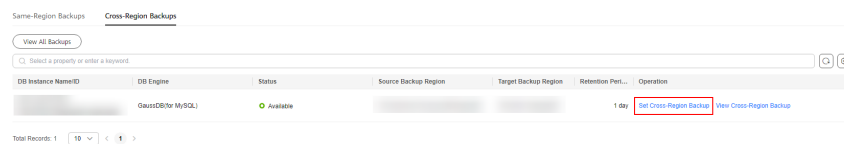
3. In the displayed dialog box, click ⬤ next to **Cross-Region Incremental Backup** and **Cross-Region Full Backup**.
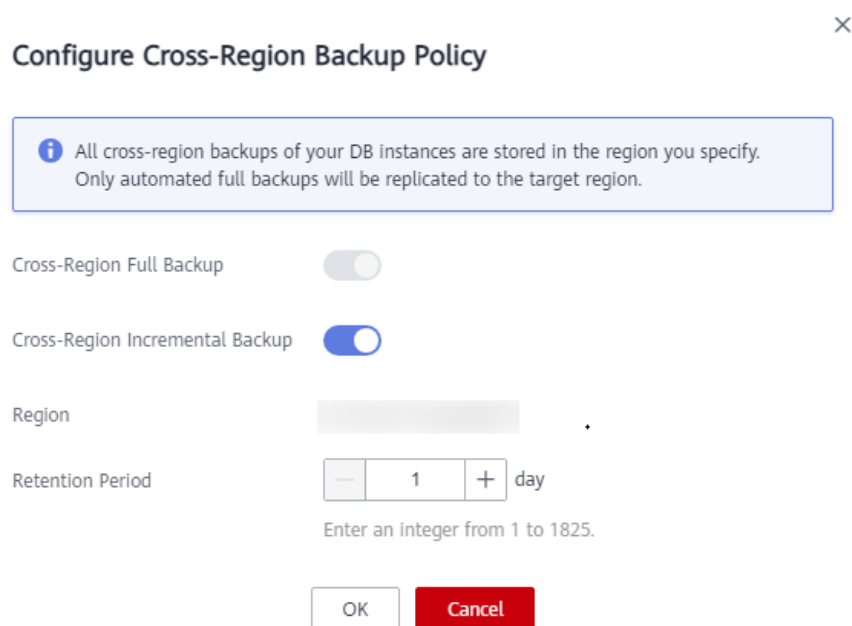
Figure 7-13 Disabling cross-region backup
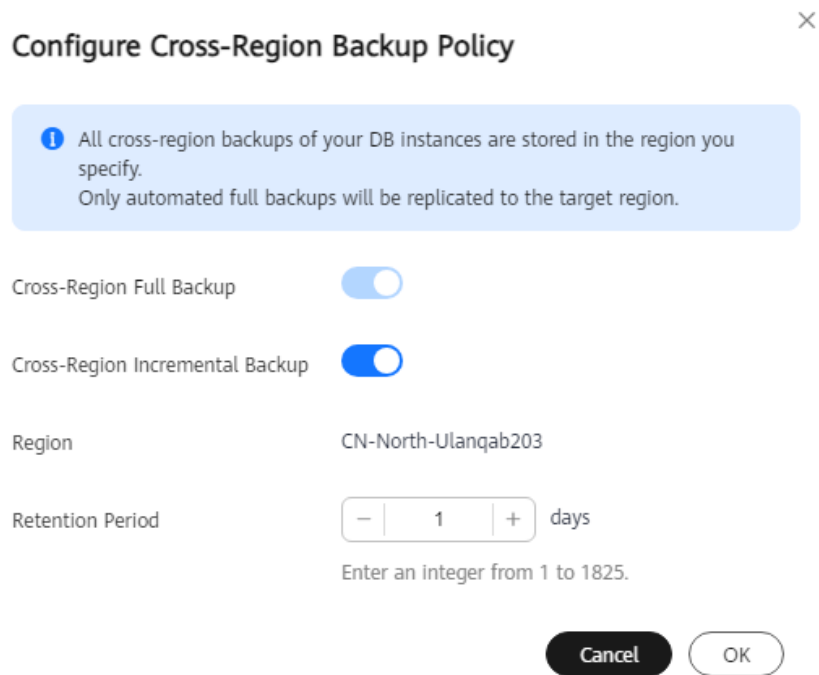
4. Click **OK**.

Method 2:

1. On the **Instances** page, click the instance name.
2. In the navigation pane, choose **Backups**.
3. Click **Configure Cross-Region Backup Policy**.
4. In the displayed dialog box, click next to **Cross-Region Incremental Backup** and **Cross-Region Full Backup**.

Figure 7-14 Disabling cross-region backup

5.   Click **OK**.

----End

7.5 Creating a Manual Backup

Scenarios

GaussDB(for MySQL) allows you to create manual backups for available DB instances. You can use these backups to restore data.

Constraints

- You can create manual backups only when your account balance is no less than $0 USD.

- The backup efficiency is in direct proportion to the instance data volume.

- The system verifies the connection to the DB instance when starting a full backup task. If the backup lock failed to be obtained from the DB instance, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.

- When an account is deleted, both automated and manual backups are deleted.

- The time required for creating a manual backup depends on the data volume of the DB instance.

- When a DB instance is being backed up, data is copied and uploaded to OBS. The time required depends on the amount of data to be backed up.

Backup Clearing

When a DB instance is deleted, its automated backups are also deleted, but its manual backups are retained until **you manually delete them**.

Creating a Manual Backup

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

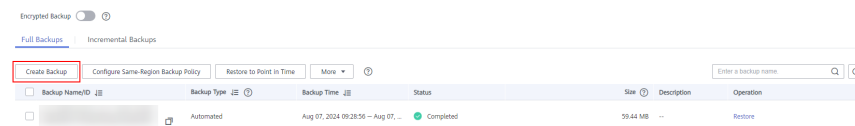**Step 4** On the **Instances** page, locate the instance for which you want to create a manual backup and choose **More** > **Create Backup** in the **Operation** column.

Figure 7-15 Creating a backup
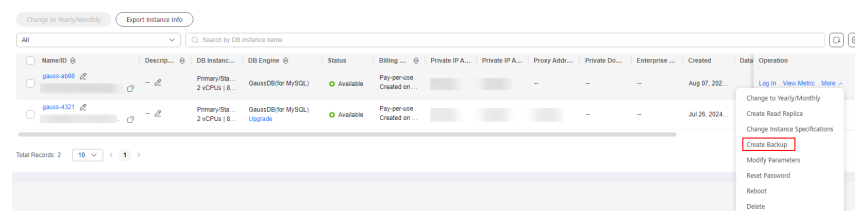
Alternatively, on the **Instances** page, click the instance name. Choose **Backups** in the navigation pane and click **Create Backup**.

Figure 7-16 Creating a backup

Step 5 In the displayed dialog box, enter a backup name and description and click **OK**.

Figure 7-17 Creating a backup

- The backup name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.

- The backup description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters: !<"='>&

Step 6 View the created backup on the **Backups** page.

----End

Deleting a Manual Backup

Step 1 On the **Backups** page, locate the backup you want to delete and click **Delete** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name. On the **Backups** page, locate the backup you want to delete and click **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored or created

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

----End

APIs

- **Creating a Manual Backup**
- **Querying Backups**

7.6 Enabling or Disabling Encrypted Backup

Scenarios

GaussDB(for MySQL) can encrypt backups. After encrypted backup is enabled, a key is required, which is generated and managed by **Data Encryption Workshop (DEW)**.

Precautions

- Only the backups generated after encrypted backup is enabled will be encrypted.

- After encrypted backup is disabled, new backup files will not be encrypted for storage. Backup files created before encrypted backup is disabled will not be decrypted.

- Currently, only the SM4 and AES_256 key algorithms are supported. After encrypted backup is enabled, the key algorithm cannot be changed.

- The key cannot be disabled, deleted, or frozen while in use, or the encrypted backups cannot be used for restoration.

- Encrypted backups can be directly used to restore data on the management console. You do not need to manually decrypt backups.
- Once encrypted backup is enabled for your DB instance, data cannot be restored to an existing DB instance, even if encrypted backup is disabled later.
- Cross-region backup and encrypted backup cannot be both enabled.
- When encrypted backup is enabled for a DB instance, only the key of the corresponding enterprise project can be selected. To view keys in an enterprise project, see **Viewing a CMK**.

Enabling Encrypted Backup

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

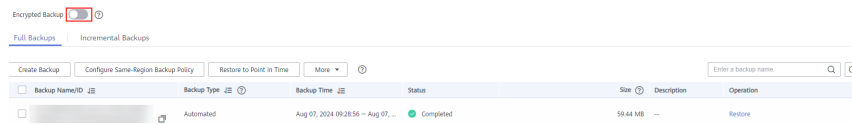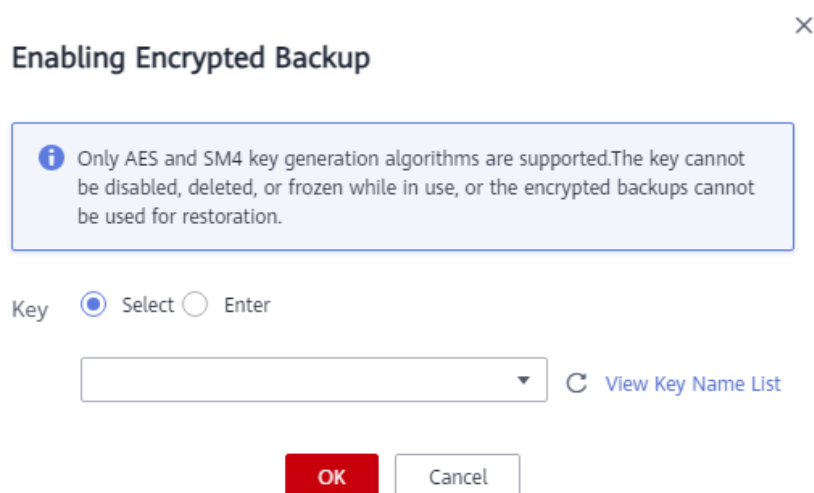**Step 4** On the **Instances** page, click the instance name.

Step 5 Choose **Backups** in the navigation pane and click ⬭ next to **Encrypted Backup**.

Figure 7-18 Enabling encrypted backup

Step 6 In the displayed dialog box, select a key or enter a key ID and click **OK**.

Only SM4 and AES_256 key algorithms are supported.

Figure 7-19 Selecting a key

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

Step 8 Refresh the page and check whether encrypted backup is enabled.

----End

Disabling Encrypted Backup

Step 1 On the **Instances** page, click the instance name.

Step 2 Choose **Backups** in the navigation pane and click ⬤▬ next to **Encrypted Backup**.

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

----End

7.7 Exporting Backup Information

Scenarios

You can export backup information of a GaussDB(for MySQL) instance to an Excel file for further analysis. The exported information includes the instance name/ID, backup name/ID, DB engine, backup type, backup time, status, size, and description.

Constraints

Automated and manual backups cannot be downloaded.

Procedure

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

Step 2 Click 🔘 in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

**Step 4** In the navigation pane, choose **Backups**.

Step 5 On the **Same-Region Backups** tab, select the backups to be exported and click **Export**.

Figure 7-20 Exporting backup information
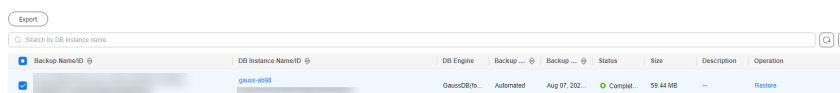
Alternatively, on the **Instances** page, click the instance name. In the navigation pane, choose **Backups**. On the **Full Backups** tab, select the backups to be exported and choose **More** > **Export**.

Figure 7-21 Exporting backup information
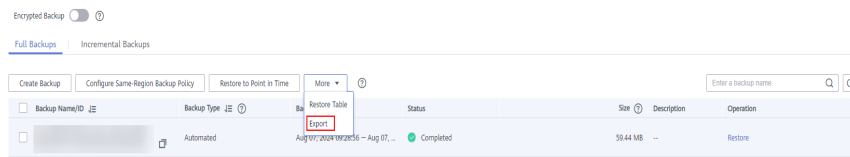
- Currently, only the backup information displayed on the current page can be exported.
- The backup information is exported to an Excel file.

Step 6 View the exported backup information.

----End

8 Data Restorations

8.1 Restoration Solutions

If your database is damaged or mistakenly deleted, you can restore it using the following methods.

Restoring or Migrating Data to GaussDB(for MySQL)

- You can restore data using backups. For details, see **Restoring a DB Instance from Backups**.
- You can migrate data using DRS, mysqldump, or DAS. For details, see **Data Migration Schemes**.

Restoring Deleted or Modified Data

The following table describes how to restore deleted tables, deleted databases, deleted instances, deleted or modified columns, rows, and data in tables, and overwritten tables.

Table 8-1 Restoration solutions

Scenario	Restoration Solution	Restoration Scope	Restore To	Operation Guide
Restoring a deleted instance	If the deleted instance is in the recycle bin, rebuild it by referring to **Rebuilding a DB Instance in the Recycle Bin**.	All databases and tables	The original instance	**Rebuilding a DB Instance**

Scenario	Restoration Solution	Restoration Scope	Restore To	Operation Guide
	If **a manual backup has been created** before the instance was deleted, restore the instance on the **Backups** page.	All databases and tables	- A new instance - An existing instance - The original instance	**Restoring a DB Instance from Backups**
Restoring a deleted table	Use the database table restoration method to restore the table.	- All databases and tables - Some databases and tables	- A new instance - An existing instance - The original instance	**Restoring Tables to a Point in Time**
Restoring a deleted database	Use the database table restoration method to restore the database.	- All databases and tables - Some databases and tables	- A new instance - An existing instance - The original instance	**Restoring Tables to a Point in Time**
Restoring deleted or modified columns, rows, and data in tables, and overwritten tables	If more than 100,000 data records are deleted or modified, use the database table restoration method to restore data in the table.	- All databases and tables - Some databases and tables	- A new instance - An existing instance - The original instance	**Restoring Tables to a Point in Time**

8.2 Restoring a DB Instance from Backups

Scenarios

You can use an automated or manual backup to restore data to the point in time when the backup was created. The restoration is at the instance level.

A full backup will be downloaded from OBS for restoration. The time required depends on the amount of data to be restored.

Prerequisites

There is an automated or manual backup available for the DB instance. If you want to restore a deleted DB instance, you need to ensure that there is a manual backup because automated backups are deleted along with the DB instance.

Precautions

- Data can be restored to a new, the original, or an existing DB instance.
- Keep your account balance above zero so that backups can be restored to a new DB instance. You will be billed for the new DB instance.
- Data on the original DB instance will be overwritten and the original DB instance will be unavailable during the restoration.
- Restoring to an existing DB instance will overwrite data and reset the root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.
- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.
- Once encrypted backup is enabled for your DB instance, data cannot be restored to an existing DB instance, even if encrypted backup is disabled later.
- Ensure that the storage space of the selected DB instance is at least that of the original DB instance. Otherwise, data will not be restored.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

Step 5 Select where you want to restore the backup to:

- Restoring the backup to a new DB instance

  a. Set **Restoration Method** to **Create New Instance** and click **OK**.

Figure 8-1 Restoring to a new DB instance
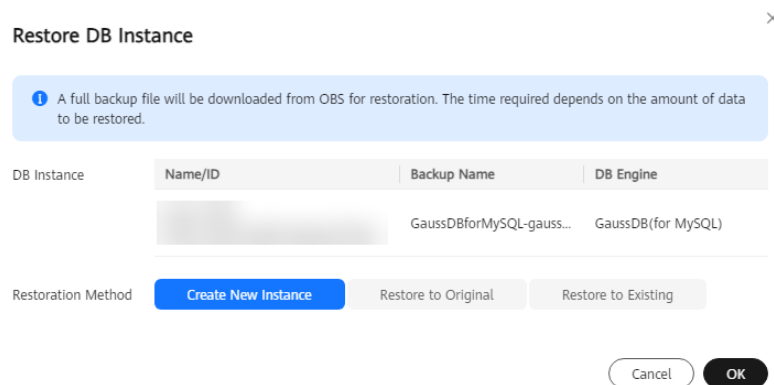
b. On the displayed page, set required parameters and click **Next**.

- The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.

- The default database port is **3306**.

- Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

- Restoring the backup to the original DB instance

  a. Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.

  Figure 8-2 Restoring to the original DB instance
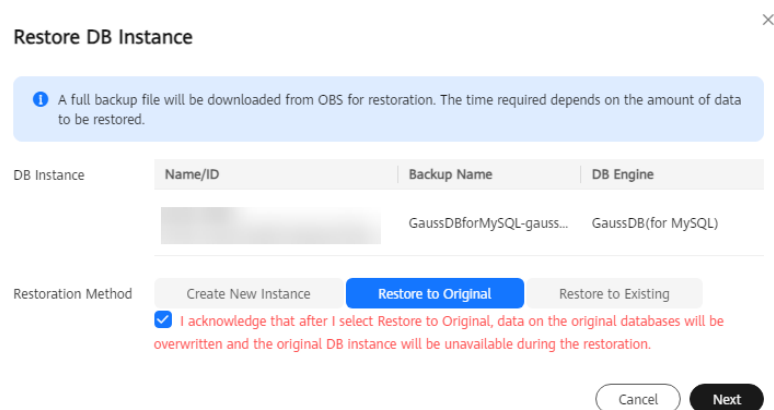
  b. Confirm the task details and click **OK**.

  Data on the original DB instance will be overwritten and the original DB instance will be unavailable during the restoration.

- Restoring the backup to an existing DB instance

  a. Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, and click **Next**.

Figure 8-3 Restoring to an existing DB instance

b. Confirm the task details and click **OK**.

- Restoring to an existing DB instance will overwrite data and reset the root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.

- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.

Step 6 View the restoration results.

- Restoring the backup to a new DB instance

  A new DB instance is created, where data is restored based on the point in time when the backup was created. When the instance status changes from **Creating** to **Available**, the restoration is complete.

  The new DB instance is independent from the original one. If you want to offload read pressure from the primary node, create one or more read replicas for the new DB instance.

  After the restoration, a full backup will be automatically triggered.

- Restoring the backup to the original DB instance

  On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete.

  After the restoration, a full backup will be automatically triggered.

- Restoring the backup to an existing DB instance

  On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. If the existing DB instance contains

read replicas, the read replica status is the same as the existing DB instance status.

After the restoration, a full backup will be automatically triggered.

----End

APIs

- **Restoring Data to the Original Instance or an Existing Instance**
- **Querying the Restoration Time Range**

8.3 Restoring a DB Instance to a Point in Time

Scenarios

You can restore a DB instance to a specified point in time.

The most recent full backup will be downloaded from OBS for restoration. After the restoration is complete, incremental backups will be replayed to the specified point in time. The time required depends on the amount of data to be restored.

Precautions

- Data can be restored to a new, the original, or an existing DB instance.
- Keep your account balance above zero so that backups can be restored to a new DB instance. You will be billed for the new DB instance.
- Do not run the **reset master** command on DB instances within their lifecycle. Otherwise, an exception may occur when restoring a DB instance to a specified point in time.
- Data on the original DB instance will be overwritten and the original DB instance will be unavailable during the restoration.
- Restoring to an existing DB instance will overwrite data and reset the root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.
- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.
- Once encrypted backup is enabled for your DB instance, data cannot be restored to an existing DB instance, even if encrypted backup is disabled later.
- Ensure that the storage space of the selected DB instance is at least that of the original DB instance. Otherwise, data will not be restored.

## Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Restore to Point in Time**.

Figure 8-4 Restoring to a point in time
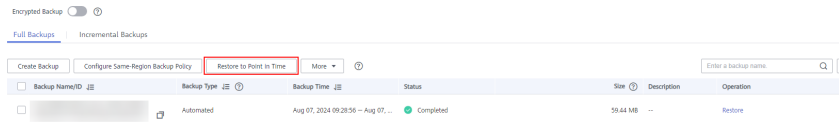
Step 6 Select the restoration date and time range, enter a time point within the selected time range, and select a restoration method. Then, click **OK**.

> **NOTE**
>
> If you have enabled operation protection, click **Start Verification** in the **Restore DB Instance** dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify**. The page is closed automatically.
>
> For details about how to enable operation protection, see *Identity and Access Management User Guide*.

- Restoring the backup to a new DB instance

  a. Set **Restoration Method** to **Create New Instance** and click **OK**.

  Figure 8-5 Restoring to a new DB instance

  b. On the displayed page, set required parameters and click **Next**.

  - The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.

  - The default database port is **3306**.

  - To synchronize database parameters of the original DB instance, select **Original DB instance parameter template**.

NOTE

- If the original DB instance is deleted, the database parameters of the original DB instance cannot be synchronized using backups.

- When you synchronize the database parameters of the original DB instance, the following parameters cannot be synchronized and you need to **manually modify them** after the DB instance is restored.

  innodb_write_io_threads

  innodb_read_io_threads

  max_connections

  innodb_log_buffer_size

  innodb_parallel_read_threads

  temptable_max_ram

  threadpool_size

  innodb_buffer_pool_size

  innodb_page_cleaners

– Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

- Restoring the backup to the original DB instance

Figure 8-6 Restoring to the original DB instance

a. Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.

b. Confirm the task details and click **OK**.

   Data on the original DB instance will be overwritten and the original DB instance will be unavailable during the restoration.

- Restoring the backup to an existing DB instance

Figure 8-7 Restoring to an existing DB instance

a. Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, and click **Next**.

b. Confirm the task details and click **OK**.

- Restoring to an existing DB instance will overwrite data and reset the root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.

- The restored DB instance contains the data and account information of the original DB instance, but does not contain the parameter settings of the original DB instance.

- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.

Step 7 View the restoration results.

- Restoring the backup to a new DB instance: A new DB instance is created, where data is restored based on the point in time when the backup was created. When the instance status changes from **Creating** to **Available**, the restoration is complete. The new DB instance is independent from the original one. If you want to offload read pressure from the primary node, create one or more read replicas for the new DB instance.

After the restoration, a full backup will be automatically triggered.

- Restoring the backup to the original DB instance: When the instance status changes from **Restoring** to **Available**, the restoration is complete.
- Restoring the backup to an existing DB instance: When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

8.4 Restoring Tables to a Point in Time

Scenarios

To ensure data integrity and reduce impact on the original instance performance, the system restores the full and incremental data at the selected point in time to a temporary instance, exports the tables to be restored, and then restores the tables to the original instance. The time required depends on the amount of data to be backed up and restored on the instance. Restoring tables will not overwrite data in the instance.

Constraints

- Tables that have triggers cannot be restored.
- To prevent restoration failures and impact on original data, table-level restoration removes foreign key constraints.
- If the tables to be restored are not found at the selected point in time, the restoration will fail.
- The DB instance cannot be rebooted or deleted, and the instance specifications cannot be modified.
- The number of tables to be restored must be no more than 20,000. If the number of tables to be restored exceeds 2,000, you can restore the instance to a point in time. For details, see **Restoring a DB Instance to a Point in Time**.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

**Step 4** On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Backups**. On the **Full Backups** tab, choose **More** > **Restore Table** above the backup list.

Figure 8-8 Restoring tables to a specified point in time
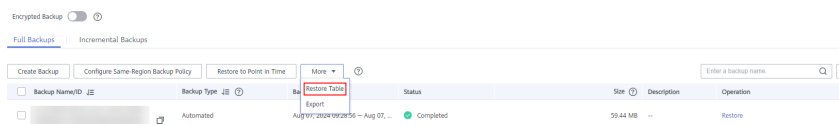
Step 6 On the displayed page, set the restoration date, time range, time point, and tables to be restored.

Figure 8-9 Setting required parameters
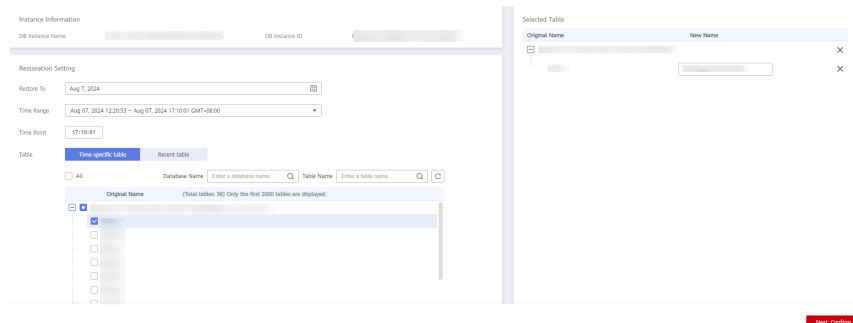
- To facilitate your operations, you can search for the tables and databases to be restored.

- After the restoration is complete, new tables with timestamps as suffixes are generated in the instance. You can rename the new tables. The new table name must be unique. It can contain up to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.

- **Time-specific table**: The tables to be restored are read from the latest full backup before the selected point in time. **Recent table**: The tables to be restored are read from the current point in time.

- If a full backup is performed before your selected point in time, you can select **Recent table** to view the latest table details.

- If the tables to be restored are not found or are deleted by mistake, you need to log in to the databases and create tables with the same names. Then, the tables to be restored will be displayed when you select **Recent table**.

- Only specified tables are restored. Ensure that all tables to be restored are selected.

Step 7 Click **Next: Confirm**. On the displayed page, confirm the information about the tables to be restored and click **Restore Now**.

If you need to modify your settings, click **Previous**.

Step 8 On the **Instances** page, view the instance status, which is **Restoring**. During the restoration process, services are not interrupted.

You can also view the progress and result of restoring tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the tables as required.

----End

8.5 Restoring a DB Instance from a Cross-region Backup

Scenarios

GaussDB(for MySQL) can store backups in a different region from your DB instance. If your DB instance is faulty, you can use a backup to restore data to a new DB instance in the region where the backup is stored.

Prerequisites

A cross-region backup has been created. For details, see **Configuring a Cross-Region Backup Policy**.
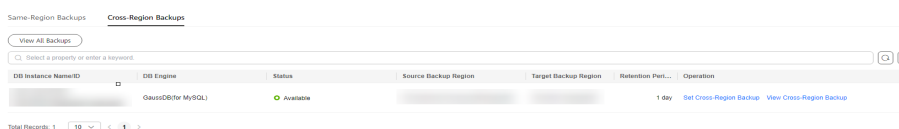
Procedure

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

Step 2 Click ⦿ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the upper left corner of the page, select the region where the backup is located.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click the **Cross-Region Backups** tab.

Step 6 Locate the target DB instance and click **View Cross-Region Backup** in the **Operation** column.

Figure 8-10 Cross-region backups
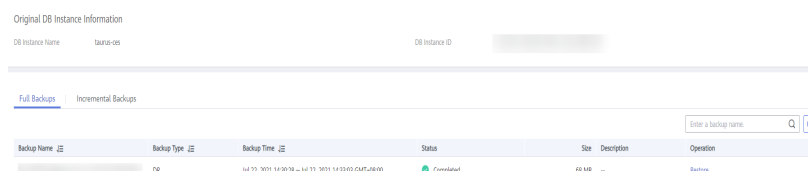
Step 7 On the displayed page, select the backup to be restored.

Currently, full and incremental backups can be restored to new DB instances.

Restoring a full backup

1. Locate the target backup and click **Restore** in the **Operation** column.

    Figure 8-11 Restoring a full backup

2. In the displayed dialog box, confirm instance details and click **OK**.

Figure 8-12 Restoring a full backup to a new DB instance
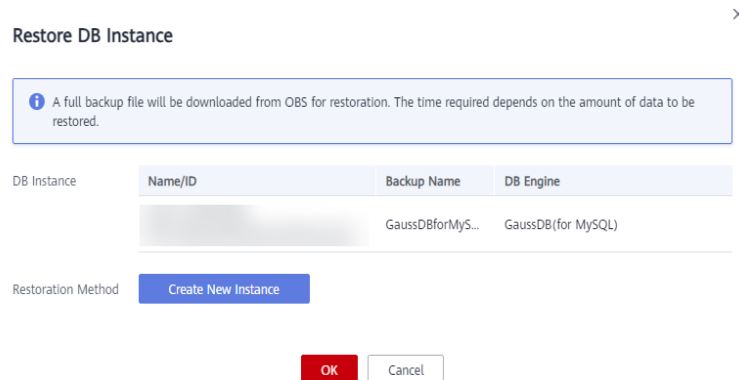
3. On the displayed page, set the parameters of the new DB instance and click **Next**.

- The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.

- The default database port is **3306**.

- Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

Restoring an incremental backup

1. Click **Restore to Point in Time**.

Figure 8-13 Restoring an incremental backup

2. Select the date and time range, and select or enter a time point within the time range.

Figure 8-14 Restoring an incremental backup to a point in time
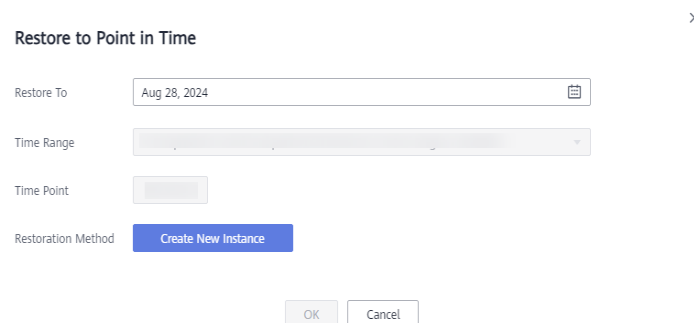
3. Click **OK**.

4. On the displayed page, set the parameters of the new DB instance and click **Next**.

   – The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.

   – The default database port is **3306**.

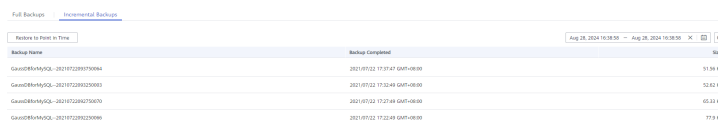   – Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

----End

9 Serverless Instances
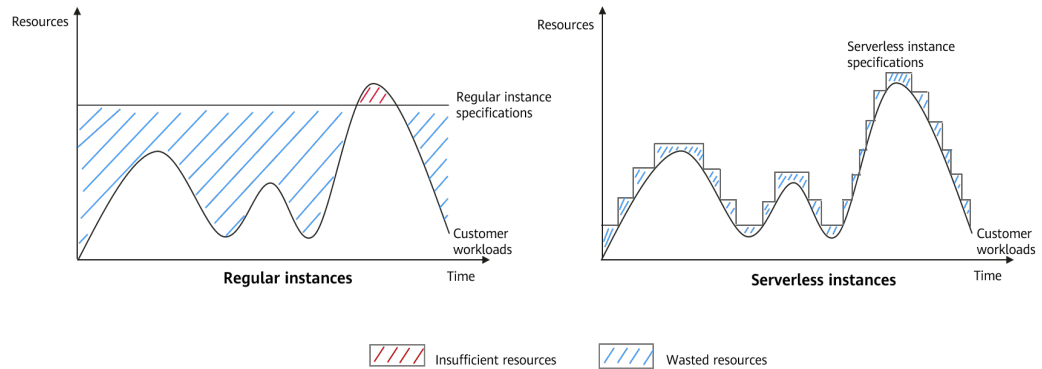
9.1 What Is a Serverless Instance?

Context

The stability and reliability of databases are crucial in modern enterprise-level IT systems. If a database is not stable, the entire system cannot run properly. To solve such an issue, users typically configure various parameters and redundant resources (such as compute, memory, and storage).

However, during off-peak hours, those redundant resources are often left idle, resulting in wasted costs. Even with those configurations, there is still a risk of temporary resource shortages in the face of unexpected surges in workloads, which can compromise the overall system.

In addition to typical enterprise scenarios, there are also many low-frequency, small-scale database usage scenarios in the cloud, such as development and testing environments, mini-program development, and school laboratory teaching environments. In these scenarios, users often have minimal specification requirements but demand workload continuity. Constantly creating or deleting pay-per-use instances is not feasible, and buying low-spec yearly/monthly instances results in a significant waste of money when there are no workloads to process.

To address those concerns, GaussDB(for MySQL) has introduced serverless instances. These instances can dynamically adjust resources based on workloads and are billed on a pay-per-use basis, which reduces costs and improves efficiency. Additionally, serverless instances make it easier for small- and medium-sized enterprises to use cloud databases.

The following figure shows the resource usage and specification changes of regular and serverless instances during significant workload fluctuations.

Figure 9-1 Resource usage and specification changes of regular and serverless instances

As seen in the figure, regular and serverless instances perform differently during significant workload fluctuations.

- Regular instances: Resources are wasted during off-peak hours and insufficient during peak hours, which will affect workloads.

- Serverless instances: The specifications are adjusted based on workload demands to achieve minimal resource wastes. Even during peak hours, workload demands can still be met, ensuring workload continuity and improving system stability.

How a Serverless Instance Works

GaussDB(for MySQL) serverless instances use a write once read many (WORM) architecture and shared storage. They provide the ability to dynamically scale with system workloads. Each instance node can achieve vertical scaling of vCPUs and memory in seconds and horizontal scaling of read replicas, allowing you to quickly and independently change compute capabilities during peak and off-peak hours. This helps you quickly respond to workload changes less expensively and more efficiently.

Figure 9-2 Serverless architecture
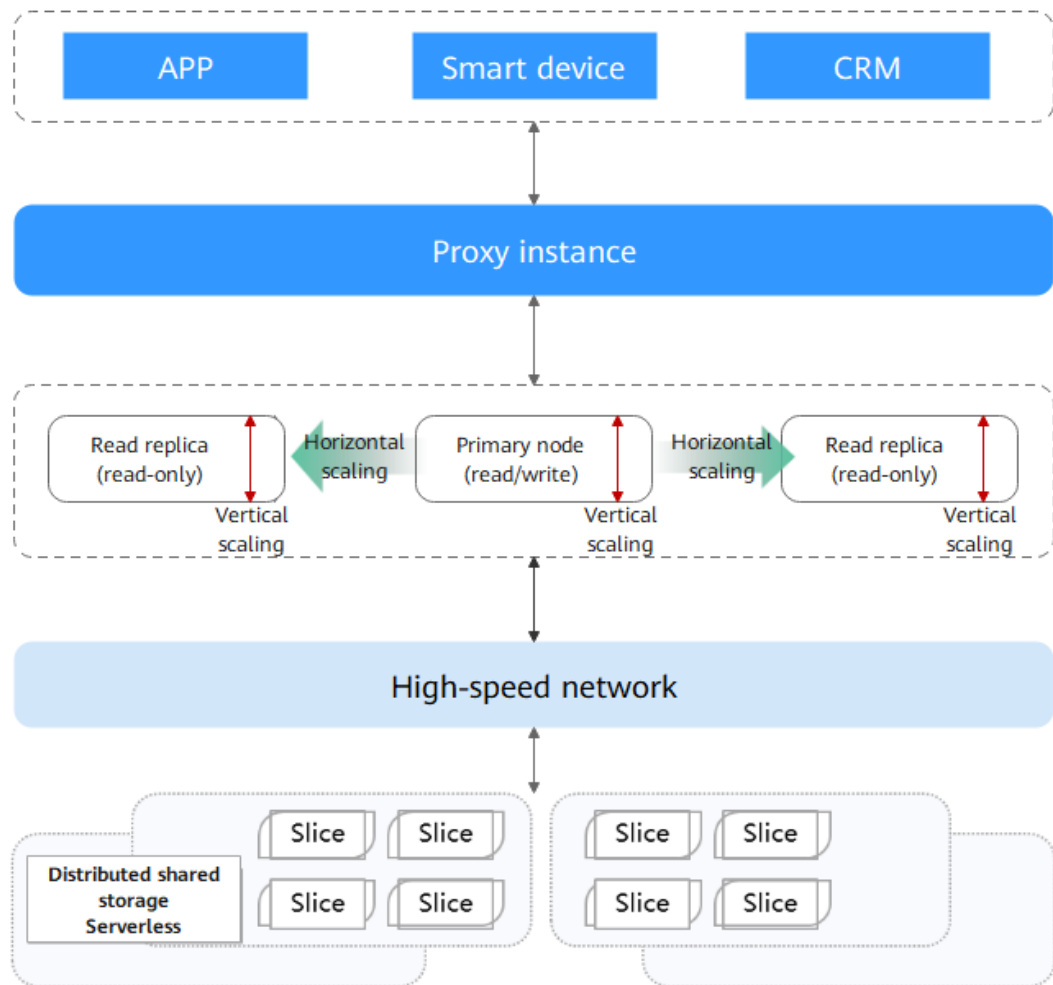
- Both the primary node and read replicas are serverless. They use distributed shared storage and can be scaled based on workload changes.

- The billing unit is TCU. 1 TCU is approximately equal to 1 vCPU and 2 GB of memory. When the primary node or a read replica is scaled, its TCU increases or decreases accordingly.

- When creating a serverless instance, you can specify a TCU range, instead of configuring specific specifications. Then the instance can be scaled based on the CPU usage and memory usage.

  Vertical scaling: The node performance (CPU and memory specifications) changes.

  Cloud Eye monitors the CPU usage and memory usage of serverless instances. If any of the following conditions is met, a scale-up is automatically triggered:

  – The CPU usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up.

  – The memory usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up.

  – The CPU usage is greater than 60% for 20 seconds and it has been at least 10 seconds since the last scale-up.

  If the following condition is met, a scale-down is automatically triggered:

The CPU usage is less than 30% for 15 seconds and it has been at least 15 seconds since the last scale-down.

Horizontal scaling: The number of read replicas changes.

If the compute has already been scaled up as much as possible but the CPU or memory usage still meets a compute scale-up condition, read replicas will be added.

If the compute has already been scaled down as much as possible but the CPU or memory usage still meets a compute scale-down condition, read replicas will be removed.

Billing

For details, see **Serverless Billing**.

Advantages

- Lower cost: GaussDB(for MySQL) serverless instances do not depend on other infrastructure or related services. They can be used right out of the box and provide stable and efficient data access services. You are only billed for the resources you use.

- Larger storage space: The storage space of a serverless instance can reach up to 32,000 GB. It is automatically scaled up based on the data volume of the instance to avoid any impacts on workloads due to insufficient storage resources.

- Auto scaling of compute resources: Compute resources required for read and write operations can be scaled automatically, greatly reducing O&M costs and system risks.

- Fully managed and O&M-free experience: All O&M work, such as specification scaling, storage autoscaling, monitoring and alarms, and intelligent O&M, is completed by Huawei Cloud professional teams. Users will not even notice, and their workloads will not be affected, ensuring continuous availability and a truly O&M-free experience.

Application Scenarios

- Infrequently-used databases in enterprise testing environments and personal development

- Intermittent scheduled tasks, such as data statistics and archiving, school teaching, and R&D

- Unpredictable fluctuations in workloads, such as check-in and edge computing

- O&M-free or fully managed databases

- Lower database usage costs during off-peak hours

9.2 Changing the Compute Range

After **buying a serverless instance**, you can change its compute range. When certain trigger conditions are met, instance compute is automatically changed.

Conditions for Changing Compute

Cloud Eye monitors the CPU usage and memory usage of serverless instances.

If any of the following conditions is met, a compute scale-up is automatically triggered:

- The CPU usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up.

- The memory usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up.

- The CPU usage is greater than 60% for 20 seconds and it has been at least 10 seconds since the last scale-up.

If the following condition is met, a compute scale-down is automatically triggered:

The CPU usage is less than 30% for 15 seconds, it has been at least 15 seconds since the last scale-down, and the memory usage is 80% or less.

Constraints

- As data grows and with daily usage, there may be some cache or memory fragments that cannot be released, leading to high memory usage. If you want to reduce compute to the minimum, you can reboot the instance.

- If resources are insufficient when a compute change is triggered, nodes with the desired specifications will be created on a physical machine that has enough resources. If resources on the primary node are insufficient, there will be a primary/standby failover.

Changing the Compute Range

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

Step 2 Click ⬦ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Serverless Information** area, click **Change** under **Compute Range**.

Figure 9-3 Changing serverless information

Serverless Information

Compute Range	Current Compute	Read Replica Range	Existing Read Replicas
2–8 TCUs Change	2 TCUs	1~4	1

∧ Hide

Step 6 In the displayed dialog box, set **Min. Compute** and **Max. Compute**, and click **OK**.

Figure 9-4 Changing the compute range

Step 7 Check the new compute range in the **Serverless Information** area.

----End

9.3 Changing the Maximum and Minimum Numbers of Read Replicas

After **buying a serverless instance**, you can change the maximum and minimum numbers of read replicas. When certain trigger conditions are met, the number of read replicas of a serverless instance is automatically changed.

Conditions for Changing the Number of Read Replicas

If the compute has already been scaled up as much as possible but the CPU or memory usage still meets a compute scale-up condition, read replicas will be added.

If the compute has already been scaled down as much as possible but the CPU or memory usage still meets a compute scale-down condition, read replicas will be removed.

Constraints

- If database proxy is not enabled for an instance, the number of read replicas cannot be adjusted.

- To adjust the number of read replicas, there must be at least one proxy instance and new nodes will automatically be associated with the proxy instance. To associate new read replicas with a proxy instance, go to the **Database Proxy** page.

- To prevent your workloads from being affected, you are advised not to use the private IP address for read to connect to your application.

- After you set **Scaling Type** to **Number of read replicas**, the routing policy of the proxy instance will be changed to load balancing.

- Manually created read replicas are affected by the configured auto scaling policy. For example, if you set the minimum number of read replicas to 1 and manually create four read replicas, when the CPU or memory usage meets the scale-down conditions, the manually created read replicas will be removed until there is only one read replica.

Changing the Maximum and Minimum Numbers of Read Replicas

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Serverless Information** area, click **Change** under **Compute Range**.

Figure 9-5 Changing serverless information

Step 6 In the displayed dialog box, set **Scaling Type** to **Number of read replicas**, change the maximum and minimum numbers of read replicas, and click **OK**.

Figure 9-6 Changing the maximum and minimum numbers of read replicas

Step 7 Check the new maximum and minimum numbers of read replicas in the **Serverless Information** area.

----End

9.4 Adding Serverless Read Replicas to an Instance with Fixed Specifications

You can add serverless read replicas to a pay-per-use or yearly/monthly instance.

Constraints

- To add serverless read replicas, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

- The first time you create a serverless read replica, you need to initialize the serverless policy. If you need to modify the serverless policy later, access the **Serverless Information** area on the **Basic Information** page.

● Adding serverless read replicas is mutually exclusive with the auto scaling function. If the serverless function has been enabled for an instance with fixed specifications, the auto scaling function cannot be enabled and vice versa.

Procedure

Step 1 **Log in to the management console.**
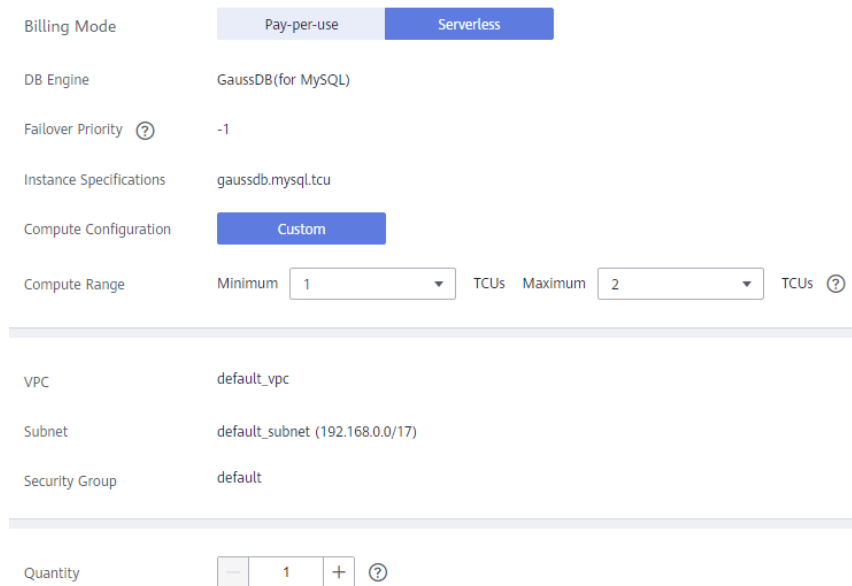
Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to add read replicas to and choose **More** > **Create Read Replica** in the **Operation** column.

Step 5 On the displayed page, select the serverless billing mode and set other parameters.

Figure 9-7 Creating serverless read replicas

Table 9-1 Parameter description

Parameter	Description
Failover Priority	The failover priority of a serverless read replica is fixed to **-1**. During a failover, the serverless read replica will not be promoted to primary. The failover priority cannot be changed.

Parameter	Description
Compute Range	● **Minimum TCUs**: Set the minimum compute. 1 TCU is approximately equal to 1 vCPU and 2 GB of memory. The initial specifications of a new serverless read replica are the minimum compute. ● **Maximum TCUs**: Set the maximum compute.
Quantity	Up to seven serverless read replicas can be created for each instance.

Step 6 Click **Next**.

Step 7 Check the read replica settings.

- If you need to modify the settings, click **Previous**.

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

Step 8 View or manage the created read replicas in the **Node List** area on the **Basic Information** page.

- If you want to modify the serverless scaling policy, see **Changing the Maximum and Minimum Numbers of Read Replicas**.

      📖 NOTE

     – To adjust the number of read replicas, there must be at least one proxy instance and new nodes will automatically be associated with the proxy instance, and there are serverless read replicas. If you require that read replicas can be automatically associated with the proxy instance, go to the **Database Proxy** page.

     – Once you enable the function to adjust the number of serverless read replicas for an instance with fixed specifications, the routing policy of proxy instances will remain unchanged. However, if the routing policy is weighted, an improper weight configuration may render the function invalid. To prevent your workloads from being affected, you are advised to configure the same weight for serverless read replicas or use the load balancing routing policy when adjusting the number of serverless read replicas.

     – Each instance supports a maximum of 15 read replicas. The total number of serverless read replicas and existing non-serverless read replicas in an instance cannot exceed 15, and the number of serverless read replicas cannot exceed 7.

      Examples:

      If there are 13 non-serverless read replicas in an instance, the maximum number of serverless read replicas is 2 instead of 7.

      If there are 5 non-serverless read replicas in an instance, the maximum number of serverless read replicas is 7 instead of 10.

- If you want to delete a serverless read replica, see **Deleting a Read Replica**.

----End

10 Read Replicas

10.1 Introducing Read Replicas

What Are Read Replicas?

In read-intensive scenarios, a primary node may be unable to handle the read pressure and service performance may be affected. To offload read pressure from the primary node, you can create one or more read replicas. These read replicas can process a large number of read requests and increase application throughput. To do this, connection addresses need to be scheduled separately for the primary node and each read replica on your applications so that all read requests can be sent to read replicas and write requests to the primary node.

Billing Standards

Read replicas are billed as well. The billing mode is the same as that of the primary node.

Functions

- You do not need to maintain accounts and databases of read replicas. They are synchronized from the primary node.
- The system can monitor the performance of read replicas.

Constraints

- You can create a maximum of 15 read replicas for a yearly/monthly or pay-per-use instance, and seven read replicas for a serverless instance.
- Read replicas do not support restoration from backups.
- Data cannot be migrated to read replicas.
- You cannot create or delete databases on read replicas.
- You cannot create database accounts for read replicas.
- There may be a latency between the read replicas and the primary node. The latency of the full-text index is significant due to its special mechanism. For

latency-sensitive application workloads, you are advised to send queries to the primary node.

10.2 Adding Read Replicas to a DB Instance

Scenarios

Read replicas of a DB instance are used to enhance instance capabilities and reduce the read pressure on the primary node. After a DB instance is created, you can add read replicas.

There are synchronous and asynchronous read replicas.

- Synchronous read replicas: Their failover priority is 1 and specifications are the same as those of the primary node. To avoid failover failures caused by inconsistent specifications between the primary node and read replicas, a DB instance must have a synchronous read replica, and a multi-AZ DB instance must have a synchronous read replica in a different AZ from the primary node.
- Asynchronous read replicas: Their failover priority is not 1 and specifications are different from those of the primary node.

For more information about read replicas, see **Introducing Read Replicas**.

Deployment Relationships Between the Primary Node and Read Replicas

- If you select single-AZ deployment, read replicas are deployed in the same AZ as the primary node.
- If you select multi-AZ deployment, read replicas are evenly deployed in different AZs to ensure high reliability.

Constraints

- Each yearly/monthly or pay-per-use DB instance has a maximum of 15 read replicas.
- Each serverless DB instance has a maximum of 7 read replicas.
- If all synchronous read replicas are unavailable during a failover, an asynchronous read replica will be promoted to primary.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance you want to add read replicas to and choose **More** > **Create Read Replica** in the **Operation** column.

Step 5 On the displayed page, set required parameters.

Table 10-1 Parameter description

Parameter	Description
Billing Mode	- Pay-per-use DB instance: Pay-per-use and serverless read replicas can be added. - Yearly/Monthly DB instance: Yearly/Monthly, pay-per-use, and serverless read replicas can be added. - Serverless DB instance: Only serverless read replicas can be added.
Failover Priority	Failover priority ranges from 1 for the first priority to 16 for the last priority. This priority determines the order in which read replicas are promoted when recovering from a primary node failure. Read replicas with the same priority have a same probability of being promoted to the new primary node. You can configure a failover priority for up to 9 read replicas, and the default priority for the remaining read replicas is **-1**, indicating these read replicas cannot be promoted to primary. You can change the failover priority of a read replica. **NOTE** - Serverless DB instance: The failover priority for the primary node can only be **1**, while that for a newly added read replica can be 1 to 15. - Yearly/Monthly DB instance: When a pay-per-use or serverless read replica is added, the failover priority is **-1** by default and cannot be changed. - Pay-per-use DB instance: When a serverless read replica is added, the failover priority is **-1** by default and cannot be changed.
AZ	GaussDB(for MySQL) multi-AZ instances allow you to select an AZ when creating a read replica. - If no AZs are specified, the created read replicas are evenly distributed in each AZ. - If too many nodes are created in a specified AZ, the read replicas may fail to be created due to insufficient resources. **NOTE** - To specify AZs, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. - Serverless DB instances do not allow you to specify AZs for read replicas.
Instance Specifications	This parameter is only available for primary/standby DB instances. If the failover priority is set to **1**, the specifications of read replicas must be the same as those of the primary node.
Quantity	A DB instance can contain up to 15 read replicas.

Step 6 For a yearly/monthly instance, click **Next** and select a payment mode.

Step 7 For a pay-per-use instance, click **Next**.

Step 8 Check the read replica settings.

- If you need to modify the settings, click **Previous**.

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

Step 9 View the new read replicas in the **Node List** area of the **Basic Information** page. You can also promote a read replica to primary or delete a read replica.

----End

APIs

- **Creating a Read Replica**
- **Deleting or Unsubscribing from a Read Replica**

10.3 Promoting a Read Replica to Primary

A GaussDB(for MySQL) instance consists of a primary node and multiple read replicas. In addition to **automatic failover** scenarios, you can perform a **manual switchover** to promote a read replica to the new primary node.

Manual Switchover
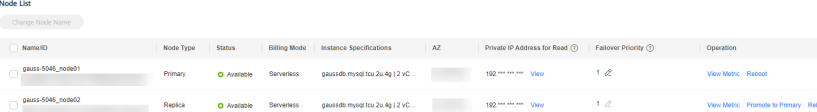
Step 1 **Log in to the management console.**

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, locate the read replica to be promoted and click **Promote to Primary** in the **Operation** column.

Figure 10-1 Promoting a read replica to the new primary node

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

- During a manual switchover, there may be a brief disconnection lasting about 30 seconds. Ensure that your applications support automatic reconnection.

- During a manual switchover, the DB instance status is **Promoting to primary** and this process takes several seconds or minutes.

- After a switchover is complete, the node types of the original primary node and read replica have been exchanged, and the read replica status changes to **Available**.

> **NOTICE**
>
> – A read replica whose failover priority is -**1** cannot be promoted to the primary node.
>
> – Services may be intermittently interrupted for several seconds or minutes when a read replica is promoted to the primary node.
>
> – Promoting a read replica to primary will switch over the private IP addresses for read of the primary node and read replica. To ensure workloads are not interrupted, connect to your DB instance using the private IP address from the **Network Information** area in the **Basic Information** page or the proxy address from the **Database Proxy** page. For details about the differences between the two addresses, see **Description of Each IP Address**.
>
> – To ensure workload continuity, you first **enable Application Lossless and Transparent (ALT)** and then promote a read replica to primary.

----End

Automatic Failover

GaussDB(for MySQL) uses an active-active HA architecture that automatically fails over to a read replica selected by the system.

Each read replica has a failover priority that determines which read replica is promoted if the primary node fails.

- Priorities range from 1 for the highest priority to 16 for the lowest priority.
- If two or more read replicas share the same priority, they have a same probability of being promoted to the new primary node.

GaussDB(for MySQL) selects a read replica and promotes it to the new primary node as follows:

1. Read replicas available for promotion are identified.
2. One or more read replicas with the highest priority are identified.
3. One of the read replicas with the highest priority is selected and promoted. If the promotion fails due to network faults or abnormal replication status, GaussDB(for MySQL) attempts to promote another read replica by priority and repeats the process until a read replica is successfully promoted.

10.4 Deleting a Read Replica

Scenarios

You can delete read replicas billed on a pay-per-use or serverless basis on the **Basic Information** page.

Constraints

- Deleted read replicas cannot be recovered. Exercise caution when performing this operation.

- You can only delete a read replica when the DB instance has two or more read replicas.

- If another operation is being performed on a DB instance, the read replicas of the instance cannot be manually deleted.

- For multi-AZ deployment, make sure that the primary node and remaining read replicas are located in different AZs after a read replica is deleted.

  If a primary node and a read replica are deployed in AZ1 and the other read replica is deployed in AZ2, the read replica in AZ2 cannot be deleted.

- Before deleting the last serverless read replica, ensure that the function for adjusting the number of serverless read replicas has been disabled.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, locate the read replica to be deleted and choose **More** > **Delete** in the **Operation** column.

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

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 7 In the displayed dialog box, click **Yes**. Refresh the **Instances** page later to confirm that the deletion has completed.

----End

APIs

- **Creating a Read Replica**
- **Deleting or Unsubscribing from a Read Replica**

10.5 Unsubscribing from a Read Replica

Scenarios

You can unsubscribe a read replica of a yearly/monthly instance.

Constraints

- You can only unsubscribe a read replica when the DB instance has two or more read replicas.
- Only isolated read replicas can be unsubscribed.
- If a read replica of a DB instance is being isolated, you cannot perform the following operations for the instance:
  - Creating read replicas
  - Scaling up storage space
  - Changing instance specifications
  - Rebooting the instance
  - Resetting the password
  - Upgrading the patch
  - Changing the private IP address
  - Changing the database port
  - Enabling or disabling SSL
  - Binding an EIP
  - Operations related to proxy instances
- The following operations cannot be performed on other read replicas of the instance:
  - Changing a failover priority
  - Promoting a read replica to primary
  - Isolating a read replica

Procedure

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

Step 2 Click ⬚ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, locate a read replica and click **Isolate** in the **Operation** column.

> **NOTICE**
>
> - When a read replica is isolated, you can only unsubscribe or release it.
> - When the workloads are heavy, you can release the isolated read replica if necessary.

Step 6 After the read replica status changes to **Isolated**, choose **More** > **Unsubscribe** in the **Operation** column.

> **NOTE**
>
> - It takes about 1 minute to isolate a read replica.
> - When a read replica is isolated, read operations and database synchronization cannot be performed.
> - To avoid being billed for an isolated read replica, unsubscribe it in a timely manner.

Step 7 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For unsubscription details, see **Unsubscription Rules**.

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

> **NOTICE**
>
> After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.

Step 9 View the unsubscription result. After the order is successfully unsubscribed, the unsubscribed read replica of the instance will be deleted.

----End

APIs

- **Creating a Read Replica**
- **Deleting or Unsubscribing from a Read Replica**

11 Database Proxy (Read/Write Splitting)

11.1 What Is Database Proxy

Database Proxy is a network proxy service between GaussDB(for MySQL) and the application service. It is used to proxy all the requests for the application service to access GaussDB(for MySQL).

Read/write splitting means that read and write requests are automatically forwarded through database proxy addresses. After creating a GaussDB(for MySQL) instance, you can **create a proxy instance**. Write requests are automatically forwarded to the primary node and read requests are forwarded to each node based on the routing policy of the proxy instance.

Basic Concepts

- Proxy address

  After buying a proxy instance, you can view the proxy address on the **Database Proxy** page. The proxy instance sends write requests to the primary node and read requests to read replicas through this address.

- Proxy mode

  There are read/write and read-only proxy modes.

  Read/Write: All write requests are routed only to the primary node, and all read requests are routed to the selected nodes based on the read weights or active connections.

  Read-only: All read requests are routed to the selected read replicas based on the read weights or active connections. The read requests will not be routed to the primary node.

- Transaction splitting

  With transaction splitting enabled for a proxy instance, the proxy instance can route read requests prior to write operations in a transaction to read replicas, reducing the load on the primary node.

  For more information about transaction splitting, see **Enabling Transaction Splitting for a Proxy Instance**.

- Connection pool

  Proxy instances provide session-level connection pools, which help reduce the database load caused by frequent establishment of short connections.

  For more information about connection pools, see **Enabling the Connection Pool for a Proxy Instance**.

- Routing policy

  Proxy instances support weighted and load balancing routing policies.

  - **Weighted**: Read requests are assigned to nodes based on the weights you specify.

  - **Load balancing**: Read requests are assigned to nodes with fewer active connections. In the load balancing policy, you do not need to configure the weights of nodes.

  For more information about routing policies, see **Modifying the Routing Policy of a Proxy Instance**.

How Read/Write Splitting Works

You can create one or more proxy instances for your GaussDB(for MySQL) instance to enable read/write splitting.

- Single proxy instance

  If your GaussDB(for MySQL) instance has only one proxy instance, applications connect to the proxy instance through the proxy address. Write requests are forwarded to the primary node and read requests to the primary node or read replicas based on the **routing policy** you specify.

  Figure 11-1 Read/write splitting with only one proxy instance
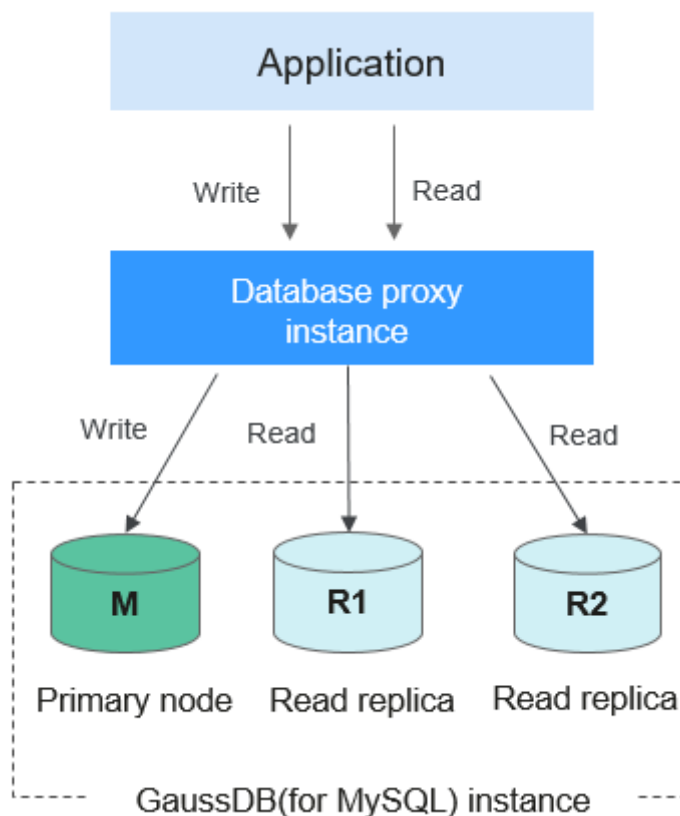
- Multiple proxy instances

  To isolate workloads from different applications, you can create up to four proxy instances for your GaussDB(for MySQL) instance. Different applications can connect to different proxy instances as required. The associated read replicas of the proxy instances process read requests from different applications for workload isolation.

Figure 11-2 Read/write splitting with multiple proxy instances
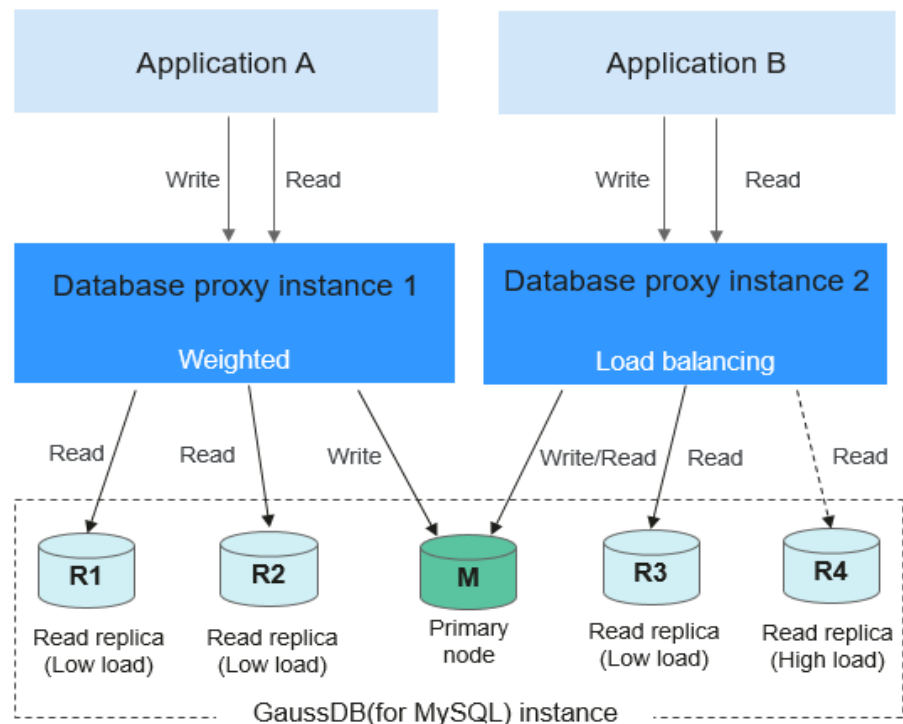
Application Scenarios

- The primary node is overloaded due to a large number of requests in a transaction.
- The primary node is overloaded due to excessive connections.
- Read/write splitting is required.

Read/Write Splitting Advantages

- Compared with manual read/write splitting in applications, the read/write splitting using proxy addresses features flexible scale-out and low maintenance costs.
- Read requests are distributed to your read replicas based on weights to balance your database traffic and improve resource utilization.
- A proxy instance routes read requests of an application only to its associated read replicas to isolate workloads.
- By default, proxy instances provide overload protection to prevent server OOM (out of memory) due to heavy pressure when you perform operations on large result sets. This function is enabled by default and does not need to

be configured separately. The pressure caused by the slow kernel depends on flow control.

Request Routing Rules

- **Write requests sent only to the primary node**
  - INSERT, UPDATE, and DELETE
  - All DDL operations (such as table/database creation, table/database deletion, table structure change, and permission change)
  - All requests in transactions (But if transaction splitting is enabled, some read requests in transactions may be sent to read replicas. For details, see **Enabling Transaction Splitting for a Proxy Instance**.)
  - User-defined functions
  - Stored procedures
  - EXECUTE statements
  - Multi-statement requests
  - Requests that use temporary tables
  - All changes to user variables
  - KILL in SQL statements (not command KILL)

- **Read requests sent only to the primary node**
  - If query statements are in transactions, the transaction requests are routed to the primary node. If **SET AUTOCOMMIT=0** is added before a query statement, the transaction requests are routed to the primary node.
  - If all read replicas are abnormal or the read weights allocated to the read replicas are 0, requests will be routed to the primary node. You can set read weights for the primary node and read replicas after read/write splitting is enabled.
  - When running SQL statements:
    - If multi-statements (for example, **insert xxx;select xxx**) are executed, all subsequent requests will be routed to the primary node. To restore read/write splitting, disconnect your application from your instance and then connect it back again.
    - Read operations with locks (for example, **SELECT for UPDATE**) will be routed to the primary node.
    - When **/*FORCE_MASTER*/** is used, requests will be routed to the primary node.
    - If **the HANDLER statement** is executed, all subsequent requests will be routed to the primary node. To restore read/write splitting, disconnect your application from your instance and then connect it back again.
  - SELECT last_insert_id()
  - All queries of user variables

- **Requests sent either to the primary node or a read replica**
  - SELECT not in a transaction

- The COM_STMT_EXECUTE command
- **Requests always sent to all nodes**
  - Changes to all system variables
  - The USE command

Read/Write Attribute Processing Logic

There are read-only and read/write modes for proxy instances. The read/write attribute processing logic varies depending on the proxy mode.

Proxy Mode	Routing Policy	Weight of Primary Node	Normal Case	All Read Replicas Are Faulty
Read-only	Weighted Load balancing	Not configurable	The primary node does not process read-only requests. Proxy address: readable but not writable	The primary node does not process read-only requests. Proxy address: connection error
Read/Write	Load balancing	Assigned by system	Primary node: readable and writable Proxy address: readable and writable	Primary node: readable and writable Proxy address: readable and writable
	Weighted	> 0	Primary node: readable and writable Proxy address: readable and writable	Primary node: readable and writable Proxy address: readable and writable
		= 0	Primary node: not readable but writable Proxy address: readable and writable	Primary node: readable and writable Proxy address: readable and writable

Billing

Proxy instances are free.

Precautions

Table 11-1 Precautions for proxy instances

Category	Precaution
Version constraints	- If the kernel version of your GaussDB(for MySQL) instance is one of the following, proxy instances cannot be created: – From 2.0.26.2 to 2.0.28.3 – 2.0.29.1 - If the kernel version of your GaussDB(for MySQL) instance is earlier than 2.0.42.230601, only one proxy instance can be created. - If the kernel version of your GaussDB(for MySQL) instance is 2.0.42.230601 or later, up to four proxy instances can be created.
Unsupported functions	- Proxy instances do not support compression protocols. - Proxy instances do not support the READ-UNCOMMITTED transaction isolation level. - Proxy instances do not support reads from and writes to any column containing more than 16 MB of data in a table. - Database proxies do not support the SQL mode parameter **PAD_CHAR_TO_FULL_LENGTH**.

Category	Precaution
Usage constraints	- To create a proxy instance, a GaussDB(for MySQL) instance must have at least 8 vCPUs. - Read/write splitting can be enabled only when at least one read replica is created. - After read/write splitting is enabled, the database port and private IP address of your GaussDB(for MySQL) instance cannot be changed. - If **multi-statements** are executed, all subsequent requests will be routed to the primary node. To restore the read/write splitting function, disconnect the connection from your applications and establish a connection again. - When a proxy address is used, all transaction requests are routed to the primary node (you can use transaction splitting to route read requests prior to write operations in a transaction to read replicas). The non-transaction read consistency is not ensured. To ensure read consistency, encapsulate the read requests into a transaction. - When a proxy address is used, you can run **show processlist** command on the proxy instance or GaussDB(for MySQL) instance. If **show processlist** is executed on a proxy instance, only the services delivered through proxy nodes are displayed. - If a proxy node is abnormal, running **show processlist** or **Kill** on the proxy instance may take a long time, but services are not affected. - After a proxy node is deleted, services on the deleted proxy node may be displayed when **show processlist** is executed on the proxy instance. - If **Kill** is executed on the proxy instance, error information such as timeout may be displayed occasionally. You can run **show processlist** again to check whether the services are killed successfully. - If a proxy node is abnormal, there may be frame freezing for 2 seconds when you run **show processlist** on the proxy instance. The results will still be returned. - When a proxy instance is used, the size of a concatenate SQL statement cannot exceed 100 MB to prevent statement parsing from consuming too many resources.
HTAP analysis	- Consistency levels and connection pools are not supported. - Only the weighted routing policy is supported. - Only the read/write proxy mode is supported.

11.2 How to Use a Proxy Instance to Enable Read/Write Splitting

After creating a GaussDB(for MySQL) instance, you can create a proxy instance. With the proxy address, write requests are automatically forwarded to the primary node, and read requests are forwarded to each node based on the routing policy of the proxy instance to offload read pressure from the primary node.

This section describes how to use a proxy instance to enable read/write splitting.

Step 1: Create a Proxy Instance

Step 2: Perform User Authentication

Step 3: Check Security Group Rules

Step 4: Use the Proxy Address to Connect to Your GaussDB(for MySQL) Instance

Step 5: Verify Read/Write Splitting

Constraints

Before creating a proxy instance, you need to ensure that:

- You have purchased a GaussDB(for MySQL) instance.
- You have been understood precautions. For details, see **Precautions**.

Step 1: Create a Proxy Instance

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

Step 2 Click 📍 in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click **Create Proxy Instance**.

Step 7 In the displayed dialog box, configure related parameters.

Table 11-2 Parameter description

Parameter	Description
Proxy Instance Name	The name can consist of 4 to 64 characters and must start with a letter. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Proxy Mode	You can select **Read/Write** or **Read-only** as required. ● **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights or active connections. The default read weight of a node is **100**. ● **Read-only**: Write requests are not forwarded to any node. All read requests are forwarded to the selected read replicas based on the read weights or active connections. The read requests are not forwarded to the primary node, even if the primary node is selected. **NOTE** – In the read-only mode, only read requests are supported. If write requests are forwarded to the selected nodes, an error message is displayed. – DDL, DML, and temporary table operations are not supported in the read-only mode.
Consistency Level	The consistency level can be configured only when the kernel version of your GaussDB(for MySQL) instance is 2.0.28.1 or later. Value: ● **Eventual consistency** After a proxy instance is created, requests for SELECT operations are routed to different nodes based on their read weights. Because there is a replication latency between the primary node and each read replica and the replication latency varies for different read replicas, the result returned by each SELECT statement may be different when you repeatedly execute a SELECT statement within a session. In this case, only eventual consistency is ensured. To offload read requests from the primary node to read replicas, you can select eventual consistency. ● **Session consistency** To eliminate data inconsistencies caused by eventual consistency, session consistency is provided. Session consistency ensures the result returned by each SELECT statement in a session is the data that was updated after the last write request. To use session consistency, the kernel version of your proxy instance must be 2.7.4.0 or later.

Parameter	Description
Routing Policy	Value: ● **Weighted**: Read requests are assigned to nodes based on the weights you specify. ● **Load balancing**: Read requests are assigned to nodes with fewer active connections. To use load balancing, the kernel version of your proxy instance must be 2.22.07.000 or later. For more information about routing policies, see **Modifying the Routing Policy of a Proxy Instance**.
Proxy Instance Specifications	You can select the proxy instance specifications as needed. ● Kunpeng general computing-plus: 2 vCPUs | 4 GB, 4 vCPUs | 8 GB, and 8 vCPUs | 16 GB ● General-enhanced: 2 vCPUs | 4 GB, 4 vCPUs | 8 GB, and 8 vCPUs | 16 GB
Subnet	To specify this parameter, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. When creating a proxy instance, you can specify a subnet for the proxy instance. If the subnet where the GaussDB(for MySQL) instance is associated with is a secondary CIDR block, you can only select the same subnet as the GaussDB(for MySQL) instance for the proxy instance.
Proxy Instance Nodes	You can enter an integer from 2 to 16. The default value is **2**. Number of recommended proxy instance nodes = (Number of vCPUs of the primary node + Total number of vCPUs of all read replicas)/(4 x Number of vCPUs of the proxy instance), rounded up.
Associate New Nodes	After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the proxy instance.
New Node Weight	If **Routing Policy** is **Weighted**, you need to set read weights of the new nodes. The default weight of a node is **100**. Nodes with higher weights process more read requests.

Parameter	Description
Database Nodes	You need to select the nodes to be associated with the proxy instance for processing read requests. ● If **Routing Policy** is **Load balancing**, you do not need to configure read weights for selected nodes. Read requests are forwarded to nodes with fewer active connections. ● If **Routing Policy** is **Weighted**, you need to configure read weights of the primary node and read replicas. Read requests are forwarded to nodes based on the weights you specify. For example, read weights assigned to one primary node and two read replicas are 100, 200, and 200, respectively. In the read/write mode, the primary node and two read replicas process read requests in the ratio of 1:2:2. The primary node processes 20% of read requests, and each read replica processes 40% of read requests. Write requests are automatically routed to the primary node. In the read-only mode, the read weight of the primary node does not take effect, and the two read replicas process 50% of read requests, respectively.

Step 8 Click **OK**.

Step 9 View the proxy instance and associated nodes.

After the proxy instance creation is complete, you can view the created proxy instance on the **Database Proxy** page.

Figure 11-3 Viewing the create proxy instance
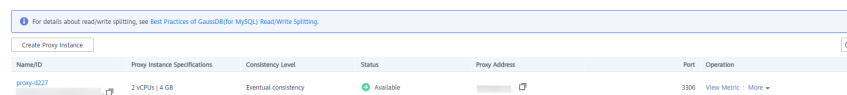
Click **Basic Information** in the navigation pane. Click ••• in the upper right corner of the page and select **View Instance Topology**. In the instance topology, you can view the database nodes associated with the proxy instance. You can move the pointer to a node name to view its details.

Figure 11-4 Viewing information about nodes associated with a proxy instance
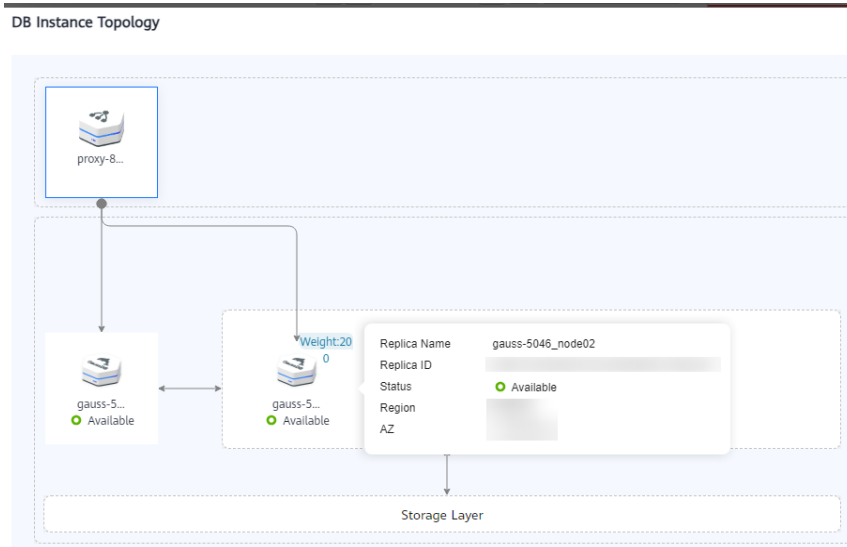
----End

Step 2: Perform User Authentication

Before using a proxy instance to connect to your GaussDB(for MySQL) instance, ensure that the current database account has the permissions to access the proxy address, or the proxy instance cannot connect to your GaussDB(for MySQL) instance.

You can perform the following steps to check the permissions and grant the account the permissions to access the proxy address.

Step 1 Connect to your GaussDB(for MySQL) instance.

For details, see **Connecting to a DB Instance**.

Step 2 After the instance is connected, run the following SQL statement to check whether the host of the current database account contains a proxy address:
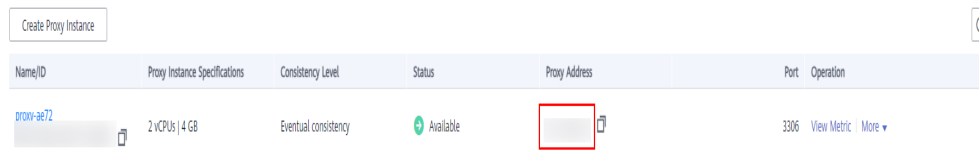
SELECT user,host FROM mysql.user;

To obtain the proxy address:

Click the name of the GaussDB(for MySQL) instance to go to the **Basic Information** page. In the navigation pane, choose **Database Proxy**. In the proxy instance list, view the proxy address.

Figure 11-5 Viewing a proxy address

Step 3 If the host does not contain the CIDR block where the proxy instance is associated with, assign the remote access permissions to the host.

For example, if you want to connect to the GaussDB(for MySQL) instance using 192.168.0 as user **root**, set **Host** to **192.168.%** on the DAS user management page. For details, see **Editing a User**.

Figure 11-6 Configuring a host IP address

----End

Step 3: Check Security Group Rules

You need to ensure that the inbound and outbound rules allow access from the proxy address. The default port number is 3306.

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click the security group name in the **Security Group** field.

Step 6 On the **Inbound Rules** tab, check whether access through port **3306** is allowed by default.

Figure 11-7 Allowing access through port 3306

If there is no such a rule, click **Fast-Add Rule**. In the displayed dialog box, select **MySQL (3306)** and click **OK**.

Figure 11-8 Fast adding port 3306

----End

Step 4: Use the Proxy Address to Connect to Your GaussDB(for MySQL) Instance

Step 1 View the proxy address and port on the GaussDB(for MySQL) console.

Click the name of the GaussDB(for MySQL) instance to go to the **Basic Information** page. In the navigation pane, choose **Database Proxy**. In the proxy instance list, view the proxy address and port.

Figure 11-9 Viewing a proxy address and port
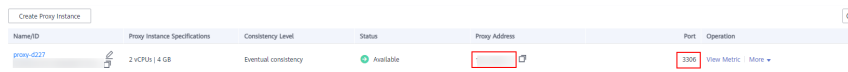
Step 2 Log in to an ECS.

For details, see *Elastic Cloud Server User Guide*.

Step 3 Run the following command to connect to the GaussDB(for MySQL) instance using the proxy address:

mysql -h *<hostIP>* **-P** *<port>* **-u** *<userName>* **-p** *<password>*

Table 11-3 Parameter description

Parameter	Description
<hostIP>	Proxy address obtained in **Step 1**.
<port>	Port obtained in **Step 1**.
<userName>	Username, that is, the GaussDB(for MySQL) database administrator account. The default value is **root**.
<password>	Password of the GaussDB(for MySQL) database administrator.

----End

Step 5: Verify Read/Write Splitting

After each read operation is complete, you can run the **show last route** command to view the routing result of the read operation.

The following is an example.

Step 1 After the GaussDB(for MySQL) instance is connected, perform a read operation.

Example: **select 1;**

Step 2 Run the following command to view the routing result of the read operation in **Step 1**:

show last route

Figure 11-10 Viewing a query result

☐ **NOTE**

Do not use **show last route** for service code or multi-statement execution.

----End

APIs

- **Creating a Proxy Instance**
- **Querying Proxy Instances**
- **Querying Proxy Instance Specifications**
- **Deleting a Proxy Instance**

11.3 Changing Configurations of a Proxy Instance

11.3.1 Changing the Consistency Level of a Proxy Instance

You can configure a consistency level when **creating a proxy instance** or change the consistency level of an existing proxy instance.

This section describes how to change the consistency level of a proxy instance.

Introducing Consistency Levels

There are several consistency levels to meet requirements in different scenarios.

- Eventual consistency (default)

  After a proxy instance is created, requests for SELECT operations are routed to different nodes based on their read weights. Because there is a replication latency between the primary node and each read replica and the replication latency varies for different read replicas, the result returned by each SELECT statement may be different when you repeatedly execute a SELECT statement within a session. In this case, only eventual consistency is ensured.

- Session consistency

  To eliminate data inconsistencies caused by eventual consistency, session consistency is provided. Session consistency ensures the result returned by each SELECT statement in a session is the data that was updated after the last write request.

  Proxy instances record the log sequence number (LSN) of each node and session. When data in a session is updated, a proxy instance records the LSN of the primary node as a session LSN. When a read request arrives subsequently, the proxy instance compares the session LSN with the LSN of each node and routes the request to a node whose LSN is at least equal to the session LSN. This ensures session consistency.

  Figure 11-11 Principle of session consistency
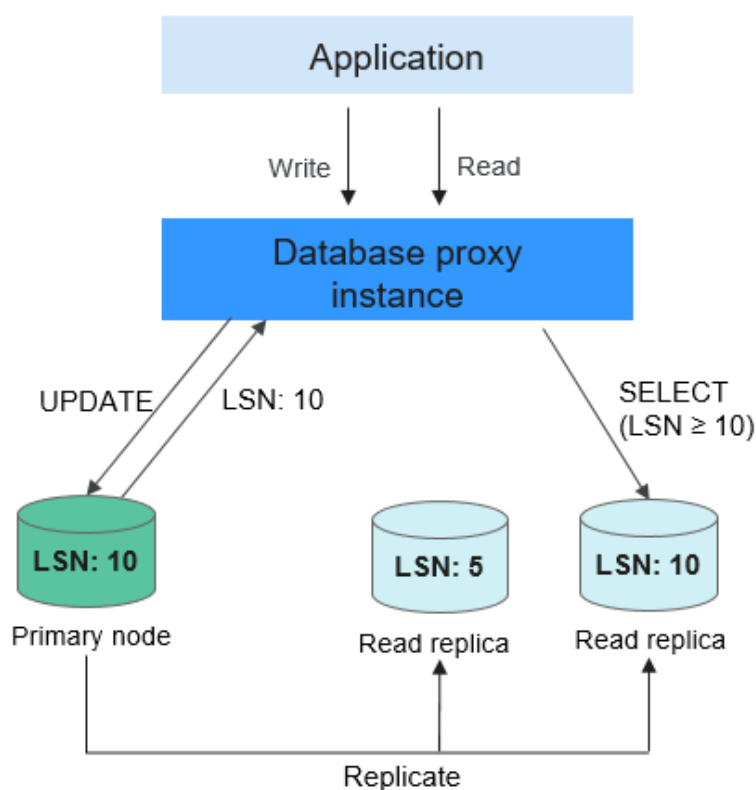
  ⬚ **NOTE**

  In session consistency, if there is significant replication latency between the primary node and read replicas and the LSN of each read replica is smaller than the session LSN, requests for SELECT operations will be routed to the primary node. In this case, loads on the primary node are heavy and instance performance suffers.

Constraints

- To use session consistency, the kernel versions of GaussDB(for MySQL) instances must be 2.0.28.1 or later, and the kernel versions of proxy instances must be 2.7.4.0 or later.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the proxy instance name to go to the **Basic Information** page. In the **Proxy Instance Information** area, click ✎ next to **Consistency Level**.

Figure 11-12 Changing a consistency level

Proxy Instance Information			
Proxy Instance Name	proxy-d227 ✎ ⧉	Proxy Instance ID	⧉
DB Engine Version	2.24.06.000 Upgrade	Proxy Mode	Read/Write
Consistency Level ⑦	Eventual consistency ✎	Status	● Available
Specifications	2 vCPUs | 4 GB Change	Access Control ⑦	Configure
Proxy Instance Nodes	2 Change	Proxy Address	Change
Proxy Port ⑦	3306 ✎	Transaction Splitting ⑦	
Connection Pool ⑦	Disabled Configure	Routing Policy ⑦	Weighted Configure
Subnet	subnet-rds (192.168.0.0/24)	Associate New Nodes	

Step 7 Select a consistency level and click ✓ .

☐ NOTE

After the consistency level is changed, you need to manually reboot the proxy instance or re-establish the connection to the proxy instance on the management console.

For details about how to reboot a proxy instance, see **Rebooting a Proxy Instance**.

----End

APIs

Changing Session Consistency of a Proxy Instance

11.3.2 Enabling the Connection Pool for a Proxy Instance

A session-level connection pool helps reduce the database load caused by frequent establishment of short connections.

Connection Pool is disabled by default. You can enable a session-level connection pool.

A session-level connection pool is suitable for short connections. When your client disconnects from your database, the system checks whether the connection is idle. If it is, the system places the connection in the connection pool of a proxy instance and retains the connection for a short period of time. When your client re-initiates a connection, any available connection in the connection pool is used, reducing

the overhead of establishing a new connection to the database. If no connections are available in the connection pool, a new connection will be established.

Constraints

To use a connection pool, the kernel versions of proxy instances must be 2.22.07.000 or later.

Procedure

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

Step 2 Click ⚲ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click **Change** next to **Connection Pool**.

Step 8 Set **Connection Pool** to **Session level** and click **OK**.

Figure 11-13 Configuring a connection pool

Set Connection Pool   ×

Connection Pool   ○ Disable   ● Session level

                OK   Cancel

----End
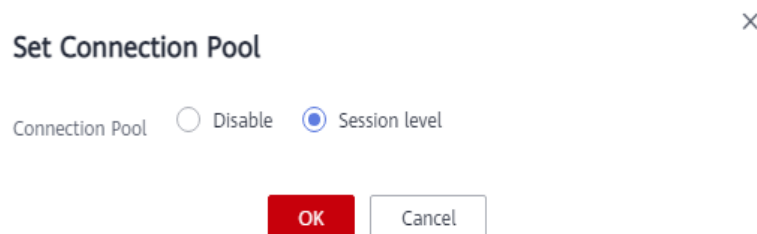
APIs

- **Changing the Connection Pool Type of a Proxy Instance**
- **Querying Proxy Instances**
- **Querying Proxy Instance Specifications**

11.3.3 Enabling Transaction Splitting for a Proxy Instance

In most cases, a proxy instance sends all requests in transactions to the primary node to ensure transaction correctness. However, in some frameworks, all requests are encapsulated into transactions that are not automatically committed using **set autocommit=0**. This causes heavy load on the primary node.

With transaction splitting enabled for a proxy instance, the proxy instance can route read requests prior to write operations in a transaction to read replicas, reducing the load on the primary node.

Transaction splitting is disabled by default. After transaction splitting is enabled and **autocommit** is set to **0**, GaussDB(for MySQL) starts a transaction only for write requests. Before the transaction starts, read requests are routed to read replicas through load balancers.

Constraints

- The kernel versions of proxy instances must be 2.3.9.5 or later.
- Transaction isolation levels of GaussDB(for MySQL) instances must be READ UNCOMMITTED or READ COMMITTED. The default isolation level is REPEATABLE READ.
- Proxy instances must be in the read/write mode.
- After transaction splitting is enabled, the transaction isolation level can only be changed to READ UNCOMMITTED or READ COMMITTED. To change the isolation level to a higher level, disable transaction splitting first.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click next to **Transaction Splitting**.

Figure 11-14 Configuring transaction splitting

Proxy Instance Information			
Proxy Instance Name	proxy-d227	Proxy Instance ID	
DB Engine Version	2.24.06.000 Upgrade	Proxy Mode	Read/Write
Consistency Level	Eventual consistency	Status	Available
Specifications	2 vCPUs | 4 GB Change	Access Control	Configure
Proxy Instance Nodes	2 Change	Proxy Address	Change
Proxy Port	3306	Transaction Splitting	
Connection Pool	Disabled Configure	Routing Policy	Weighted Configure
Subnet	subnet-rds (192.168.0.0/24)	Associate New Nodes	

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

📖 **NOTE**

- To disable transaction splitting, click 🔵 .
- Transaction splitting takes effect only for new connections established after this function is enabled or disabled.

----End

APIs

- **Enabling or Disabling Transaction Splitting of a Proxy Instance**
- **Querying Proxy Instances**
- **Querying Proxy Instance Specifications**
- **Deleting a Proxy Instance**

11.3.4 Modifying the Routing Policy of a Proxy Instance

You can configure the routing policy when **creating a proxy instance**. The default routing policy is weighted. You can also change the routing policy of an existing instance.

Working Principles of the Routing Policy

There are weighted and load balancing routing policies.

- **Weighted**: Read requests are assigned to nodes based on the weights you specify.
- **Load balancing**: Read requests are assigned to nodes with fewer active connections. In the load balancing policy, you do not need to configure the weights of nodes.

Figure 11-15 Working principles of the routing policy
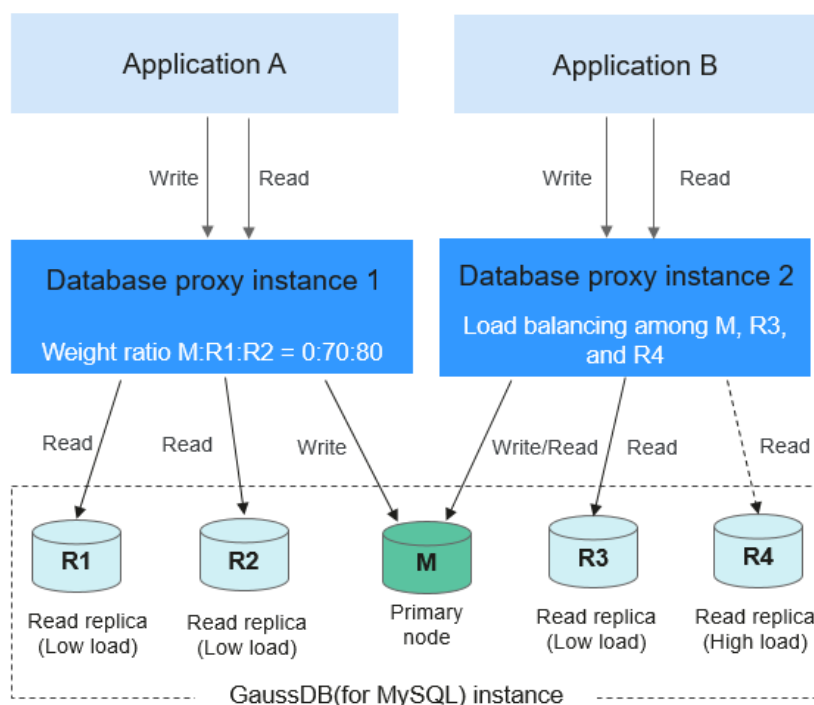
Example:

As shown in the preceding figure, a GaussDB(for MySQL) instance contains one primary node and four read replicas.

For the database proxy instance 1, the routing policy is weighted and the selected nodes include the primary node, read replica R1, and read replica R2, with their read weight ratio of 0:70:80. The write requests of the Application A are automatically forwarded to the primary node through the proxy instance, and the read requests are routed to read replicas R1 and R2 in the ratio of 7:8.

For the database proxy instance 2, the routing policy is load balancing and the selected nodes include the primary node, read replica R3, and read replica R4. The proxy instance determines the node to which the read requests are forwarded based on the number of real-time active connections.

When there are many active connections in read replica R4, the proxy instance forwards most read requests to read replica R3 and the primary node to offload the pressure of read replica R4.

Constraints

- To use the load balancing policy, the kernel versions of proxy instances must be 2.22.07.000 or later. To upgrade a kernel version, see **Upgrading the Kernel Version of a Proxy Instance**.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click **Configure** next to **Routing Policy**.

Step 8 In the displayed dialog box, configure required parameters.

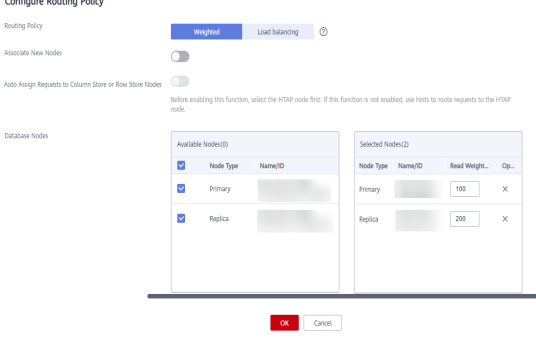**Figure 11-16** Changing the routing policy of a proxy instance
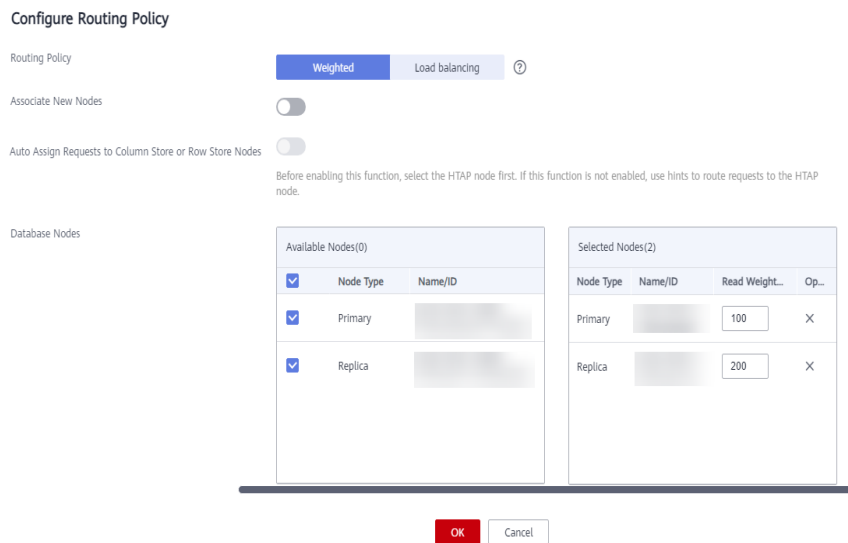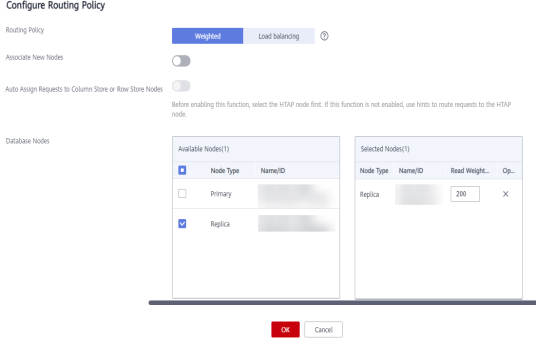
Table 11-4 Parameter description

Parameter	Description
Routing Policy	• **Weighted**: Read requests are assigned to nodes based on the weights you specify. • **Load balancing**: Read requests are assigned to nodes with fewer active connections. In the load balancing policy, you do not need to configure the weights of nodes.
Associate New Nodes	After this function is enabled, new read replicas will be automatically associated with the current proxy instance. If **Routing Policy** is **Weighted**, you need to configure read weights for the new nodes. The default weight of a node is 100. Nodes with higher weights process more read requests.

Parameter	Description
Database Nodes	The proxy mode of a proxy instance determines which nodes read requests are assigned to. ● Read-only mode: All read requests are assigned to the selected, but not to the primary node. **Figure 11-17** Read-only mode ● Read/write mode: All read requests are assigned to the selected nodes (including the primary node and read replicas) based on the routing policy. **Figure 11-18** Read/Write mode

----End

APIs

[Changing the Routing Policy of a Proxy Instance](#)

11.3.5 Changing Read Weights of Nodes

After a proxy instance is created, you can modify the read weights of its associated nodes. Read requests are forwarded to each node based on the read weights you specify, enabling read/write splitting and reducing the load of the primary node.

Constraints

- The routing policy of proxy instances must be weighted.
- You can configure read weights for both the primary node and read replicas.
- The default read weight of the primary node is 0. The higher read weight the primary node is assigned, the more read requests it can process.
- When the read weights of all nodes are 0, services are not affected. In this case, the primary node processes all read and write requests by default.
- The weight of a read replica ranges from 0 to 1000.
- After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the current proxy instance. The default read weight of any new node is 100.
- After a read replica is deleted, its weight is automatically removed while the weights of other read replicas remain unchanged.

Procedure

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

Step 2 Click 📍 in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy** and click the name of a proxy instance to go to the **Basic Information** page.

Step 6 On the **Basic Information** page, click **Configure** next to **Routing Policy**.

Step 7 In the displayed box, configure **Database Nodes**.

1. In the **Available Nodes** area on the left, select the nodes that you want associate with the current proxy instance or deselect the nodes that you want remove from the current proxy instance.

2. In the **Read Weight** column of the **Selected Nodes** area on the right, configure read weights for nodes.

Figure 11-19 Configuring read weights
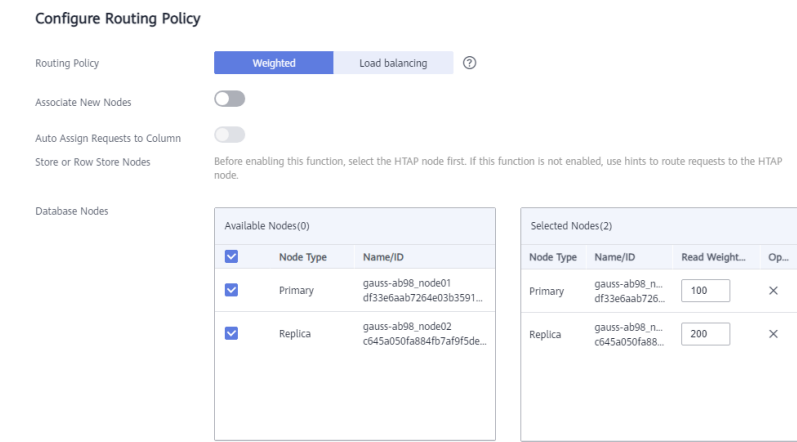
Example:

As shown in **Figure 11-20**, one GaussDB(for MySQL) instance has one primary node and three read replicas. Two proxy instances have been created and they both use the weighted routing policy.

- Proxy instance 1 is in the read/write mode. The primary node and read replica 1 are associated with proxy instance 1 and assigned with a read weight of 100 and 200, respectively. They process read requests in the ratio of 1:2, that is, the primary node processes 1/3 read requests and read replica 1 processes 2/3 read requests. Write requests are automatically routed to the primary node.

- Proxy instance 2 is in the read-only mode. Read replica 2 and read replica 3 are associated with proxy instance 2 and assigned with a read weight of 100 and 200, respectively. Read replica 2 and read replica 3 process read requests in the ratio of 1:2, that is, read replica 2 processes 1/3 read requests, and read replica 3 processes 2/3 read requests.

Figure 11-20 Read/Write splitting in multi-proxy scenarios (weighted routing policy)
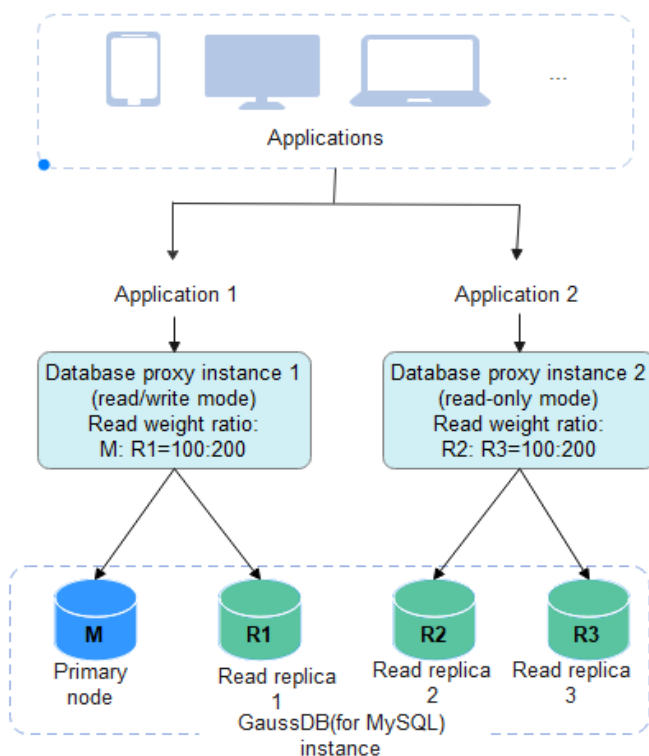
----End

APIs

Assigning Read Weights

11.3.6 Modifying the Multi-statement Processing Mode of a Proxy Instance

When you enable **multi-statement execution** for a proxy instance, you can set multi-statement processing mode to **Strict**, **Loose**, or **Parse**.

- **Strict** (default)

  If a request containing multiple statements is routed to the primary node, the subsequent requests are all routed to the primary node. Read/write splitting can be restored only after you disconnect the current connection and reconnect it.

  Your proxy instances will not parse these statements, so the performance is better. It is suitable for short connections.

- **Loose**

  If a request containing multiple statements is routed to the primary node, the subsequent requests of the current connection can still be routed to the primary node or read replicas.

  Your proxy instances will not parse these statements, so the performance is better. It is good for when multiple statements contain only DML SQL

statements and do not contain operations like setting session variables, creating temporary tables, creating stored procedures, or executing uncommitted transactions.

- **Parse**

  If a request containing multiple statements is routed to the primary node, your proxy instance parses these statements and determines whether to restore read/write splitting for subsequent requests of the current connection based on the operations in the SQL statements. For details about operations in SQL statements, see **• Parse-based mode description:If multi-statements contain the operations listed here, all subsequent requests are routed to the primary node. To restore read/write splitting, you need to disconnect the connection and then re-establish it.Creating temporary tablesCreating stored proceduresExecuting uncommitted transactions (for example, begin is executed but commit or rollback is not executed)Executing complex or special syntax. These statements will not be parsed.**.

  Parsing statements affects the proxy instance performance. The degree of the impact depends on the length and complexity of statements. It is recommended that the statements be less than 100 MB.

Constraints

- To configure the multi-statement processing mode on the management console, contact customer service.
- The changed multi-statement processing mode applies to your proxy instance immediately. You do not need to reboot the proxy instance. If a read/write splitting connection fails due to a multi-statement execution, changing the multi-statement processing mode will not restore the connection. You will need to reconnect the connection manually.
- Parse-based mode description:

  If multi-statements contain the operations listed here, all subsequent requests are routed to the primary node. To restore read/write splitting, you need to disconnect the connection and then re-establish it.

  – Creating temporary tables

  – Creating stored procedures

  – Executing uncommitted transactions (for example, **begin** is executed but **commit** or **rollback** is not executed)

 – Executing complex or special syntax. These statements will not be parsed.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click a proxy instance name to go to the **Basic Information** page.

Step 7 In the navigation pane, choose **Parameter Modifications**.

Step 8 Configure the parameter **multiStatementType** as required.

Figure 11-21 Configuring the parameter **multiStatementType**

You can set this parameter to **Strict**, **Loose**, or **Parse**.

Step 9 Click **Save** to save your change. In the displayed dialog box, click **Yes**.

----End

11.3.7 Enabling Automatic Association of New Nodes with a Proxy Instance

After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the current proxy instance.

This section describes how to enable or disable **Associate New Nodes** for an existing proxy instance. To enable this function during the proxy instance creation, see **How to Use a Proxy Instance to Enable Read/Write Splitting**.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy** and click the name of a proxy instance to go to the **Basic Information** page.

Step 6 In the **Proxy Instance Information** area, click next to **Associate New Nodes**.

Figure 11-22 Going to the Basic Information page

Step 7 In the displayed dialog box, enable **Associated New Nodes**.

Figure 11-23 Enabling automatic association of new nodes with a proxy instance
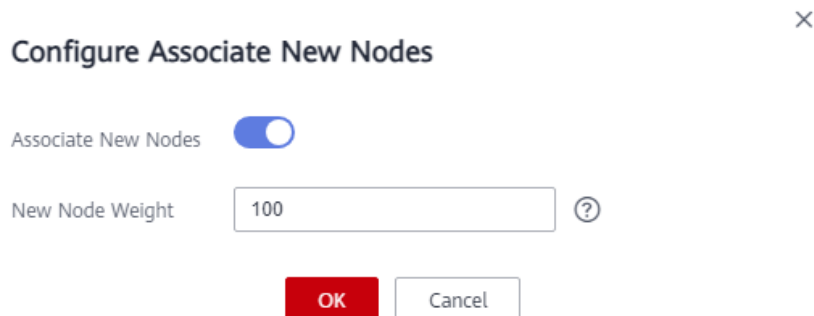
When the routing policy is weighted, you need to configure weights for the new nodes as required. The default read weight of any new node is 100. Nodes with higher weights process more read requests.

Step 8 Click **OK**.

To disable the function, click ⬤━.

----End

11.3.8 Enabling Access Control for a Proxy Instance

If load balancing is enabled for a proxy instance, the security group associated with the proxy instance does not apply. You need to use access control to limit access from specific IP addresses.

Constraints

If access control is not displayed on the management console, the security group associated with the proxy instance is used.

Enabling Access Control

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy** and click the name of a proxy instance to go to the **Basic Information** page.

Step 6 Click ⬤ next to **Access Control**.

Step 7 Click **Configure**. In the displayed dialog box, configure required parameters.

Figure 11-24 Configuring access control
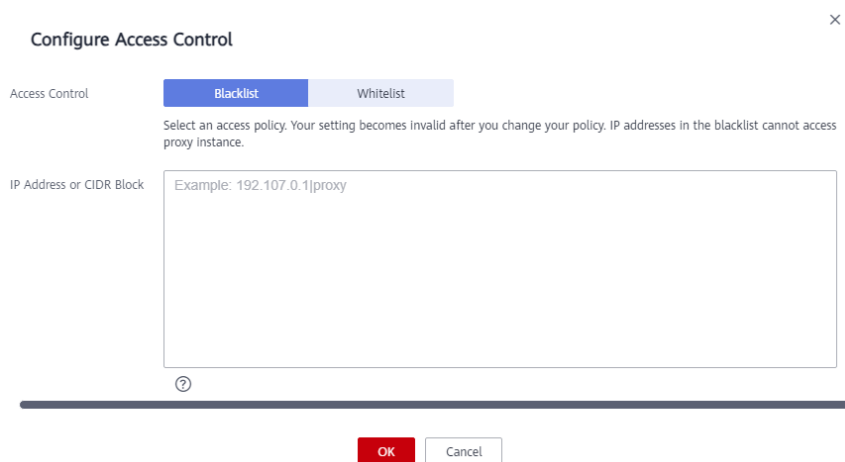
Table 11-5 Parameter description

Parameter	Description
Access Control	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 or CIDR blocks in the blacklist are not allowed to access the proxy instance.
IP Address or CIDR Block	You need to enter IP addresses or CIDR blocks that meet the following requirements: ● Each line contains an IP address or a CIDR block and ends with a line break. ● Each IP address or CIDR block can include a description separated by a vertical bar symbol (|), for example, **192.168.10.10|GaussDBforMySQL01**. The description can include up to 50 characters but cannot contain angle brackets (<>). ● Up to 300 IP addresses or CIDR blocks can be added.

----End

Disabling Access Control

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy** and click the name of a proxy instance to go to the **Basic Information** page.

Step 6 Click [toggle] next to **Access Control**.

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

----End

11.3.9 Changing the Specifications of a Proxy Instance
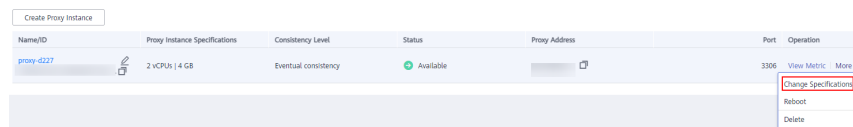
If the proxy instance specifications cannot meet your workload requirements, you can manually upgrade them.

Constraints

- The proxy instance specifications can be changed only when your GaussDB(for MySQL) instance, primary node, and read replicas are normal.
- A proxy instance cannot be deleted when its CPU and memory specifications are being changed.

Procedure

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

Step 2 Click [icon] in the upper left corner and select a region and project.

Step 3 Click [icon] in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 On the **Database Proxy** page, locate the desired proxy instance and choose **More** > **Change Specifications** in the **Operation** column.

Figure 11-25 Changing proxy instance specifications (1)

Alternatively, click the proxy instance name. In the **Proxy Instance Information** area, click **Change** next to the **Specifications** field.

Figure 11-26 Changing proxy instance specifications (2)

Step 7 In the displayed dialog box, select new specifications and click **OK**. You can reduce or expand the specifications as required.

Figure 11-27 Changing proxy instance specifications (3)
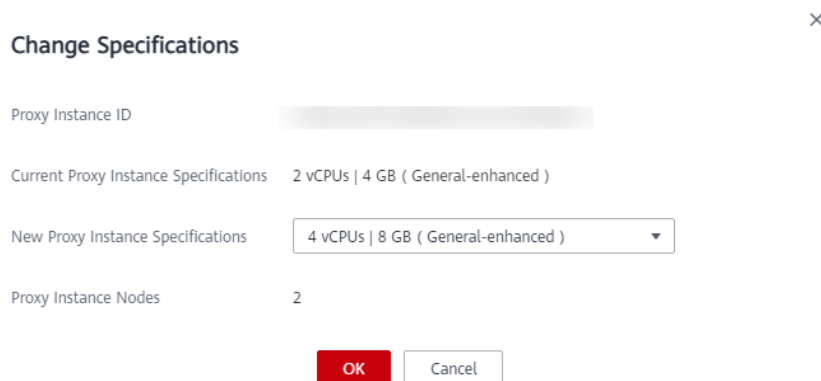
Step 8 View the new specifications on the **Database Proxy** page.

----End

APIs

- **Changing Specifications of a Proxy Instance**
- **Querying Proxy Instances**
- **Querying Proxy Instance Specifications**

11.3.10 Changing the Number of Nodes for a Proxy Instance

Scenarios

You can change the number of proxy instance nodes as required.

Constraints

- Your GaussDB(for MySQL) instance must be available.
- If a proxy instance is abnormal, you can only add nodes to it but cannot reduce nodes.
- The number of proxy nodes ranges from 2 to 32.

● The number of proxy nodes ranges from 2 to 16.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**. Click the name of a proxy instance.

Step 6 In the **Proxy Instance Information** area, click **Change** next to **Proxy Instance Nodes**.

Step 7 In the displayed dialog box, set the number of proxy instance nodes and click **OK**.

Number of recommended proxy instance nodes = (Number of vCPUs of the primary node + Total number of vCPUs of all read replicas)/(4 x Number of vCPUs of the proxy instance), rounded up.

Figure 11-28 Changing the number of proxy nodes
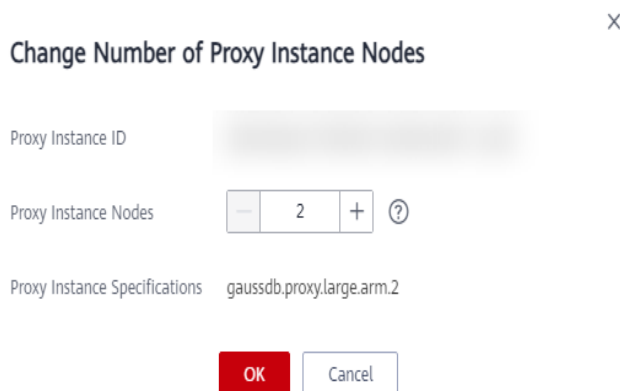
----End

APIs

● **Adding Proxy Nodes**

● **Querying Proxy Instances**

● **Deleting Proxy Nodes**

11.3.11 Applying for a Private Domain Name for a Proxy Instance

You can use a private network domain name to connect to a proxy instance.

Constraints

To use this function, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

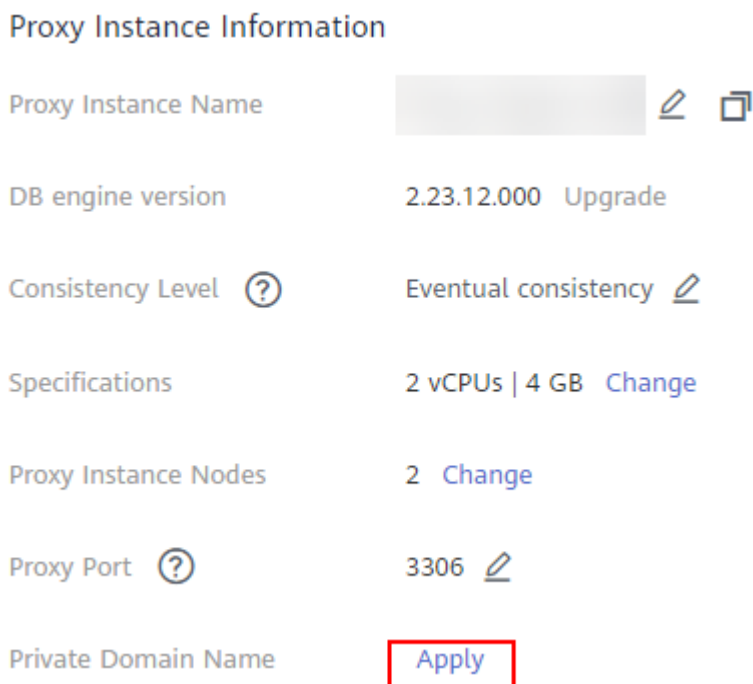## Applying for a Private Domain Name for a Proxy Instance

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance.

Step 7 In the **Proxy Instance Information** area on the **Basic Information** page, click **Apply** in the **Private Domain Name** field.

Figure 11-29 Applying for a private domain name

Proxy Instance Information

Proxy Instance Name		
DB engine version	2.23.12.000 Upgrade	
Consistency Level ⑦	Eventual consistency	
Specifications	2 vCPUs | 4 GB Change	
Proxy Instance Nodes	2 Change	
Proxy Port ⑦	3306	
Private Domain Name	Apply	

Step 8 Click **OK**.

Step 9 In the **Private Domain Name** field, view the generated private domain name.

----End

Changing the Private Domain Name of a Proxy Instance

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 In the **Proxy Instance Information** area on the **Basic Information** page, click **Change** in the **Private Domain Name** field.

Step 7 In the displayed dialog box, enter a new domain name and click **OK**.

> **NOTE**
>
> - Only the prefix of a private domain name can be modified.
> - The prefix of a private domain name contains 8 to 63 characters, and can include only lowercase letters and digits.
> - The new private domain name must be different from existing ones.

----End

Deleting the Private Domain Name of a Proxy Instance

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 In the **Proxy Instance Information** area on the **Basic Information** page, click **Delete** in the **Private Domain Name** field.

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

----End

11.3.12 Changing the Port of a Proxy Instance

Scenarios

You can change the port for a proxy instance.

Constraints

- Changing a proxy port will interrupt the database connection. You are advised to change the port during off-peak hours.
- Only the port of the current proxy instance will be changed.

Procedure

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

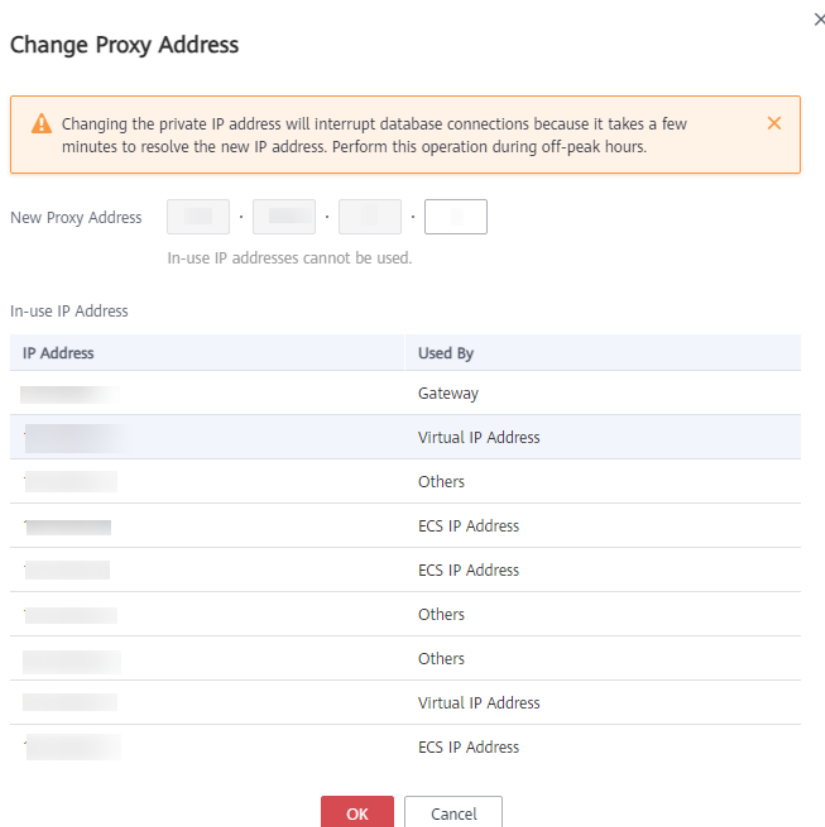**Step 2** Click ⦾ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click ✎ next to **Proxy Port**.

Proxy port range: 1025 to 65534 (except for 1033, 5342, 5343, 5344, 5345, 12017, 20000, 20201, 20202, 33060, 33062, and 33071, which are reserved by the system)

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

----End

APIs

Changing the Port of a Proxy Instance

11.3.13 Changing the Proxy Address of a Proxy Instance

Scenarios

You can change the proxy address of a proxy instance.

Constraints

- Changing a proxy address will interrupt database connections and services. Perform the operation during off-peak hours or when services are stopped.
- The new proxy address is not in use and must be associated with the same subnet as your GaussDB(for MySQL) instance.

Procedure

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

Step 2 Click ⦾ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance. In the **Proxy Instance Information** area, click **Change** next to **Proxy Address**.

Figure 11-30 Changing the address of a proxy instance (1)

Step 7 In the displayed dialog box, enter a new IP address and click **OK**.

In-use IP addresses cannot be used.

Figure 11-31 Changing the address of a proxy instance (2)

----End

11.3.14 Modifying Parameters of a Proxy Instance

Scenarios

You can change parameters for a proxy instance.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Database Proxy**, select a proxy instance and click its name.

Step 6 In the navigation pane on the left, choose **Parameter Modifications**. On the displayed page, change parameters if needed.

You can save, cancel, or preview your changes.

- To save your changes, click **Save**.
- To cancel your changes, click **Cancel**.
- To preview your changes, click **Preview**.

----End

11.3.15 Binding an EIP to a Proxy Instance

After a proxy instance is created, you can bind an EIP to it. Later, you can also unbind the EIP from the proxy instance as required.

Constraints

To use this function, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

**Step 3** Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click a proxy instance name to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click **Bind** next to **Public IP Address (EIP)**.

**Figure 11-32** Binding an EIP to a proxy instance (1)
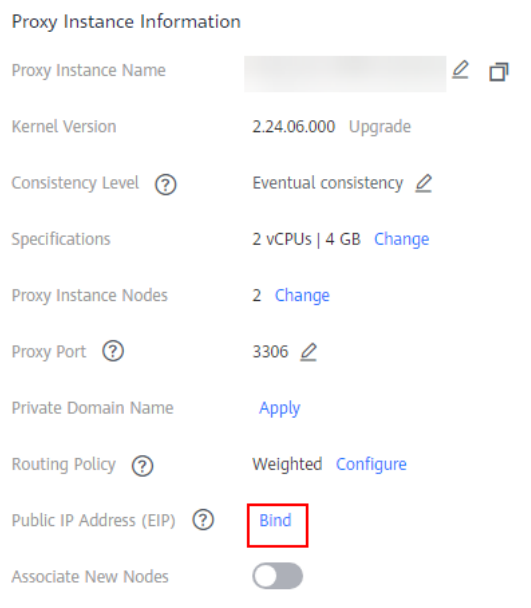
Step 8 In the displayed dialog box, select an EIP and click **OK**.

Figure 11-33 Binding an EIP to a proxy instance (2)
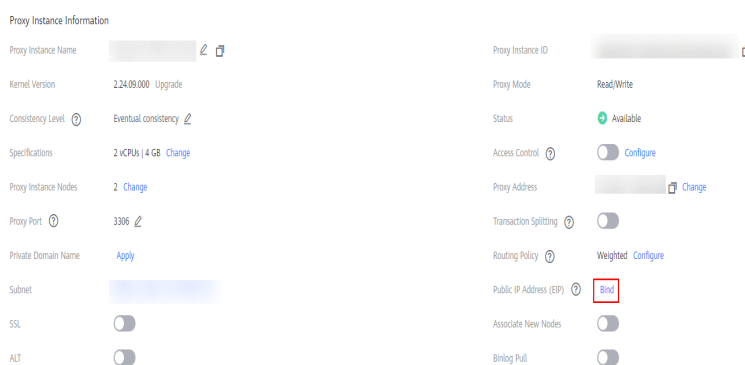
Step 9 On the **Basic Information** page, view that the EIP has been bound to the proxy instance.

To unbind an EIP from the proxy instance, click **Unbind** next to **Public IP Address (EIP)**. In the displayed dialog box, click **Yes** to unbind the EIP.

Figure 11-34 Unbinding an EIP from a proxy instance
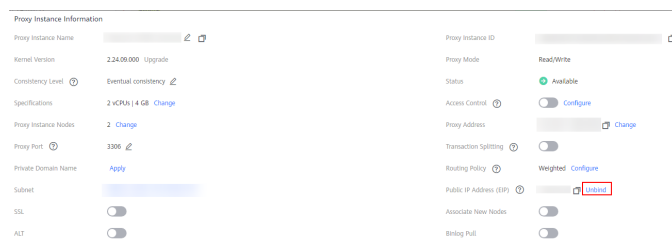
----End

11.4 Proxy Instance Lifecycle

11.4.1 Rebooting a Proxy Instance

Scenarios

You can reboot a proxy instance you have created.

Constraints

- If the proxy instance status is **Abnormal**, the reboot may fail.
- Reboot a proxy instance interrupts the database connection. You are advised to reboot it during off-peak hours. To shorten the time required, reduce database activities during the reboot to reduce rollback of transit transactions.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.
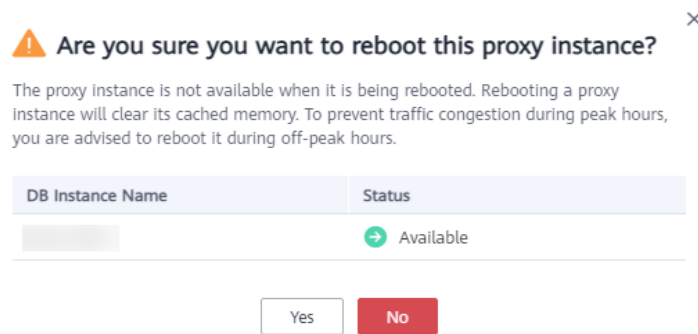
Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Database Proxy**, locate the target proxy instance, and choose **More** > **Reboot** in the **Operation** column.

Figure 11-35 Rebooting a proxy instance
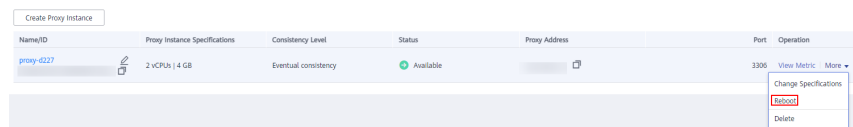
Step 6 In the displayed dialog box, click **Yes**.

Figure 11-36 Confirming information

----End

11.4.2 Deleting a Proxy Instance

You can delete a proxy instance as required.

Constraints

If a proxy instance is deleted, read/write splitting is disabled and workloads using the proxy address are interrupted. You need to connect your applications to the GaussDB(for MySQL) instance address.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Select the target proxy instance and choose **More** > **Delete** in the **Operation** column.

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

Figure 11-37 Deleting a proxy instance

----End
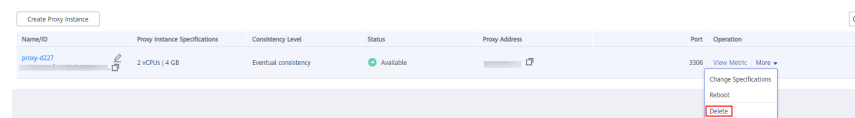
APIs

- **Creating a Proxy Instance**
- **Deleting a Proxy Instance**

11.5 Proxy Instance Kernel Versions

11.5.1 Kernel Version Release History

Released On	Version	Description
2024-05-07	2.24.03.000	Added the feature for assigning requests to row and column store nodes.
2024-01-15	2.23.12.000	- Added the feature for collecting statistics on slow query logs of proxy instances. - Fixed the issue that there is a delay when a proxy instance synchronizes authentication information from the database kernel.
2024-01-04	2.23.09.002	Fixed the logic for proxy instances to retry service SQL statements after the database is faulty.
2023-11-13	2.23.09.001	Fixed the issue that an error is occasionally reported during execution of the prepared **SELECT FOR UPDATE** statement.
2023-10-20	2.23.09.000	New features: - **Change User** protocol - Parsing of multiple hints - **SHOW PROCESSLIST** and **KILL** commands
2023-07-31	2.23.06.001	Resolved the increased backend database connections caused by enabling session connection pool.
2023-07-06	2.23.06.000	- Added binlog pulling through the proxy instance kernel. - Optimized the performance of the **PREPARE STMT** protocol again.

Released On	Version	Description
2023-06-11	2.23.02.007	Fixed issues: ● Optimized the performance of the **PREPARE STMT** protocol. ● Resolved unexpected traffic allocation of the **/* FORCE_SLAVE*/ Hint** statement. ● Fixed the issue that the **set autocommit** setting is synchronized to read replicas after transaction splitting is enabled.
2023-04-27	2.23.02.000	Optimized the proxy instance performance.
2022-12-05	2.22.11.000	Added multi-statement processing modes. Optimized the error messages reported during SQL statement execution in some scenarios.
2022-09-06	2.22.07.000	New features: ● Session-level connection pooling ● Dynamic load balancing Optimized the logic for setting session-level transaction isolation levels of proxy instances. By default, the transaction isolation levels are synchronized with those of the database.
2022-06-15	2.7.5.0	Added Application Lossless and Transparent (ALT).
2022-05-06	2.7.4.0	New features: ● A query for more than 16 MB of data ● Session consistency Optimized the way how metrics of read-only proxy instances are collected by Cloud Eye.
2022-04-01	2.3.9.8	Added batch execution of prepared statements.
2022-02-09	2.3.9.7	New features: ● Transaction splitting ● Read-only mode Optimized the execution logic of prepared statements to improve performance.

Released On	Version	Description
2021-04-23	2.3.9.0	Added proxy instance performance metrics **Front-End Connections Created per Second**, **Transaction Queries per Second**, and **Multi-Statement Queries per Second**. Fixed issues: ● Optimized the database proxy performance. ● Fixed traffic congestion occurring when your applications connect to a proxy instance over short connections.
2021-01-14	2.3.8.0	Added the feature for obtaining client IP addresses through proxy instances. Fixed issues: ● Fixed the issue that monitoring data of database proxy is inaccurate. ● Shortened the downtime of proxy instances during a primary/standby switchover.
2020-10-14	2.3.6.0	Fixed issues: ● Fixed the issue of connection failures caused by database overload. ● Improved proxies' compatibility with MySQL protocols.
2020-08-14	2.3.1.0	New features: ● Maintaining connectivity between clients and database proxies. ● Monitoring performance metrics of proxy instances.

11.5.2 Upgrading the Kernel Version of a Proxy Instance

You can manually upgrade your proxy instance to the latest kernel version to improve performance, add new functions, and fix problems.

Constraints

Intermittent disconnections occur during an upgrade. The time required to complete the upgrade depends on how many proxy instances there are. Perform the upgrade during off-peak hours.

Procedure

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

Step 2 Click [icon] in the upper left corner and select a region and project.

Step 3 Click [icon] in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click the name of a proxy instance to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click **Upgrade** next to **Kernel Version**.

Step 8 In the displayed dialog box, select a scheduled time and click **OK**.

- Upon submission: The system upgrades the proxy instance to the latest version immediately after you submit the request. You can view the task progress in **Task Center** > **Instant Tasks**.

- In maintenance window: The system upgrades the proxy instance to the latest version during a maintenance window. You can view the task progress in **Task Center** > **Scheduled Tasks**.

----End

11.6 Using Hints for Read/Write Splitting

In addition to configuring weights of nodes for read/write splitting, you can use hints in SQL statements to route read and write requests to a primary node or read replica.

Precautions

- Hints are only used as routing suggestions. In non-read-only SQL and non-transaction scenarios, SQL statements cannot be routed to read replicas.

- If you want to connect to a DB instance using the MySQL CLI and hints, add the **-c** option.

Usage

You can add the following hints at the beginning of an SQL statement as needed.

/*FORCE_MASTER*/: The SQL statement is executed on the primary node.

/*FORCE_SLAVE*/: The SQL statement is executed on read replicas.

For example, if you run **select * from table1**, the SQL statement will be executed on a read replica by default. If you change it to **/*FORCE_MASTER*/ select * from table1**, the SQL statement will be executed on the primary node.

> ⚠️ **CAUTION**
>
> **/*FORCE_MASTER*/** only works for read/write addresses. Even if you use it for a read-only address, the SQL statement will not be executed on the primary node.

12 DBA Assistant

12.1 Function Overview

Description

DBA Assistant provides visualized database O&M and intelligent diagnosis for developers and database administrators (DBAs), making database O&M easy and efficient. By analyzing alarms, resource usage, health status, performance metrics, and storage usage, it helps you quickly locate faults and keep track of instance statuses.

> **NOTE**
>
> To use DBA Assistant on the GaussDB(for MySQL) console, you need to ensure that IAM users must have the **GaussDB FullAccess**, **DAS FullAccess**, **DAS Administrator**, and **CES FullAccess** permissions. For details, see **Creating a User and Granting GaussDB(for MySQL) Permissions**.

Functions

Table 12-1 lists the functions supported by DBA Assistant.

Table 12-1 Function description

Function	Description	Reference
Dashboard	Shows the status of your instance, including alarms, resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.	**Viewing the Operating Status of a DB Instance**

Functio n	Description	Reference
Sessions	The **Sessions** page displays slow sessions, active sessions, and total sessions. You can quickly filter slow sessions or active sessions by user, host IP address, or database name. Kill Session and Concurrency Control can be used for urgent instance recovery to ensure database availability.	**Managing Real-Time Sessions**
Perform ance	The **Performance** page displays key metrics of your instance and provides metric comparison between different days. You can keep track of metric changes and detect exceptions in a timely manner. Monitoring by Seconds helps accurately locate faults.	**Performance Monitoring**
Storage Analysis	Storage occupied by data and logs and changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DBA Assistant can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner. **Autoscaling**, **Tablespaces**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.	**Managing Storage**
Slow Query Log	Displays slow queries within a specified time period. You can view top 5 slow query logs by user or IP address, sort statistics, and identify sources of slow SQL statements.	**Viewing Slow Query Logs**
SQL Explorer	After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.	• **Viewing Top SQL Statements** • **Creating an SQL Insights Task**
Concurr ency Control	Concurrency Control restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.	**Configuring SQL Statement Concurrency Control**

12.2 Performance Monitoring

12.2.1 Viewing the Operating Status of a DB Instance

The **Dashboard** page allows you to view the operating status of the current DB instance, including alarms, health check results, compute resource usage, storage resource usage, and key performance metrics.

Alarms

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

Step 2 Click ⦾ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 On the **Dashboard** page, view instance alarms provided by Cloud Eye.

You can customize alarm rules by adjusting alarm policies and severities for key metrics, such as CPU usage and disk usage. To view alarm details, click the number next to an alarm severity.

Figure 12-1 Alarms

Figure 12-2 Alarm list
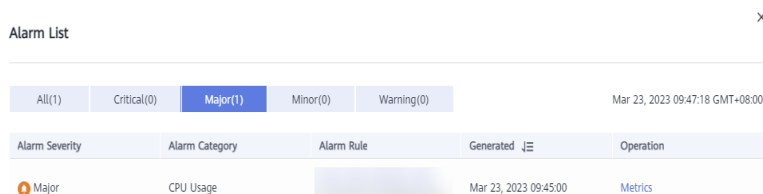
----End

Health

In the **Health** area, you can view real-time health check results. By default, the data for high vCPU utilization, memory bottlenecks, high-frequency slow SQL statements, and lock waits are displayed.

For abnormal metrics, click **Diagnose** to view diagnosis details and suggestions. For details, see **Table 12-2**.

Figure 12-3 Health

Table 12-2 Health diagnosis and suggestions

Item	Exception Trigger Condition
High vCPU utilization	Either of the following conditions is met: • After you configure alarm rules on Cloud Eye, an alarm is reported, indicating the CPU usage is high. • The CPU usage exceeds 95% for more than 2.5 minutes of a 5-minute measurement period.
Memory bottleneck	Either of the following conditions is met: • After you configure alarm rules on Cloud Eye, an alarm is reported, indicating the memory usage is high. • The memory usage exceeds 95% within a 5-minute measurement period.
High-frequency slow SQL	Either of the following conditions is met: • After you configure alarm rules on Cloud Eye, an alarm is reported, indicating there are too many slow logs. • There are more than 100 slow logs within five minutes.
Lock wait	After you configure alarm rules on Cloud Eye, any of the following alarms is reported: • Row Lock Time • InnoDB Row Locks • Row Lock Waits

☐ **NOTE**

- For details about how to configure alarm rules on Cloud Eye, see **Configuring Alarm Rules**.
- For details about metrics, see **Viewing Monitoring Metrics**.

Compute Resource Usage

In the **Compute Resource Usage** area, the vCPU usage and memory usage are displayed by default. The displayed values are the average values for 5-minute measurement periods.

Figure 12-4 Compute Resource Usage
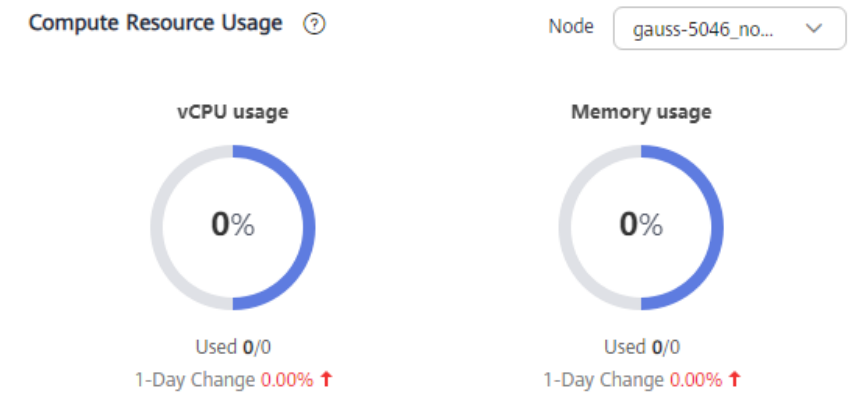
Storage Resource Usage

In the **Storage Resource Usage** area, the storage usage, disk read IOPS, and disk write IOPS are displayed by default. The displayed values are the average values for 5-minute measurement periods.
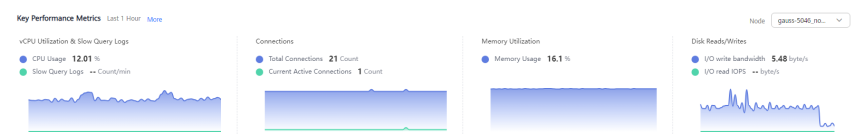
Figure 12-5 Storage Resource Usage
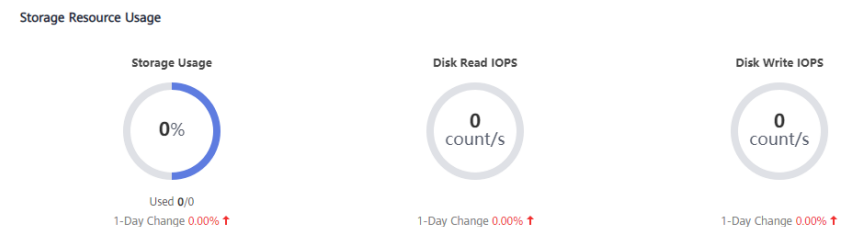
Key Performance Metrics

In the **Key Performance Metrics** area, the CPU usage & slow query logs, connections, memory utilization, and disk reads/writes from the last hour are displayed by default. The displayed values are real-time values.

Figure 12-6 Key Performance Metrics

12.2.2 Viewing Real-Time Performance Metrics

GaussDB(for MySQL) allows you to view performance metrics and trends of DB instances in real time, helping you detect and handle potential performance problems in a timely manner.

Procedure

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

Step 2 Click [icon] in the upper left corner and select a region and project.

Step 3 Click [icon] in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Performance** tab to view the performance metrics of your DB instance.

- If you select **Comparison by Date**, you can view metric trends of the DB instance in a time range on different dates. You can move the cursor to a point in time of a chart to view metric values at the point in time on different dates.

  Figure 12-7 Viewing a performance metric at a point in time on different dates
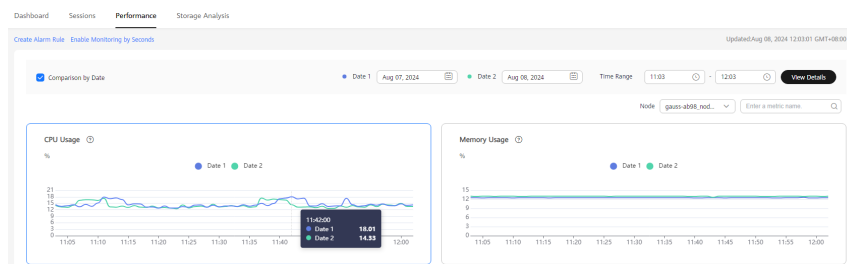
- If you deselect **Comparison by Date**, you can view performance metric trends in the last 30 minutes, last hour, last 6 hours, or a custom time range. You can move the cursor to a point in time of a chart to view the metric value at the point in time.

  Figure 12-8 Viewing a performance metric trend in the last 30 minutes
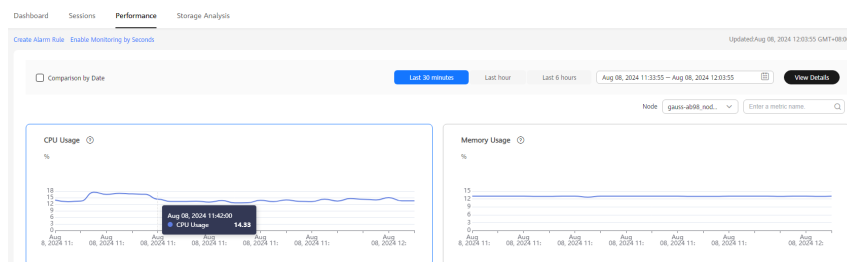
- You can also click **Create Alarm Rule** to set alarm rules for your DB instance. This will allow you to stay informed about the status of your DB instance and receive timely warnings.

- The system monitors performance data every minute by default. You can click **Enable Monitoring by Seconds** on the **Performance** tab to configure linked charts and enable monitoring by seconds.

  Linked Charts: Enabling it means that you can view all metrics at the same time.

  Monitoring by Seconds: Enabling it means that the system reports monitoring data precise to the second or to 5 second intervals. This function is billed and the pricing is listed on a per-hour basis.

Figure 12-9 Enabling monitoring by seconds
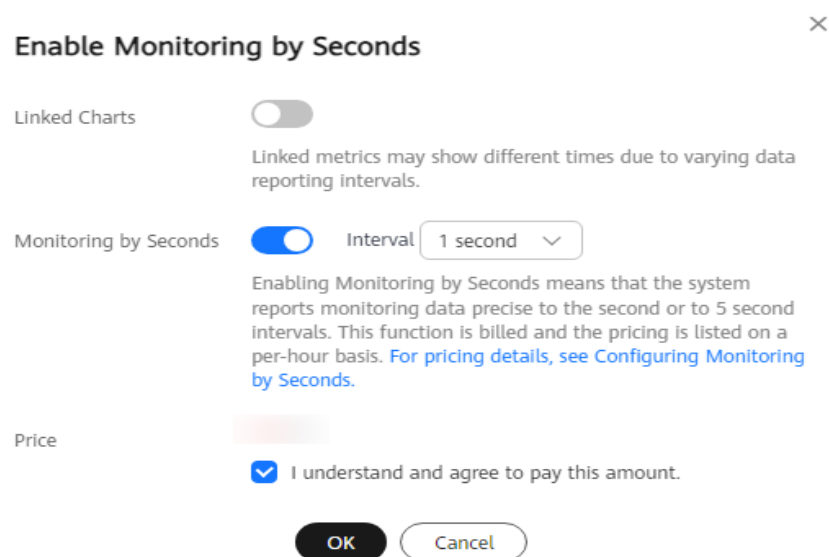
----**End**

12.3 Problem Diagnosis

12.3.1 Managing Real-Time Sessions

Scenarios

You can view current session statistics of your instance and kill abnormal sessions.

Setting a Slow Session Threshold

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Sessions** tab to view current session statistics by user, access host, and database.

Figure 12-10 Sessions
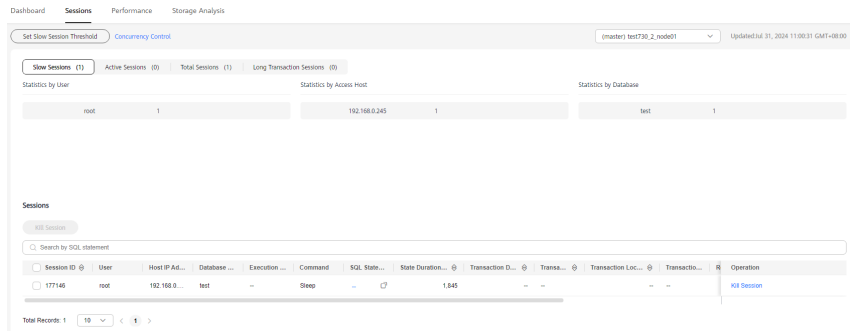
Step 7 Click **Set Slow Session Threshold**. In the displayed dialog box, configure **Max. Execution Time for a Query (s)** and click **OK**. Sessions whose execution time exceeds the threshold are automatically displayed.

Figure 12-11 Setting a slow session threshold
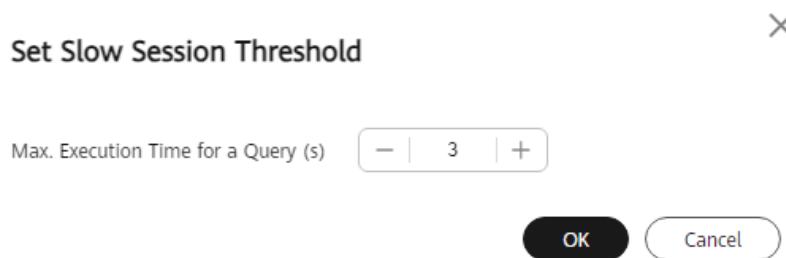
$\Box$ **NOTE**

Too long SQL statements will be truncated and displayed in the session list.

Step 8 In the session list, select the abnormal session you want to kill and click **Kill Session** to recover the database.

A maximum of 20 sessions can be killed at a time.

$\Box$ **NOTE**

To kill sessions automatically, see **Configuring Auto Flow Control**.

----End

12.3.2 Managing Storage

Storage occupied by data and logs and changes of storage usage are important for database performance. On the **Storage Analysis** page, you view the distribution and change trend of the disk space. **Autoscaling**, **Tablespaces**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.

Functions

Table 12-3 Functions

Function	Description	Related Operation
Overview	You can view storage usage, available storage, total storage, daily increase in the last week, and estimated available days of storage.	**Viewing Storage Usage**
Tablespaces	You can view tables with abnormal tablespace growth, tables without primary keys, and tables without indexes.	**Tablespaces**
Disk Space Distribution and Used Disk Space	You can view the distribution and change trend of the disk space.	**Viewing Disk Space Distribution**
Top Databases and Tables	You can view the top 50 databases and tables by physical file size and identify the high-usage databases and tables based on disk space distribution.	**Top Databases and Tables**

Viewing Storage Usage

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab. In the **Overview** area, view the storage usage.

Figure 12-12 Viewing the storage overview
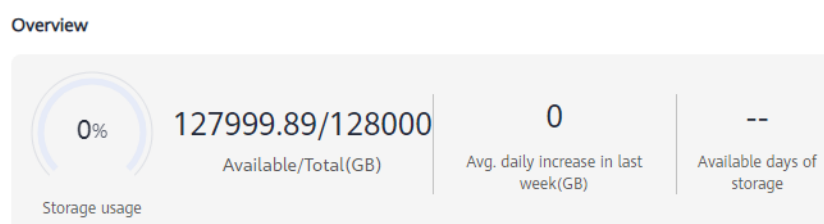
The following information is displayed:

- Storage usage
- Available and total storage
- Average daily increase in the last week
- Available days of storage

  📖 **NOTE**

  If the average daily increase in last week is 0 GB, the estimated available days of storage are unlimited and are not displayed.
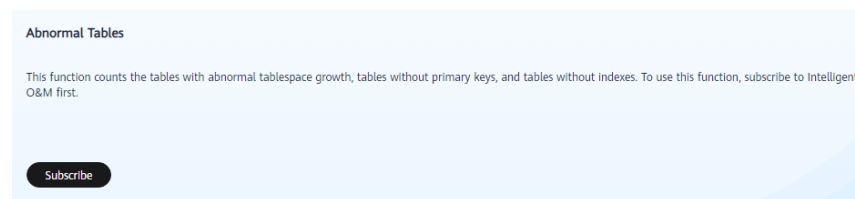
----End

Tablespaces

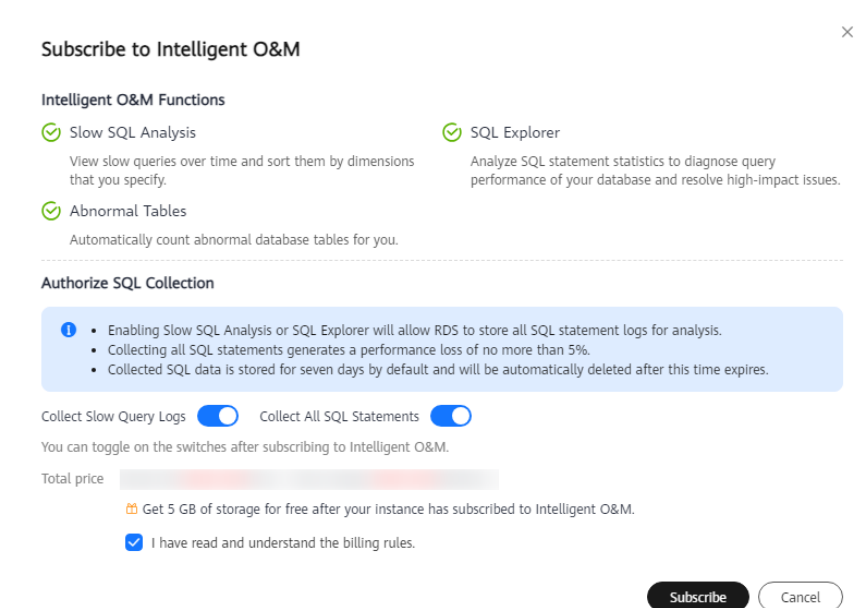You can view tables with abnormal tablespace growth, tables without primary keys, and tables without indexes.

Step 1 In the **Abnormal Tables** area, click **Subscribe**.

Figure 12-13 Abnormal Tables

Step 2 In the **Subscribe to Intelligent O&M** dialog box, confirm the information, select the agreement, and click **Subscribe**.

Figure 12-14 Subscribing to Intelligent O&M

Step 3 In the **Tablespaces** area, view table diagnosis results.

Figure 12-15 Viewing table diagnosis results
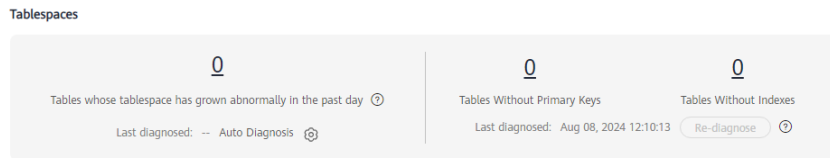
Both automated diagnosis and manual diagnosis are supported.

- Automated diagnosis

  Tables in the **Top 50 Tables** area are automatically diagnosed at about 04:00 every day.

  In the left part of the **Tablespaces** area, you can view tables whose tablespace has grown abnormally in the past day. You can click the number to view the diagnosis details and handle the abnormal tables based on the suggestions provided.

  Figure 12-16 Viewing diagnosis details
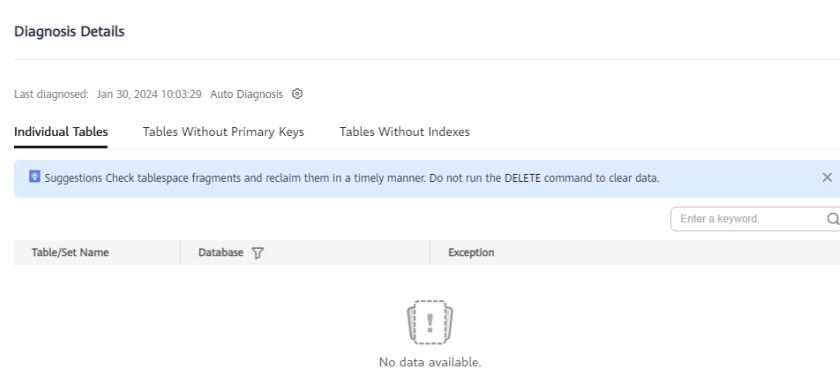
  Any table whose tablespace has grown by more than 10,240 MB in the past day is counted. You can also click ⚙ on the right of **Auto Diagnosis** to set the upper limit for daily tablespace increase.

 **Figure 12-17** Setting the upper limit

- **Manual diagnosis**

  Click **Re-diagnose** to manually trigger a diagnosis task. This operation can be performed every 10 minutes. The diagnosis scope is not limited.

  Once the diagnosis is complete, you can view the numbers of tables without primary keys and tables without indexes. You can click a number to view the diagnosis details and handle the abnormal tables based on the suggestions provided.

  Figure 12-18 Viewing diagnosis details
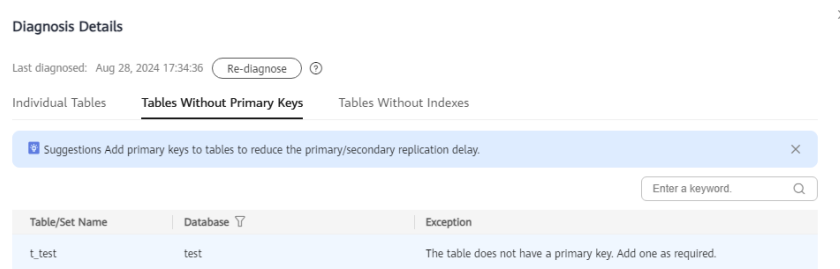
  ☐ **NOTE**

  – If there are more than 5,000 tables, manual diagnosis cannot be used.
  – If the CPU usage exceeds 90%, manual diagnosis cannot be used.

  ----End

Viewing Disk Space Distribution

You can view the distribution and change trend of the disk space.

**Figure 12-19** Viewing disk space distribution
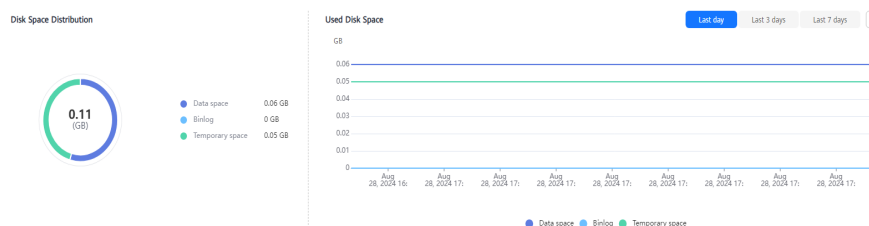
- **Data space**: Disk space occupied by user data
- **Binlog**: Disk space occupied by binlogs
- **Temporary space**: Disk space occupied by temporary files

Top Databases and Tables

Step 1 Click [toggle] on the right of **Collect Top Databases and Tables** to enable the function.

The system automatically collects data of top 50 databases and tables at about 04:00 every day.

Figure 12-20 Enabling Collect Top Databases and Tables
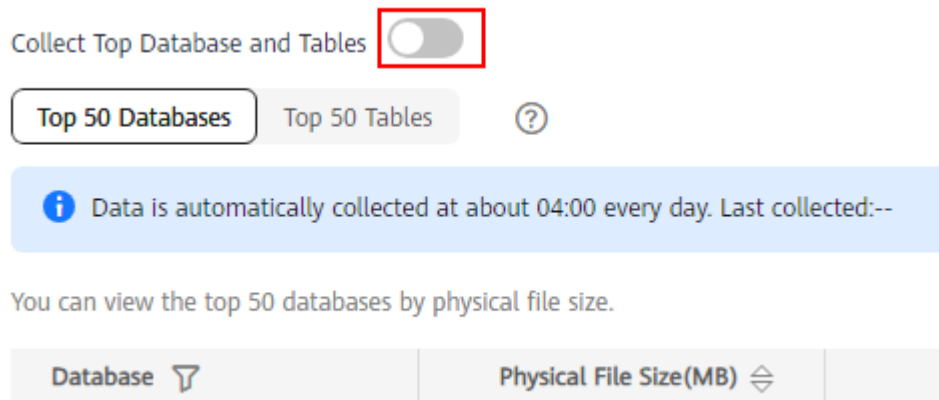
Step 2 View the top 50 databases and tables by physical file size and identify the high-usage databases and tables based on disk space distribution.

📖 NOTE

- Physical file sizes are precisely recorded, but other fields' values are estimated. If there is a large gap between a file size and another field, run ANALYZE TABLE on the table.
- A database or table whose name contains special characters, including slashes (/) and #p#p, is not counted.
- If there are more than 50,000 tables in your instance, to prevent data collection from affecting the instance performance, top databases and tables will not be counted.
- Some statistics may be missing because data of databases or tables is fluctuating.

Figure 12-21 Viewing top 50 databases and tables

Click **View Chart** in the **Operation** column to view data volume changes in the last 7 days, last 30 days, or a custom time period (no longer than 30 days).

----End

12.3.3 Viewing Anomaly Snapshots

After anomaly diagnosis is enabled, the system checks your instance health status and diagnoses faults. If there is an anomaly, its snapshots will be collected, helping you monitor instance performance in real time.

Diagnosis Item

Table 12-4 Diagnosis item

| Item	Description
Transaction uncommitted	There are uncommitted transactions.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click **Anomaly Snapshots**.

Figure 12-22 Anomaly Snapshots page
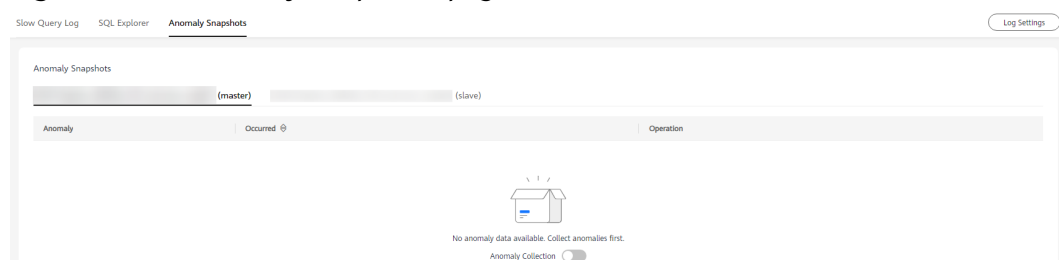
Step 7 Click ⬤ on the right of **Anomaly Collection** to enable anomaly diagnosis.

Figure 12-23 Enabling anomaly diagnosis
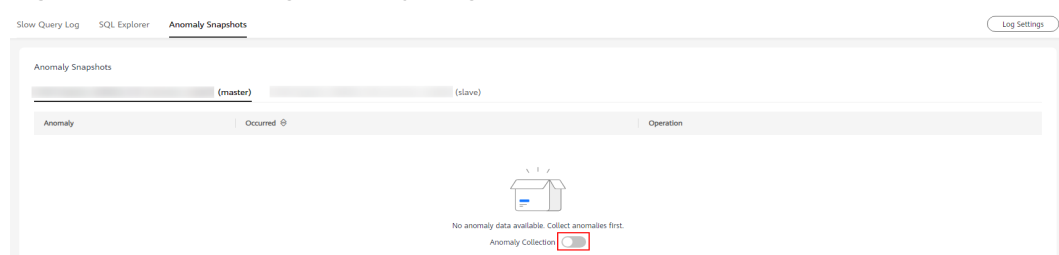
After anomaly diagnosis is enabled, if any anomaly listed in **Table 12-4** occurs, you can view its snapshots. Anomaly snapshot records are retained for seven days and will be deleted after this time expires. A maximum of 100 records can be retained for a single node.

Click **Diagnosis Details** in the **Operation** column to view diagnosis result details and optimization suggestions.

Click the **Anomaly Snapshots** tab to view session snapshots, metadata lock snapshots, InnoDB lock snapshots, and transaction snapshots.

----End

12.4 SQL Analysis and Tunning

12.4.1 Viewing Slow Query Logs

Scenarios

Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Viewing Slow Query Logs

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

Step 2   Click   in the upper left corner and select a region and project.

Step 3   Click   in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4   On the **Instances** page, click the instance name.

Step 5   In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6   Click the **Slow Query Log** tab.

> 📖 NOTE
>
> ● **Slow Query Log** needs to be purchased separately. To use this function, subscribe to Intelligent O&M first.
>
> ● Only the data of the last hour is displayed if Intelligent O&M is not subscribed. The data will be automatically deleted after one hour. After Intelligent O&M is subscribed, data can be stored for up to 30 days. For details, see **Slow Query Log Storage**.
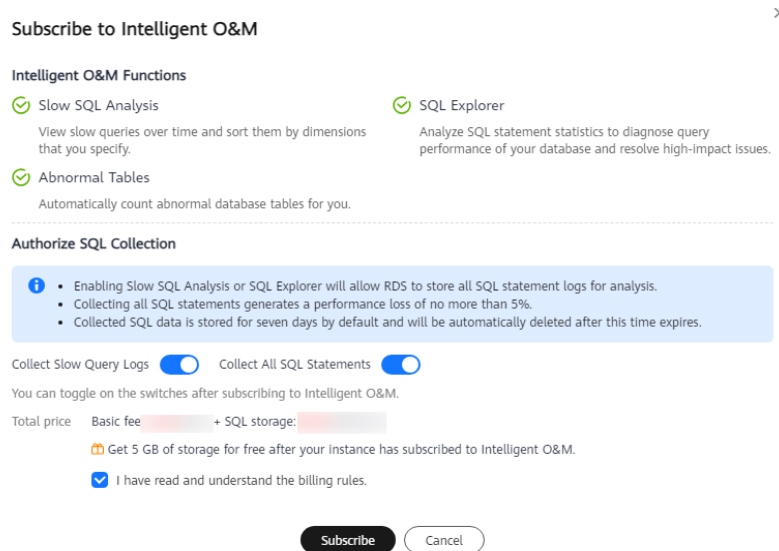
Step 7   Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.

The Intelligent O&M fee for each instance consists of:

● Basic price: $0.075 USD/hour

● SQL storage: $0.0072 USD/GB/hour

Get 5 GB of storage for free after your instance has subscribed to Intelligent O&M.

Figure 12-24 Subscribing to Intelligent O&M

Step 8 Select "I have read and understand the billing rules." and click **Subscribe**.

Step 9 Select a time range and view trends, details, and statistics of the slow query logs generated within the time range.

You can view slow query logs in the last 1 hour, 3 hours, 12 hours, or a custom time period (no longer than one day).

- **Slow Queries over Time**

  Above the chart, you can switch to another instance or node to view its slow queries.

  You can move the cursor to a point in time of the chart to view the number of slow query logs and CPU usage at the point in time.

  Figure 12-25 Slow Queries over Time
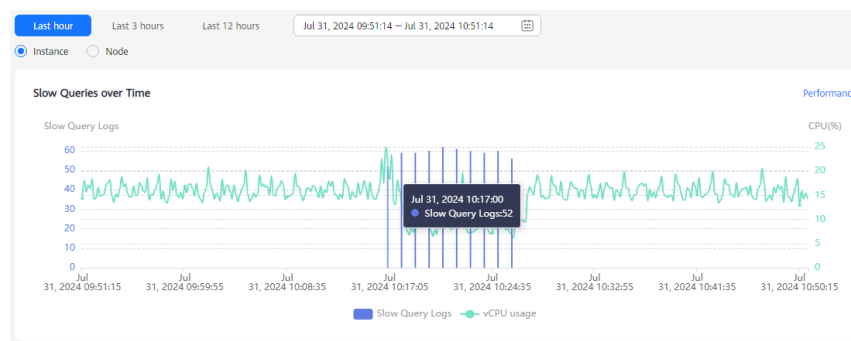
- **Details**

  View slow query log details in the lower part of the **Slow Query Log** page. The details include the SQL statement, execution start time, database, client, user, execution duration, lock wait duration, and scanned and returned rows.

Figure 12-26 Details

Click **Export** to export slow query log details to a specific OBS bucket. After the log details are exported, you can click **View Export List** to view export records.

Filter slow query log details by database, client IP address, or user.

Locate an SQL statement and click **Concurrency Control** in the **Operation** column to create an SQL concurrency control rule. For details, see **Configuring SQL Statement Concurrency Control**.

Click **Export**. In the displayed dialog box, select an OBS bucket and click **OK** to export slow query logs to the OBS bucket. Up to 100,000 logs can be exported.

If no OBS bucket is available, click **Create**. In the displayed dialog box, enter an OBS bucket name, and click **OK**.

Creating an OBS bucket is free, but you will be billed for storing data in the bucket. For pricing details, see **OBS Pricing Details**.

A bucket name:

– Cannot be the same as that of any existing bucket.

– Can contain 3 to 63 characters. Only lowercase letters, numbers, hyphens (-), and periods (.) are allowed.

– Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (.) or contain a period (.) and a hyphen (-) adjacent to each other.

– Cannot be an IP address.

– If the bucket name contains a period (.), certificate-based verification is required when you use the name to access an OBS bucket or object.

After the log details are exported, you can click **View Export List** to view export records. You can also download the details to your local PC for analysis.

● **Statistics**

Figure 12-27 Statistics

Click **View Sample** in the **Operation** to view the sample of the SQL template.

Click **Export**. In the displayed dialog box, select an OBS bucket and click **OK** to export slow query logs to the OBS bucket. Up to 100,000 logs can be exported.

If no OBS bucket is available, click **Create**. In the displayed dialog box, enter an OBS bucket name, and click **OK**.

Creating an OBS bucket is free, but you will be billed for storing data in the bucket. For pricing details, see **OBS Pricing Details**.

A bucket name:

– Cannot be the same as that of any existing bucket.

– Can contain 3 to 63 characters. Only lowercase letters, numbers, hyphens (-), and periods (.) are allowed.

– Cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (.) or contain a period (.) and a hyphen (-) adjacent to each other.

– Cannot be an IP address.

– If the bucket name contains a period (.), certificate-based verification is required when you use the name to access an OBS bucket or object.

– After the templates are exported, you can click **Export Slow Query Logs** to view export records. You can also download the details to your local PC for analysis.

● **Top 5 Slow Query Logs**

View the top 5 slow query logs by user or client IP address.

Figure 12-28 Top 5 Slow Query Logs
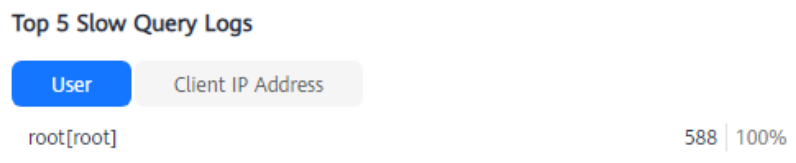
----End

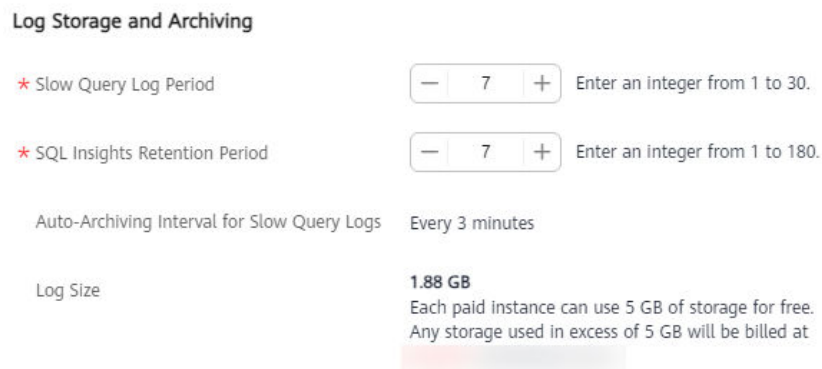## Slow Query Log Storage

After **Collect Slow Query Logs** is enabled, SQL text content will be stored in OBS for analysis.

● If you have subscribe intelligent O&M, click **Log Settings** in the upper right corner.

– **Slow Query Log Period**: The default value is **7**. The value ranges from **1** to **30**. After the period expires, the logs are automatically deleted.

- **SQL Insights Retention Period**: The default value is **7**. The value ranges from **1** to **180**.
- **Log Size**: Each paid instance can use 5 GB of storage for slow query logs for free. Any storage used in excess of 5 GB will be billed on a pay-per-use basis.

Figure 12-29 Log storage and archiving (Intelligent O&M subscribed)

Log Storage and Archiving

★ Slow Query Log Period | − 7 + | Enter an integer from 1 to 30.

★ SQL Insights Retention Period | − 7 + | Enter an integer from 1 to 180.

Auto-Archiving Interval for Slow Query Logs | Every 3 minutes

Log Size | 1.88 GB
Each paid instance can use 5 GB of storage for free.
Any storage used in excess of 5 GB will be billed at

- If you do not subscribe intelligent O&M, click **Log Settings** in the upper right corner.
  - **Slow Query Log Period**: The default value is 1 hour and cannot be changed. After the period expires, the logs are automatically deleted.
  - **SQL Insights Retention Period**: 1 hour

Figure 12-30 Log storage and archiving (Intelligent O&M not subscribed)

Log Storage and Archiving

Slow Query Log Period | 1 hour

SQL Insights Retention Period | 1 hour

Auto-Archiving Interval for Slow Query Logs | Every 3 minutes
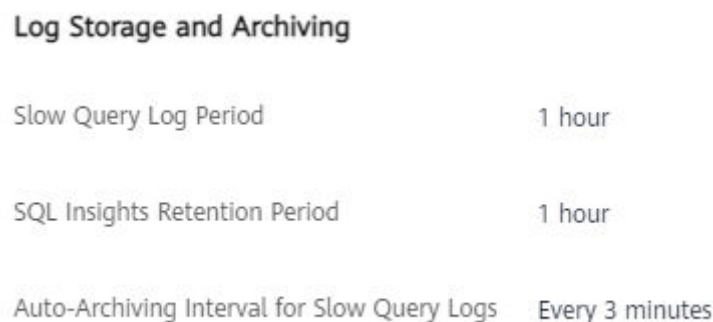
12.4.2 Viewing Top SQL Statements

Scenarios

After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.

Procedure

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

Step 2 Click ⓥ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

**Step 5** In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Top SQL**.

> **NOTE**
>
> - Top SQL needs to be purchased separately. To use this function, subscribe to Intelligent O&M first.
> - Only the data of the last hour is displayed if Intelligent O&M is not subscribed. The data will be automatically deleted after one hour. If Intelligent O&M is subscribed, you can configure how long that top SQL statements are stored for (at most one day).
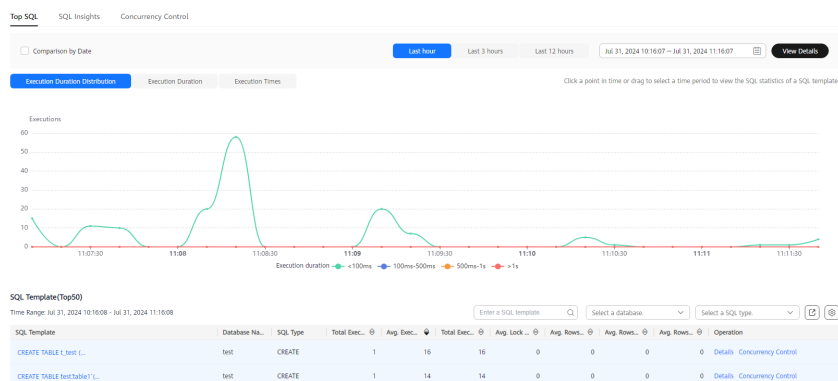
Step 7 Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.

Figure 12-31 Subscribing to Intelligent O&M
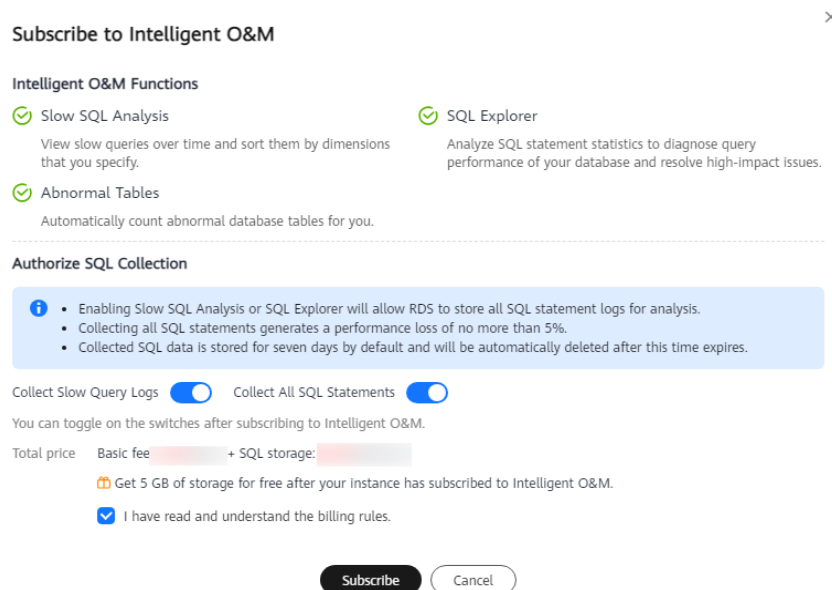
Step 8 View the top SQL statements of the DB instance.

Figure 12-32 Viewing top SQL statements

- View execution durations of the top SQL statements in the last 1 hour, last 3 hours, last 12 hours, or a custom time period (no longer than one day).

- Click a point in time or drag to select a time period to view the SQL statistics of an SQL template.

- Click ⎘ to export information about all top SQL templates in the list. To use this export function, subscribe to Intelligent O&M.

- Locate an SQL template and click **Details** to view the total execution times, average rows scanned, average execution duration, and the like.

- Locate an SQL template and click **Concurrency Control** in the **Operation** column. For details, see **Configuring SQL Statement Concurrency Control**.

- Select **Comparison by Date** and select dates and a time range to view the top SQL statements in the time range on different days.

Figure 12-33 Comparing top SQL statements
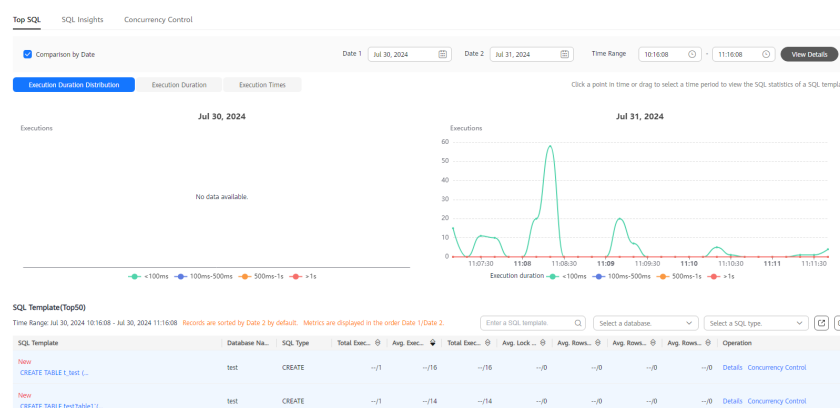
----End

12.4.3 Creating an SQL Insights Task

Scenarios

SQL Insights allows you to not only query all executed SQL statements, but also analyze and search for the tables that are accessed and updated most frequently,

and the SQL statements that have the longest lock wait, helping you quickly identify exceptions.

Constraints

- You need to enable **Collect All SQL Statements** before using SQL Insights.
- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected anymore and the collected SQL data will be deleted.
- Some data cannot be recorded if a buffer overrun occurs.
- If the length of an SQL statement exceeds the value of **rds_sql_tracer_max_record_size**, the statement is not recorded by default.

  To configure the parameter value, see **Modifying Parameters of a DB Instance**.

Procedure

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

Step 2 Click ⚲ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Under the **SQL Explorer** tab, click **SQL Insights**.

Click ⬭ next to **Collect All SQL Statements**.

☐ **NOTE**

- Collecting all SQL statements generates a performance loss of no more than 5%.
- To disable this function, click **Log Settings** in the upper right corner, toggle off the **Collect All SQL Statements** switch, and click **OK**.

Step 7 Click **Create Task**.

Figure 12-34 Creating an SQL insights task
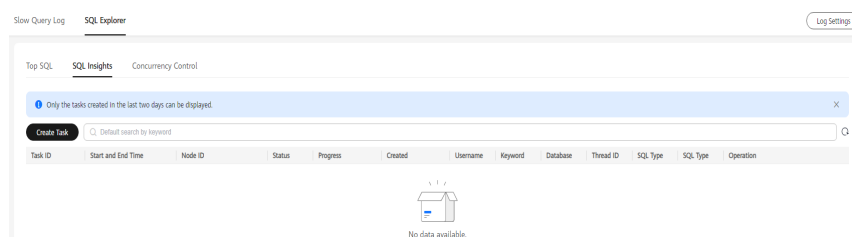
Step 8 On the displayed page, set **Time Range**, **Synchronization to Other Instances**, **Dimension**, **Username**, **Keyword**, **Database**, **Thread ID**, **SQL Type**, and **Status**.

You can set **Dimension** to **Instance** or **Node**. When **Node** is selected, you can view the SQL logs of deleted nodes.

Figure 12-35 Creating an SQL insights task

Step 9 Click **OK**.

Step 10 In the task list, click **Details** in the **Operation** column to view task details.

Step 11 Select a keyword such as **Time Range**, **Username**, **Keyword**, or **Database** to search for the SQL statements executed on the current instance or node.

The selected time range must be after the time when the new task is added.

----End

12.4.4 Configuring SQL Statement Concurrency Control

Scenarios

SQL statement concurrency control aims to keep GaussDB(for MySQL) instances running stably even if there is a sudden increase in concurrent SQL statements.

Constraints

- This function is available only to the GaussDB(for MySQL) instances that meet the following requirements:
  - 2.0.28.40 > kernel version ≥ 2.0.28.15
  - Kernel version ≥ 2.0.29.1
- Each SQL concurrency control rule can contain up to 128 keywords.
- The keywords in a rule cannot contain \t, \r, and \n, and cannot be a backslash (\) or a single null character ('').
- Spaces at the start, end of or in the middle of a keyword are ignored.
- The SQL concurrency control rule cannot end with a tilde (~).
- Keywords in a concurrency control rule are sorted in a specific order, and the system will match them from first to last. For example, if one rule contains the keyword **a~and~b**, the system only matches **xxx a>1 and b>2**.

- Each SQL concurrency control rule applies to only the SQL statements that your database received after the rule is created.

- If different rules are created for the primary node and read replicas of a DB instance, the rules still apply to the primary node and read replicas after their roles are switched over.

- If an SQL statement matches multiple concurrency control rules, only the most recently created rule is applied.

- SQL statements that have been executed before a concurrency control rule is added are not counted.

- The total length of all rules for SELECT, UPDATE, or DELETE statements and the **Concurrency** value in each rule cannot exceed 1024 bytes.

- If you add too many SQL concurrency control rules for your instance, the execution of SELECT, UPDATE, or DELETE statements will slow down.

- SQL concurrency control rules are applied based on prefix match. For example, if the concurrency control rule is **SELECT~COUNT~t1**, SQL statements **SELECT COUNT(*) FROM t1** and **SELECT COUNT(*) FROM t1 LIMIT 1** will both be intercepted.

- After concurrency control is triggered, an execution error is reported on the service side, indicating that query execution was interrupted. The error code is ERROR 1317 (70100).

- This function controls how many statements can run at the same time. However, it does not limit concurrency for:
  - system catalog
  - Queries where no database data is involved, such as **select sleep(xxx)**
  - Account **root**
  - SQL statements in stored procedures, triggers, and functions

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Concurrency Control**.

Step 7 On the displayed page, enable **Concurrency Control**.

Figure 12-36 Enabling SQL statement concurrency control
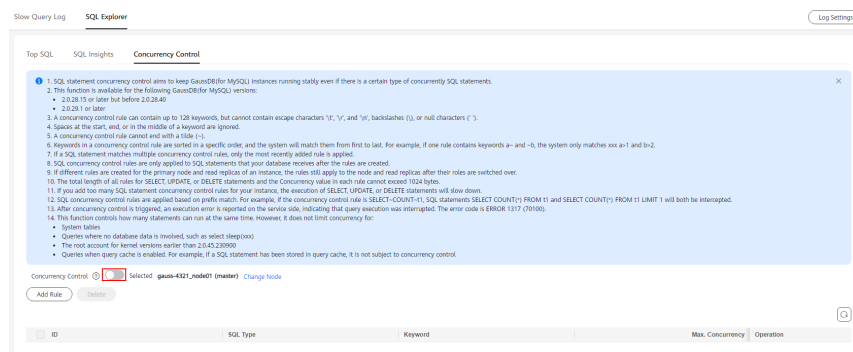
Step 8 Click **Add Rule**. In the displayed dialog box, specify **SQL Type**, **Keyword**, and **Max. Concurrency**.

- **Keyword**: You can enter keywords or copy an existing SQL statement to the text box and click **Generate Keyword**.

  Keyword: Take **select~a** as an example. **select** and **a** are two keywords contained in a concurrency control rule. The keywords are separated by a tilde (~). In this example, the rule restricts the execution of only the SQL statements containing keywords **select** and **a**.

- **Max. Concurrency**: SQL statements that meet the specified SQL type and keyword and exceed the value of **Max. Concurrency** will not be executed.

- If you select **Kill existing sessions that meet this rule**, the sessions that meet the rule will be killed.

- If you select **Synchronize rules to other nodes**, the new rules can be synchronized to other nodes in the same instance.

Step 9 Confirm the settings and click **OK**.

Step 10 If a concurrency control rule is not required, select the rule and click **Delete** above the rule list. In the displayed dialog box, click **OK**.

----End

12.4.5 Configuring Auto Flow Control

Auto flow control allows you to kill all sessions, kill specific sessions by criteria, and view history.

To kill the current session or manually kill a session, see **Managing Real-Time Sessions**.

Functions

- Killing all sessions: After you enable **Auto Kill Sessions** and click **Kill All Sessions**, all sessions are automatically deleted.

- Killing specific sessions by criteria: You can add a task for killing sessions. Sessions that meet the criteria will be killed.

- Viewing history: You can view killed sessions.

Killing Specific Sessions by Criteria

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

**Step 3** Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click **SQL Explorer** and then **Auto Flow Control**.

Step 7 Click ⬤ on the right of **Auto Kill Sessions**. In the displayed dialog box, click **OK**.

Figure 12-37 Enabling Auto Kill Sessions
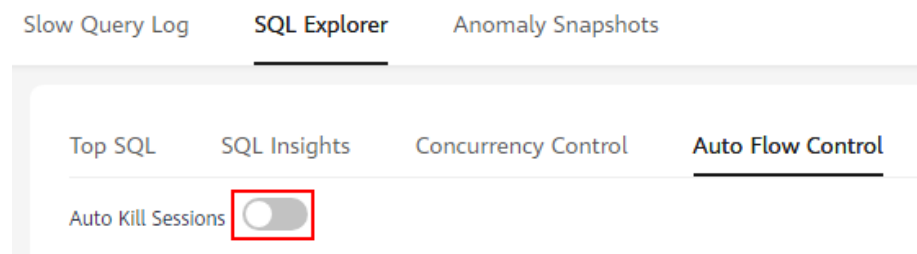
Step 8 Click **Add Kill Task**.

Figure 12-38 Adding a task for killing sessions
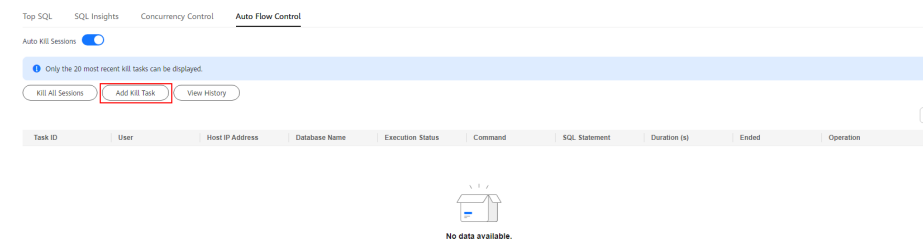
Step 9 In the displayed dialog box, set the criteria for killing sessions.

Figure 12-39 Setting the criteria for killing sessions

NOTICE

- The parameters listed in **Table 12-5** are in a logical AND relationship.
- If you only specify **Session Duration (s)** and **Task Duration (s)**, all sessions that meet the criteria will be killed.
- A maximum of five conditional kill tasks can be executed at the same time.

Table 12-5 Parameter description

Parameter	Description
User	Enter a single value, for example, **root**.
Host IP Address	Enter a single value, for example, **168.192.0.0**.
Database Name	Enter a database name.
Command	Enter a command.
SQL Statement	Enter an SQL statement.
Session Duration (s)	The value ranges from 1 to 2147483647.
Task Closure Method	If you select **Scheduled**, you need to set **Task Duration**. After the duration ends, the task is automatically closed. If you select **Manual**, you can click **Stop** in the **Operation** column of the task list to manually close a task.

Parameter	Description
Task Duration (s)	The value ranges from 10 to 31535999.

Step 10 Click **OK**.

When the criteria for killing sessions are met, the system automatically kills the sessions.

----End

Killing All Sessions

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

**Step 3** Click **SQL Explorer** and then **Auto Flow Control**.

Step 4 Click on the right of **Auto Kill Sessions**. In the displayed dialog box, click **OK**.

Figure 12-40 Auto Kill Sessions
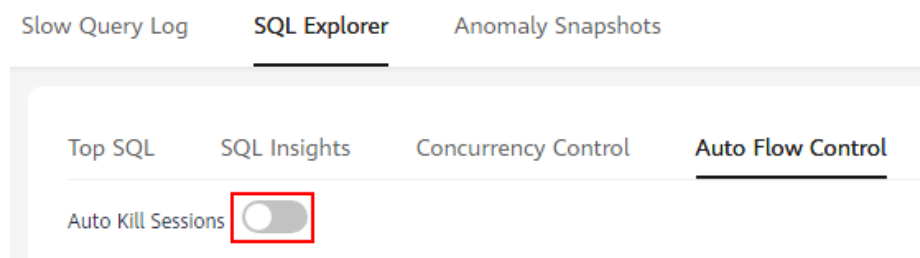
Step 5 Click **Kill All Sessions**.

Figure 12-41 Killing all sessions
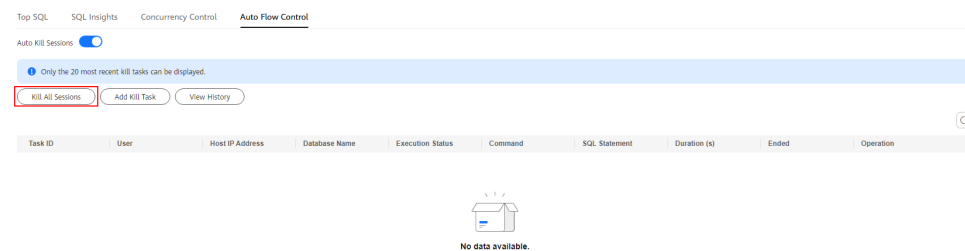
Step 6 In the displayed dialog box, click **OK**.

----End

Viewing History

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 3 Click **SQL Explorer** and then **Auto Flow Control**.

Step 4 Click ⬤ on the right of **Auto Kill Sessions**. In the displayed dialog box, click **OK**.

Figure 12-42 Auto Kill Sessions
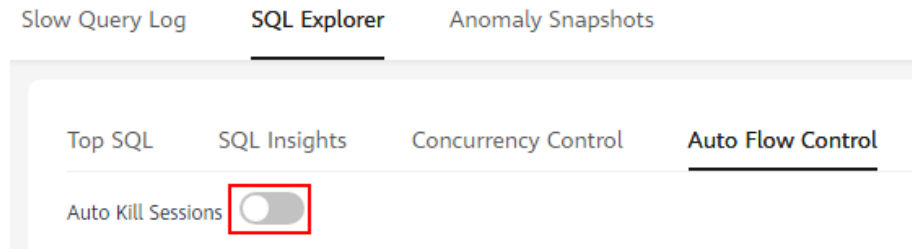
Step 5 Click **View History**.

Figure 12-43 Viewing killed sessions
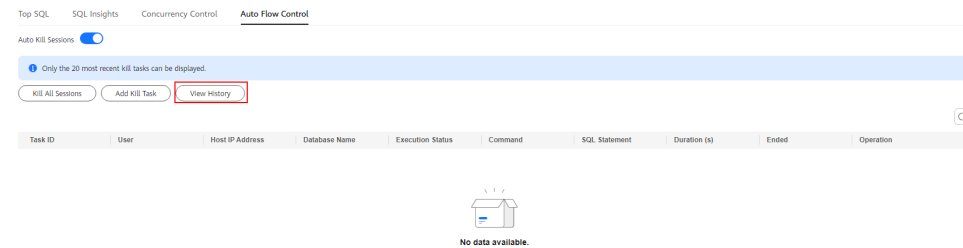
Step 6 In the displayed dialog box, select a time range to view killed sessions within that period.

Figure 12-44 Viewing killed sessions
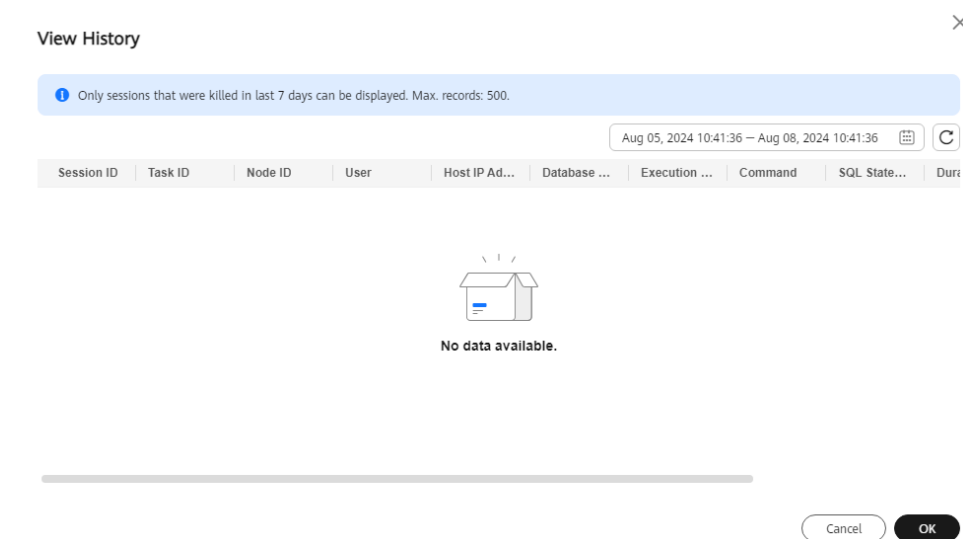
A maximum of 500 session records can be displayed.

----End

13 Parameter Management

13.1 Viewing Parameters of a DB Instance

You can view the parameter settings of your DB instance on the console or through the CLI.

Viewing Parameters of a DB Instance on the Console

📖 NOTE

> You can only view the parameters in the parameter list on the console. To view all parameters of a DB instance, see **Viewing Parameters of a DB Instance Through the CLI**.

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Parameters**.

Step 6 On the **Parameters** tab, view the parameter settings of the DB instance.

Figure 13-1 Viewing parameters of a DB instance

You can search for the desired parameter by parameter name.

----End

Viewing Parameters of a DB Instance Through the CLI

Step 1 Connect to a DB instance.

For details about the connection method, see **Overview**.

Step 2 Run the following command to view all parameter settings of the DB instance:

SHOW VARIABLES;

Run the following command to view the setting of a specified parameter:

SHOW VARIABLES LIKE '*parameter_name***';**

☐ **NOTE**

A percent sign (%) can appear anywhere in *parameter_name* for fuzzy search. Examples:
- Querying all parameters that start with **binlog**:
  SHOW VARIABLES LIKE 'binlog%';
- Querying all parameters that end with **binlog**:
  SHOW VARIABLES LIKE '%binlog';
- Querying all parameters that start with **thread** and end with **size**:
  SHOW VARIABLES LIKE 'thread%size';
- Querying all parameters:
  SHOW VARIABLES LIKE '%';

----End
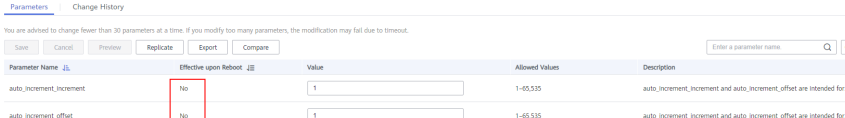
13.2 Modifying Parameters of a DB Instance

You can modify parameters of a DB instance to optimize performance if needed.

Precautions

- To ensure DB instance stability, you can only modify the parameters that are available on the console.
- To apply certain parameter modifications, you need to reboot the DB instance. After you modify a parameter value, check the value in the **Effective upon Reboot** column. You are advised to perform the operation during peak-off hours.

  Figure 13-2 Parameter list

- The value of **validate_password.length** cannot be smaller than that of **validate_password.number_count+validate_password.special_char_count**

+(2 * validate_password.mixed_case_count). Otherwise, the allowed minimum value of **validate_password.length** is used when the parameter template is applied.

- If you want to use a custom parameter template during instance creation, ensure that the value of **validate_password.length** in the template is at most 16. Otherwise, the DB instance fails to be created.

- If you want to use a custom parameter template during instance creation, ensure that the values of **validate_password.mixed_case_count**, **validate_password.number_count**, and **validate_password.special_char_count** are at most 4. Otherwise, the DB instance may fail to be created. The default value **1** is recommended.

- The value of **rds_compatibility_mode** depends on the GaussDB(for MySQL) kernel version.

Modifying Parameters of a DB Instance

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

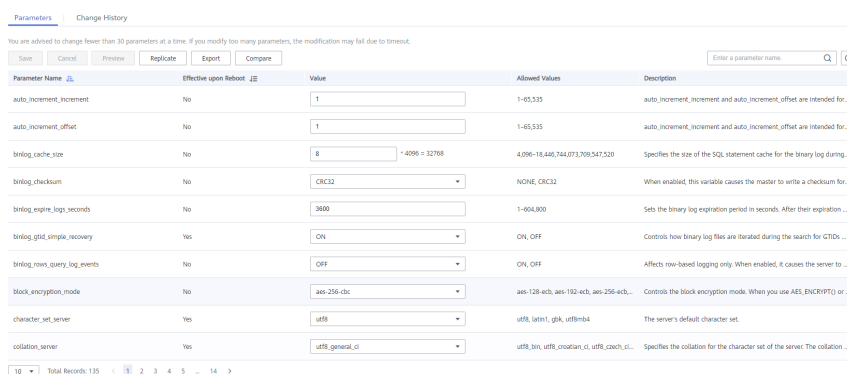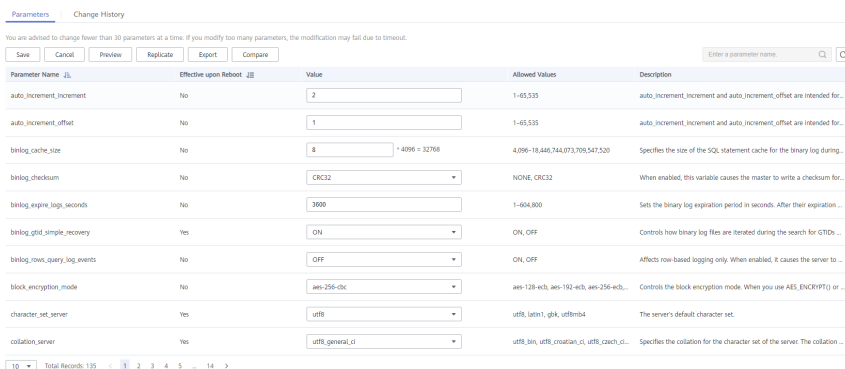**Step 2** Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Parameters**. On the displayed page, modify parameters as required.

Figure 13-3 Modifying parameters of a DB instance

- To save the modifications, click **Save**. In the displayed dialog box, click **Yes**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 6 After the parameters are modified, click **Change History** to view the modification records.

Figure 13-4 Viewing the modification records

----End

Modifying Parameters in a Parameter Template

You can modify parameters in a custom parameter template and then apply the template to multiple DB instances.

Step 1 In the navigation pane, choose **Parameter Templates**. On the **Custom Templates** tab, click the parameter template name.

Step 2 On the displayed **Parameters** page, modify parameters as required.

> **NOTICE**

Figure 13-5 Modifying parameters in a parameter template

- To save the modifications, click **Save**. In the displayed dialog box, click **Yes**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 3 After the parameters are modified, click **Change History** to view the modification records.

Step 4 After a parameter template is modified, the modification will not take effect until the template is applied to DB instances. On the **Parameter Templates** page, locate the parameter template you want to apply and choose **More** > **Apply** in the **Operation** column.

Figure 13-6 Applying a parameter template to a DB instance

Step 5 Select one or more DB instances and click **OK**.

Figure 13-7 Selecting DB instances
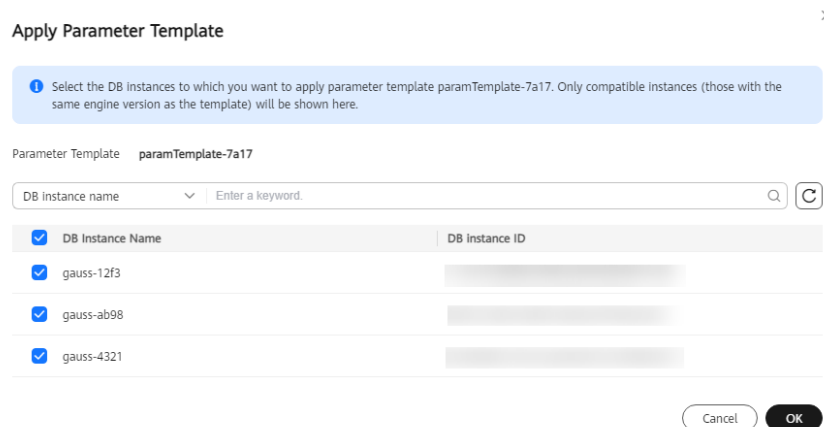
Step 6 After the parameter template is applied, click the DB instance name and check whether the parameters have been modified on the **Parameters** page.

----End

Common Parameters

Table 13-1 Common parameters

Parameter	Description	Reference
time_zone	Specifies the time zone of the server.	**How Do I Change the Time Zone?**
default_password_lifetime	Specifies the global automatic password expiration policy, in days.	**How Do I Configure a Password Expiration Policy for GaussDB(for MySQL) Instances?**
character_set_server	Specifies the server character set.	**How Do I Use the utf8mb4 Character Set to Store Emojis in a GaussDB(for MySQL) Instance?**
collation_server	Specifies the collation for the character set of the server. The collation must match the character set specified by character_set_server. Otherwise, the database cannot be started or restarted.	-
group_concat_max_len	Specifies the maximum permitted result length in bytes for the GROUP_CONCAT() function.	-

Parameter	Description	Reference
max_connections	Specifies the maximum number of concurrent client connections. If this parameter is set to **default**, the parameter value depends on how much memory there is.	**What Is the Maximum Number of Connections to a GaussDB(for MySQL) Instance?**
max_prepared _stmt_count	Limits the total number of prepared statements in the server. Too many statements may cause the server to run out of memory (OOM) and risk denial-of-service attacks. Configure this parameter as needed.	-
innodb_flush_ log_at_trx_co mmit	Controls the balance between strict ACID compliance for commit operations, and higher performance that is possible when commit-related I/O operations are rearranged and done in batches. When this parameter is set to 0, the content of the InnoDB log buffer is written to the log file approximately once per second and the log file is flushed to disk. The default value of 1 is required for full ACID compliance. With this value, the contents of the InnoDB log buffer are written out to the log file at each transaction commit and the log file is flushed to disk. When this parameter is set to 2, the contents of the InnoDB log buffer are written to the log file after each transaction commit and the log file is flushed to disk approximately once per second.	**Viewing Suggestions on GaussDB(for MySQL) Parameter Tuning**
sql_mode	Specifies the SQL server mode.	-
binlog_expire_ logs_seconds	Specifies the binary log expiration period in seconds. After their expiration period ends, binary log files can be automatically removed.	-

APIs

- **Modifying Parameters in a Parameter Template**
- **Querying Parameter Templates**
- **Obtaining Details About a Parameter Template**

13.3 Viewing Suggestions on GaussDB(for MySQL) Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect database performance. This section describes some important parameters for your reference. For details, see **MySQL official website**.

For details on how to modify GaussDB(for MySQL) parameters on the console, see **Modifying Parameters of a DB Instance**.

Performance Parameters

- innodb_flush_log_at_trx_commit

  Default value: **1**

  Function: Controls the balance between strict ACID compliance for commit operations and higher performance.

  1: InnoDB writes data in the log buffer to log files and then flushes the data to distributed storage at each transaction commit. The ACID properties of transactions are ensured.

  0: InnoDB writes data in the log buffer to log files and then flushes the data to distributed storage every second.

  2: InnoDB writes data in the log buffer to the file system cache at each transaction commit, and flushes the data to distributed storage every second.

  Impact: If this parameter is not set to **1**, data security is not guaranteed. One second of transactions can be lost in a crash.

  Recommended value for PoC: **0**. This setting can significantly improve write performance in low concurrency scenarios.

- rds_global_sql_log_bin

  Default value: **OFF**

  Function: Controls whether to enable or disable binlog. GaussDB(for MySQL) uses a shared storage architecture. The primary node and read replicas in a DB instance do not depend on binlogs for data synchronization. You can disable binlog as required.

  OFF: Binlog is disabled. The setting is applied to both existing and new connections without an instance reboot.

  ON: Binlog is enabled. The setting is applied to both existing and new connections without an instance reboot.

  Impact: Enabling or disabling it does not affect your instance.

  Recommended value for PoC: **OFF**. This setting can improve write performance.

NOTE

- In 2.0.42.230601 and earlier versions, binlog is enabled by default. To enable or disable it, you need to configure the **log-bin** parameter and then reboot your instance.
- In 2.0.45.230900 and later versions, binlog is disabled by default. To enable or disable it, you need to configure the **rds_global_sql_log_bin** parameter.

- rds_plan_cache

  Default value: **OFF**

  Function: Controls whether to cache the execution plan of a PREPARE statement.

  ON: The execution plan of the PREPARE statement is cached. The cached execution plan can be reused in the next execution, improving query performance.

  Impact: The query performance of the PREPARE statement is greatly improved, and the select_random_ranges test model of sysbench is significantly enhanced.

  Recommended value for PoC: **ON**. This setting can improve query performance.

  NOTE

  - This feature can be enabled in 2.0.51.240300 and later versions.
  - **rds_plan_cache** uses the memory allocated by the **stmt mem** memory area instead of the **innnodb_buffer** memory.
  - **rds_plan_cache_allow_change_ratio**: Table data change rate caused by query operations such as DML. If the change rate exceeds this parameter value, plan caches become invalid. If this parameter is set to 0, plan caches are not affected by the table data change ratio. They are always valid.

13.4 Introducing the High-Performance Parameter Template

To improve database performance, GaussDB(for MySQL) provides a high-performance parameter template. You can select this template when buying an instance.

This section explains the parameter settings in the high-performance parameter template and how the template enhances performance.

Introduction

The high-performance parameter template is a set of optimized configuration parameters that aim to enhance the performance and reliability of database servers. The parameter settings in the template can be adjusted based on different application scenarios and hardware configurations.

The parameters in the high-performance parameter template are as follows.

Table 13-2 Parameter description

Parameter	Description	Value in the High-Performance Template	Value in the Default Template
innodb_flush_log_at_trx_commit	If this parameter is set to 0, logs are not flushed to disks when transactions are committed. Instead, they are only flushed once per second or when the log buffer (**innodb_log_buffer_size**) is full. This provides low durability but high performance.	0	1
rds_plan_cache	If this parameter is set to **ON**, an execution plan is cached. The next time the same query is executed, the cached execution plan can be reused, which improves the database's query performance.	ON	OFF

Application Scenarios and Potential Risks

Generally, the high-performance parameter template can improve database performance. However, it should be adjusted based on specific application scenarios and hardware configurations.

While the template is designed to enhance performance and reliability of database servers, it does come with some risks during database usage.

- Setting **innodb_flush_log_at_trx_commit** to **0** can improve low-concurrency write performance, but in extreme cases, it may result in data loss of up to one second.

- Setting **rds_plan_cache** to **ON** can improve query performance because the execution plan of a PREPARE statement is cached and the optimizer does not

need to generate an execution plan again. However, it may not be effective in all read/write scenarios.

Constraints

The kernel version of your GaussDB(for MySQL) instance must be 2.0.51.240300 or later.

Usage

You can select the high-performance parameter template when buying an instance.

Figure 13-8 Selecting the high-performance parameter template
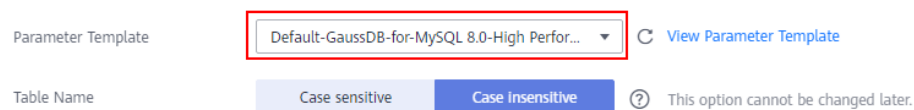
Performance Comparison

Test environment:

- GaussDB(for MySQL) instance specifications: Dedicated, 8 vCPUs | 32 GB
- Kernel version: 2.0.51.240300

Sysbench test process:

- Test scenarios: read-only, write-only, and read/write
- Data volume: 250 tables x 25,000 rows of data in read-only scenarios, 250 tables x 25,000 rows of data in write-only scenarios, and 25 tables x 250,000 rows of data in read/write scenarios
- Performance metric: queries per second (QPS) in 1, 2, 4, 8, 16, 32, 64, 128, and 256 concurrent requests. QPS indicates the number of SQL statements executed by the database per second.

Test results:

- Read-only model

Read-only
250 × 25,000

- **Write-only model**

Write-only
250 × 25,000
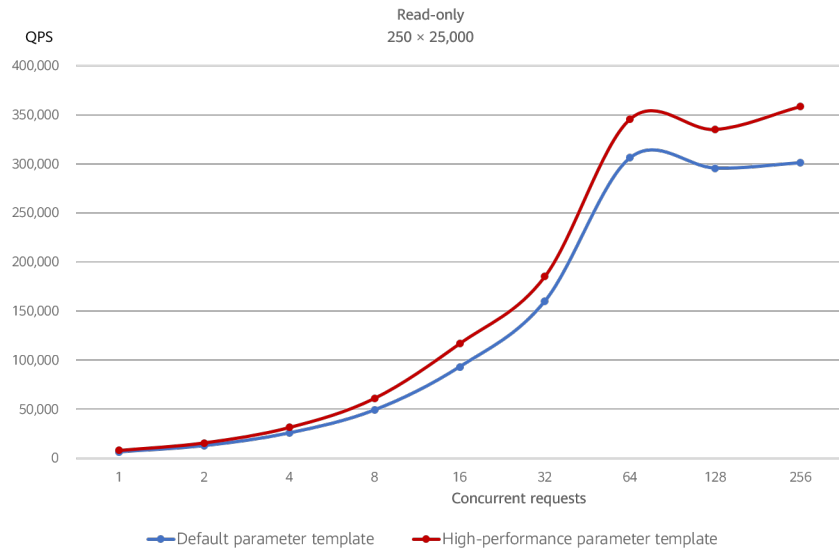
- **Read/Write model**
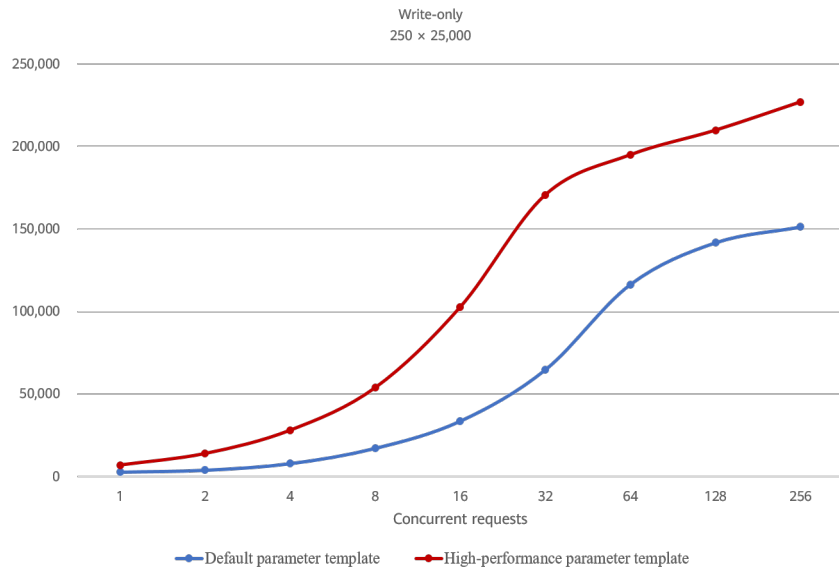
Read/Write
25 × 250,000
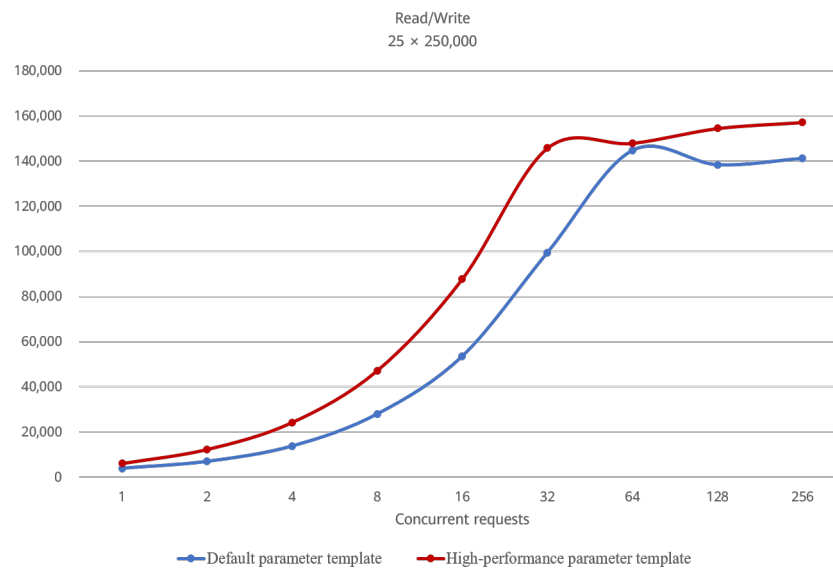
Conclusion: The preceding sysbench test results show that the high-performance parameter template significantly improves the database performance.

13.5 Parameter Template Management

13.5.1 Creating a Custom Parameter Template

You can create custom parameter templates and apply them to one or more DB instances.

There are default parameter templates and custom parameter templates.

Precautions

- Each user can create a maximum of 100 parameter templates.
- All GaussDB(for MySQL) engines share the parameter template quotas.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **Parameter Templates**. On the **Parameter Templates** page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, set required parameters and click **OK**.

Figure 13-9 Creating a parameter template
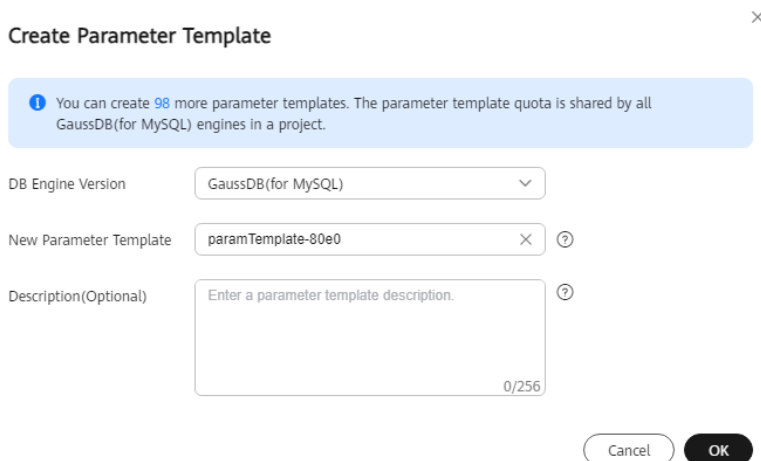
Table 13-3 Parameter description

Parameter	Description
DB Engine Version	Select **GaussDB(for MySQL)**.
New Parameter Template	The template name consists of 1 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.) are allowed.
Description	The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters: >!<"&'=

----End

APIs

- **Creating a Parameter Template**
- **Querying Parameter Templates**
- **Obtaining Details About a Parameter Template**

13.5.2 Applying a Parameter Template

After a parameter template is created or modified, you need to apply it to the desired DB instances.

Precautions

- The parameter **innodb_buffer_pool_size** is determined by the memory. Instances of different specifications have different value ranges. If this parameter value is out of range of the instance to which the parameter template is applied, the maximum value within the range is used.
- A parameter template can be applied only to instances of the same DB engine version.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, apply a default template or a custom template to DB instances.

- To apply a default template, click **Default Templates**, locate a parameter template and click **Apply** in the **Operation** column.

  Figure 13-10 Applying a default parameter template to DB instances
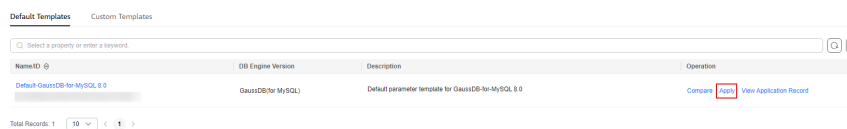
- To apply a custom template, click **Custom Templates**, locate a parameter template and choose **More** > **Apply** in the **Operation** column.

  Figure 13-11 Applying a custom parameter template to DB instances
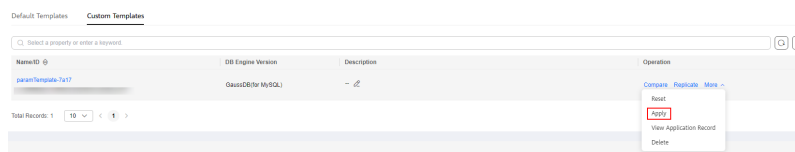
Step 5 In the displayed **Apply Parameter Template** dialog box, select DB instances and click **OK**.

Step 6 After the parameter template is applied, view the name or ID of the DB instance to which the parameter template is applied, application status, application time, and failure cause.

- On the **Default Templates** tab, locate the parameter template and click **View Application Record** in the **Operation** column.

- On the **Custom Templates** tab, locate the parameter template and choose **More** > **View Application Record** in the **Operation** column.

----End

APIs

Applying a Parameter Template

13.5.3 Replicating a Parameter Template

If you already have a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can

replicate that parameter template. You can also replicate the parameter template to generate a new parameter template for future use.

Precautions

- After the parameter template is replicated, the new template will be displayed about 5 minutes later.
- Default parameter templates cannot be replicated, but you can create custom parameter templates based on those default templates.

Replicating a Parameter Template of a DB Instance

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Parameters**. On the **Parameters** page, click **Replicate**.

Step 6 In the displayed dialog box, set required parameters and click **OK**.

Figure 13-12 Replicating a Parameter Template of a DB Instance

- The template name consists of 1 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.) are allowed.
- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:

  >!<"&'=

After the parameter template is replicated, a new template is generated in the list in the **Custom Templates** tab of the **Parameter Templates** page.

----End

Replicating a Custom Parameter Template

Step 1 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be replicated and click **Replicate** in the **Operation** column.

Step 2 In the displayed dialog box, set required parameters and click **OK**.

Figure 13-13 Replicating a Custom Parameter Template

- The template name consists of 1 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.) are allowed.
- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:
  >!<"&'=

  After the parameter template is replicated, a new template is generated in the list in the **Custom Templates** tab of the **Parameter Templates** page.

  ----End

13.5.4 Resetting a Parameter Template

You can reset all parameters in a custom parameter template to their default settings.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be reset and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

Figure 13-14 Confirming the reset

Reset Parameter Template

Are you sure you want to reset this parameter template?

Name	DB Engine Version
gaussdbformysql-170686658640523...	GaussDB(for MySQL)

⚠ All parameters in this parameter template will be reset to their default values. Exercise caution when performing this operation.

Yes   No
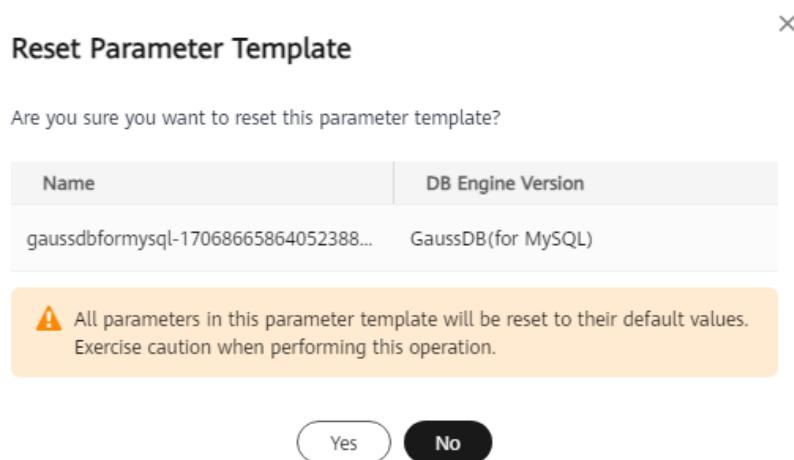
📖 **NOTE**

After you reset a parameter template, view the status of the instance to which the parameter template applies in the instance list. If the status is **Parameter change. Pending reboot**, you must reboot the instance.

----End

13.5.5 Comparing Parameter Templates

You can compare instance parameters with a parameter template to see the differences of parameter settings. You can also compare parameter templates to see the differences of parameter settings.

Comparing Instance Parameters with Those in a Specified Parameter Template

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Figure 13-15 Comparing instance parameters with those in a specified parameter template
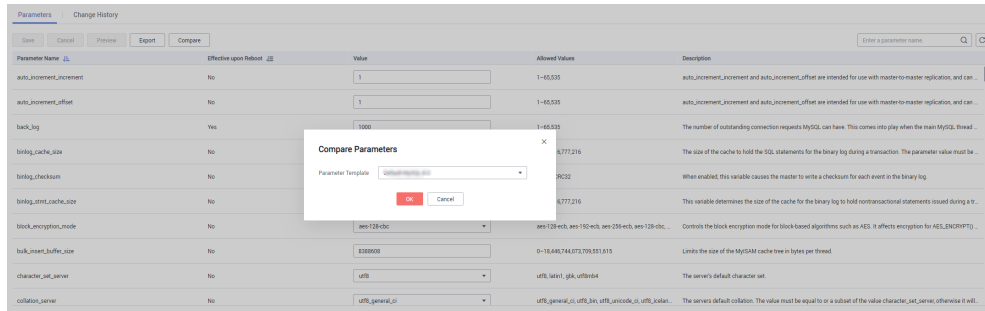
Step 6 In the displayed dialog box, select a parameter template and click **OK** to compare two parameters.

- If their settings are different, the parameter names and values of both parameter templates are displayed.

- If their settings are the same, no data is displayed.

----End

Comparing Parameter Templates

Step 1 On the **Parameter Templates** page, click **Default Templates** or **Custom Templates**. Locate a parameter template and click **Compare** in the **Operation** column.

Step 2 In the displayed dialog box, select a parameter template and click **OK**.

Figure 13-16 Selecting a parameter template to be compared
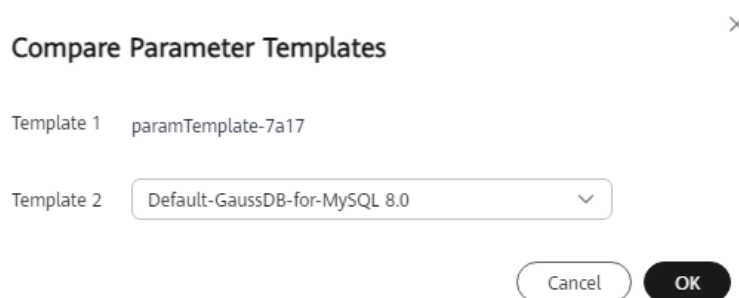
- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

Figure 13-17 Comparing parameter templates

----End

13.5.6 Exporting a Parameter Template

You can export parameter template details (parameter names, values, and descriptions) of an instance to an EXCEL file for review and analysis.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Parameters**. On the displayed page, click **Export** above the parameter list. In the displayed dialog box, enter the file name and click **OK**. You can export parameter template details (parameter names, values, and descriptions) of an instance to an EXCEL file for review and analysis.
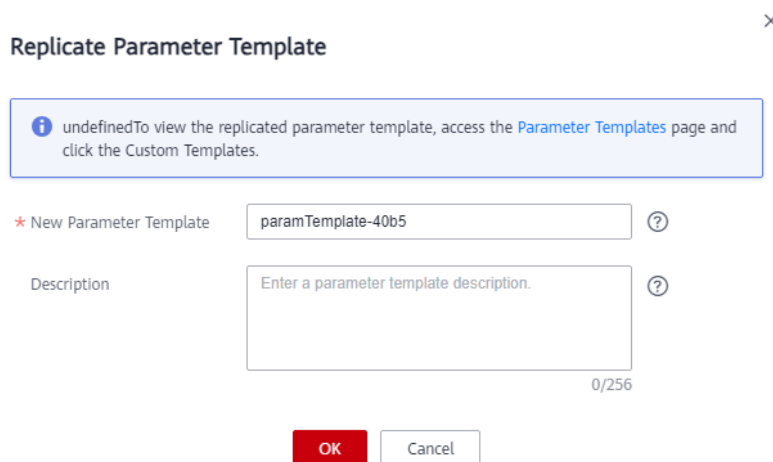
Figure 13-18 Exporting a parameter template

🔲 **NOTE**

The file name can consist of 4 to 81 characters. It must start with a letter and contain only letters, digits, hyphens (-), and underscores (_).

----End

13.5.7 Modifying the Description of a Parameter Template

You can modify the description of a parameter template you have created.

Precautions

You cannot modify the description of any default parameter template.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template for which you want to edit the description and click ✎ in the **Description** column.

Step 5 Enter a new description and click ✔ to submit or ✖ to cancel the change.

- After the modification is successful, you can view the new description in the **Description** column.

- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:

  >!<"&'=

----End

APIs

- **Modifying Parameters in a Parameter Template**
- **Querying Parameter Templates**
- **Obtaining Details About a Parameter Template**

13.5.8 Deleting a Parameter Template

You can delete a custom parameter template that is no longer needed.

Precautions

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template you want to delete and choose **More** > **Delete** in the **Operation** column.

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

----End

APIs

[**Deleting a Parameter Template**](#)

14 Security and Encryption

14.1 Configuring Database Security

Password Strength Requirements

For database password strength requirements on the management console, see the database configuration table in **Buying a DB Instance**.

GaussDB(for MySQL) has a password security policy for newly created database users. Passwords must:

- Consist of at least eight characters.
- Contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.).

When you create instances, your password strength is checked. You can modify the password strength as user **root**. For security reasons, use a password that is at least as strong as the default one.

Account Description

To provide O&M services, the system automatically creates system accounts when you create instances. These system accounts are unavailable to you.

> **NOTICE**
>
> Deleting, renaming, and changing passwords or permissions for these accounts will cause instances to run abnormally. Exercise caution when performing these operations.

- rdsAdmin: a management account with superuser permissions, which is used to query and modify instance information, rectify faults, migrate data, and restore data.
- rdsRepl: a replication account, which is used to synchronize data from the primary node to read replicas.

- rdsBackup: a backup account, which is used to back up data in the background.

- rdsMetric: a metric monitoring account, which is used by watchdog to collect database status data.

- rdsProxy: a database proxy account, which is used for authentication when the database is connected through the proxy address. This account is automatically created when you enable read/write splitting.

14.2 Resetting the Administrator Password

Scenarios

If you forget the password of your database account when using GaussDB(for MySQL), you can reset the password.

If an error occurs on the **root** account, for example, if your **root** account credentials are lost or deleted, you can restore the **root** account permissions through resetting the password.

You cannot reset the administrator password under the following circumstances:

- Your account is frozen.

- The database port is being changed.

- The instance status is **Creating**, **Restoring**, **Rebooting**, **Changing port**, **Changing instance specifications**, **Promoting to primary**, or **Abnormal**.

Precautions

- If you have changed the administrator password of a DB instance, the passwords of the read replicas associated with the instance will also be changed accordingly.

- The time it takes for the new password to take effect depends on the amount of service data currently being processed by the primary node.

- To protect against brute force hacking and improve system security, change your password periodically, such as every three or six months.

- The instance may have been restored from a backup before you reset the administrator password.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the instance for which you want to change the password and choose **More** > **Reset Password** in the **Operation** column.

Alternatively, reset the password using either of the following methods:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click **Reset Password**.

- On the **Instances** page, click the instance name to go to the **Basic Information** page. Expand **Instance Information**. In the **Configuration** area, click **Reset Password** under **Administrator**.

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 obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.
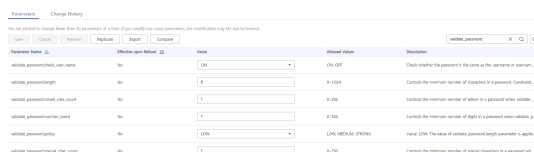
Step 6 In the displayed dialog box, enter and confirm the new password.

The new password must:

- Consist of 8 to 32 characters.

- Contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.).

- Comply with the values of **validate_password** parameters.

  To check the password-related parameters, click the instance name, choose **Parameters** in the navigation pane, and search for **validate_password** in the upper right corner of the page.

Figure 14-1 Checking the password-related parameters

Step 7 Click **OK**.

> **NOTICE**
>
> Keep your password secure. The system cannot retrieve it if it is lost.

----End

14.3 Changing the Security Group of a DB Instance

Scenarios

You can change the security group associated with your DB instance.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Instance Information** area, click **Expand** to expand all instance information.

Step 6 In the **Network Information** area, click **Modify** under **Security Group**.

Step 7 In the displayed dialog box, select a new security group and click **OK**.

----End

APIs

[Changing a Security Group](#)

14.4 Configuring SSL for a DB 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. SSL:

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

By default, SSL is enabled for new DB instances. Enabling SSL increases the network connection response time and CPU usage, and you are advised to evaluate the impact on service performance before enabling SSL.

You can use a client to connect to your DB instance through a non-SSL or SSL connection.

- If SSL is enabled for your DB instance, you can connect to your DB instance using SSL, which is more secure.
- If SSL is disabled, you can only connect to your DB instance using a non-SSL connection.

> **NOTICE**
>
> Enabling or disabling SSL will cause the instance to be rebooted immediately and temporarily unavailable. You are advised to perform this operation during off-peak hours.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Instance Information** area, click **Expand** to expand all instance information.

Step 6 In the **Configuration** area, click ⬤ under **SSL**.

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

Step 8 Wait for some seconds and check that SSL has been enabled on the **Basic Information** page.

To disable SSL, click ⬤. In the displayed dialog box, click **OK**.

----End

APIs

Enabling or Disabling SSL

14.5 Enabling TDE for a DB Instance

Transparent Data Encryption (TDE) performs real-time I/O encryption and decryption on data files. Data is encrypted before being written to disks and is decrypted when being read from disks to memory. This effectively protects the security of databases and data files.

Supported Regions

CN South-Guangzhou

Constraints on Usage

- To enable TDE, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

- To configure TDE, you must have the iam:agencies:createServiceLinkedAgencyV5 permission. If you do not have this permission, **create a custom policy**.

- You need to enable Key Management Service (KMS) for your DB instance first. The data keys used for encryption are generated and managed by KMS. GaussDB(for MySQL) does not provide any keys or certificates required for encryption.

- To enable TDE, the kernel version of GaussDB(for MySQL) instances must be 2.0.47.231100 or later.
- Your DB instance must be billed at a pay-per-use or yearly/monthly basis.
- The instance type must be single-node or primary/standby deployment.
- TDE can be enabled only when a DB instance is created. After the instance is created, TDE cannot be enabled or disabled.
- TDE encrypts instance data, including full backups but excluding incremental backups.
- After TDE is enabled, the cryptographic algorithm cannot be changed later.
- Only instance-level encryption is supported.
- After TDE is enabled for a DB instance, you cannot:
  - Enable cross-region backup for the DB instance.
  - Restore the data of the DB instance to an existing DB instance.

Procedure

Step 1 Go to the **Buy DB Instance** page.

Step 2 On the displayed page, set **TDE** to **Enabled** and select the corresponding cryptographic algorithm.

Figure 14-2 Enabling TDE

Step 3 After the DB instance is created, click the DB instance name to go to the **Basic Information** page and view the **TDE** status.

----End

15 Application Lossless and Transparent (ALT)

15.1 Introducing ALT

Database sessions may be interrupted when a read replica is promoted to primary, a minor version is upgraded, or specifications are changed. Applications need to check session statuses and react to changes by determining: whether a database connection or transaction has been interrupted, how to compensate for transactions, and how to rebuild session contexts.

To address these issues, GaussDB(for MySQL) provides ALT, which prevents database connection and transaction interruptions during database system switchover. There is no need to compensate for transactions or rebuild session contexts, ensuring application continuity.

Architecture

Figure 15-1 Architecture
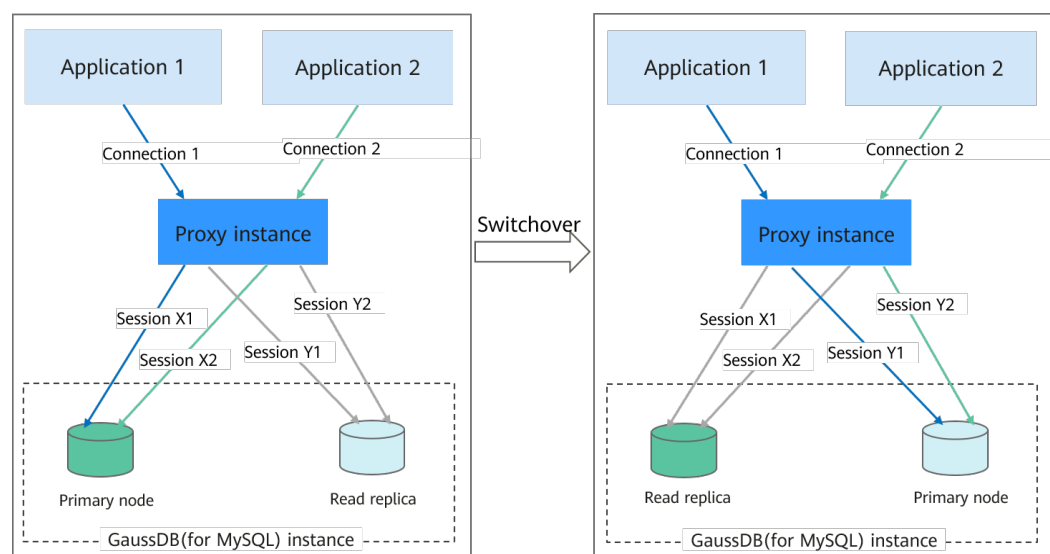
ALT can be enabled for your application connections. When you connect to a proxy instance and then promote a read replica to primary, change specifications, or upgrade the minor version, the system can replicate your backend sessions. Once a secure transaction boundary is reached, backend sessions will be fully cloned to the destination node, and workloads do not even notice.

◫ NOTE

A secure transaction boundary refers to the status that a transaction in the current session has been committed but the next transaction is not started. A secure transaction boundary can be reached in any of the following situations:

- Each statement in a transaction block with autocommit enabled is executed.
  start transaction;
  DML;
  commit;

- The commit operation is complete with autocommit disabled.

- A single DML or DDL statement is executed.

- The lock is released when a table lock, backup lock, or user-defined lock is used.

Precautions

Table 15-1 Precautions

Category	Precaution
Version constraints	- The kernel version of the GaussDB(for MySQL) instance must be 2.0.54.240600 or later. - The kernel version of the proxy instance must be 2.24.06.000 or later.

Category	Precaution
Usage constraints	- To use ALT, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. - The GaussDB(for MySQL) instance has at least one read replica. A proxy instance has been created and the GaussDB(for MySQL) instance must be connected through the proxy address. - Proxy instances in read-only mode do not support ALT. - Proxy instances in primary/standby mode do not support ALT. - Single-node or multi-primary GaussDB(for MySQL) instances do not support ALT. - When you enable ALT for the first time, the GaussDB(for MySQL) instance will reboot. Enabling or disabling ALT will cause a proxy instance to reboot. Once ALT is disabled for all proxy instances, the GaussDB (for MySQL) instance will also reboot. - The transaction draining timeout interval for ALT is controlled by **rds_tac_drain_timeout**. This parameter defaults to 5s and ranges from 1s to 60s. - Increase this interval for heavy workloads, numerous prepared statements, or time-consuming transactions. - Decreasing this interval is not recommended. If there are connections that do not drain transactions within the configured transaction draining timeout interval, ALT does not take effect for these connections. - During an ALT switchover, standby connections will be established on the new host for a brief period, equal in number to those on the original primary node. Ensure that the maximum number of connections of the GaussDB(for MySQL) instance is at least twice the current number of connections. To change the maximum number of connections, you need to evaluate the instance specifications and memory usage. For details, see **What Is the Maximum Number of Connections to a GaussDB(for MySQL) Instance?** - ALT supports prepared statements. During a switchover, the contexts of prepared statements are rebuilt. If there are a large number of prepared statements, the switchover success rate may be affected. - You are advised to perform an ALT switchover during off-peak hours. If the primary node and read replicas are overloaded, the switchover success rate may be affected. - For details about syntax and function constraints of proxy instances, see **Precautions for Proxy Instances**.

Category	Precaution

Unsupported functions	● Enabling ALT makes your instance lose support for some system variable values.
	  – *innodb_ft_user_stopword_table*: It can only be set to **NULL**.
	  – *transaction_write_set_extraction*: It can only be set to **OFF**.
	  – *profiling*: It cannot be set to **1** or **ON**.
	● ALT does not support Transparent Data Encryption (TDE).
	● ALT is unavailable when any of the following proxy capabilities is enabled:
	  – Session-level connection pool
	  – Any column containing more than 16 MB of data
	  – Prepared statement cache
	● ALT does not support temporary tables created by users.
	● ALT is not supported in the following scenarios where a secure transaction boundary cannot be reached:
	  – InnoDB transaction blocks are not committed in a timely manner.
	  – There are unreleased table locks, user locks, backup locks, and binlog locks.
	  – XA transactions are not committed or rolled back.
	● ALT will be likely to fail if a switchover, minor version upgrade, or specification change occurs frequently within a short period of time.
	● If ALT is enabled, prepared statements cannot be transferred in the following scenarios:
	  – The cursor is opened and not closed in a prepared statement.
	  – The variable of a prepared statement has saved the LONG_DATA type.

15.2 Enabling ALT

This section describes how to enable ALT.

Constraints

For details, see **Precautions**.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click a proxy instance to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click next to **ALT**.

Figure 15-2 Enabling ALT
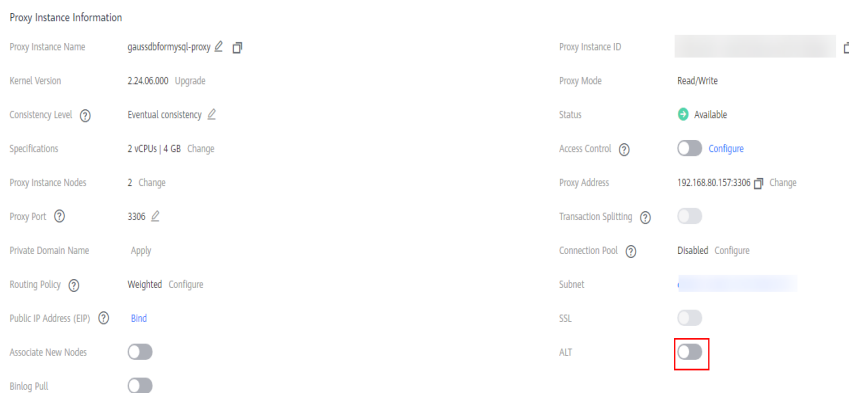
Step 8 In the displayed dialog box, click **Yes**.

> **NOTE**
>
> - If ALT is enabled for the first time, the GaussDB(for MySQL) instance and proxy instance will be rebooted immediately. During the reboot, the instances are unavailable.
>
> - Rebooting an instance will clear the cached memory in it. You are advised to reboot it during off-peak hours.
>
> - To disable ALT, click . ALT must be disabled for all proxy instances. Disabling ALT will cause the proxy instances to reboot. Once ALT is disabled for all proxy instances, the GaussDB (for MySQL) instance will also reboot.

----End

15.3 Example: Using ALT to Promote a Read Replica to Primary

This section describes how to use ALT to promote a read replica to primary. The process for minor version upgrades and specification changes is similar.

The process for using ALT to promote a read replica to primary is as follows:

Step 1: Buy a GaussDB(for MySQL) Instance

Step 2: Create a Proxy Instance

Step 3: Enable ALT

Constraints

If workloads are interrupted, see **Precautions**.

Step 1: Buy a GaussDB(for MySQL) Instance

For details, see **Buying a DB Instance**.

Step 2: Create a Proxy Instance

For details, see **Step 1: Create a Proxy Instance**.

Step 3: Enable ALT

For details, see **Enabling ALT**.

Step 4: Connect Your Application to the Proxy Instance

For details, see **Step 4: Use the Proxy Address to Connect to Your GaussDB(for MySQL) Instance**.

Step 5: Promote a Read Replica to Primary

For details, see **Promoting a Read Replica to Primary**.

Step 6: Test the ALT Effect

If ALT is enabled and you promote a read replica to primary using sysbench, tpcc-mysql, or a MySQL client that is connected to the proxy address, your database only freezes briefly.

The following figures show you what effect ALT has when you promote a read replica to primary using sysbench, tpcc-mysql, and a MySQL client.

- **Promoting a read replica to primary using sysbench**

- **Promoting a read replica to primary using tpcc-mysql**

- **Promoting a read replica to primary using the MySQL CLI**

  As shown in the following figure, user-defined variables, session variables, and databases remain unchanged before and after you promote a read replica to primary.

  The following figure shows how the transaction draining timeout interval determines whether ALT is available for the current session.

16 HTAP Analysis (Standard Edition)

16.1 What Is HTAP of Standard Edition

Hybrid Transactional and Analytical Processing (HTAP) is a data architecture that handles both online transactional processing (OLTP) and online analytical processing (OLAP) workloads.

It uses the column-based storage engine and Single Instruction Multiple Data (SIMD) for parallel compute. In massive data analysis scenarios, HTAP analysis provided by GaussDB(for MySQL) frees you from having to independently maintain data extraction and synchronization links. It reduces data management costs and provides simple and efficient real-time data analysis.

HTAP of Standard Edition is developed based on the open-source StarRocks.

Product Architecture

HTAP instances are deployed on ECSs and use extreme SSDs or ultra-high I/O disks.

You can **enable binlog of your GaussDB(for MySQL) instance** to synchronize data and operations to HTAP instances. Synchronized operations include inserting table, deleting tables, and changing table structures. After data is synchronized to an HTAP instance, you can access the HTAP instance through its private IP address for data analysis.

An HTAP instance of the standard edition provides frontend (FE) and backend (BE) nodes. The FE nodes manage metadata, manage client connections, and plan and schedule queries. Each FE node stores and maintains a complete metadata backup in the memory to ensure data consistency between FE nodes. The BE nodes are for data storage and SQL computing.

Figure 16-1 Product architecture
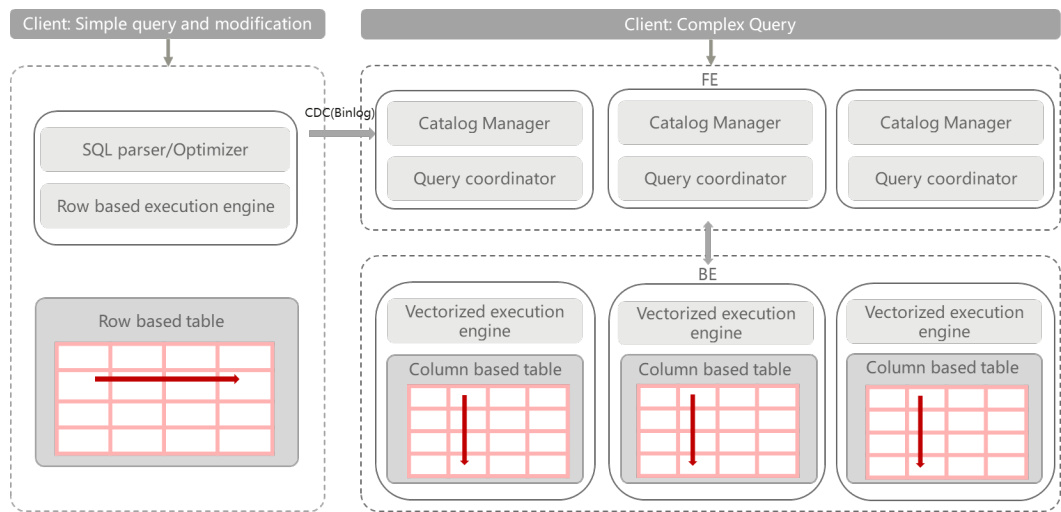
There are three roles for FE nodes.

Figure 16-2 FE node roles

- The fe-leader nodes read and write metadata. The fe-follower and fe-observer nodes can only read metadata and route write requests for metadata to the fe-leader nodes. The fe-leader nodes update the metadata and synchronize the metadata changes to the fe-follower and fe-observer nodes.

- The fe-follower nodes can only read metadata.

- The fe-observer nodes synchronize and replay logs from the fe-leader nodes to update metadata. The fe-observer nodes are used to increase query concurrency of a cluster.

Main Features

- Massively Parallel Processing (MPP) architecture

  Multiple nodes are used to execute queries in parallel.

- High performance

  It supports vectorized engines and CBO optimizers and excels in queries for large and wide tables and multi-table join operations.

- Standard SQL

  Query statements comply with the SQL-92 standard.

- Data compression for storage

  Column-based storage and data compression greatly reduce your storage costs for any given set of conditions.
- Aggregation of multiple data sources

  Data in multiple GaussDB(for MySQL) databases can be synchronized to a given HTAP instance.

Billing

Standard HTAP instances are in OBT and are free.

Precautions

To improve the stability and security of standard HTAP instances, GaussDB(for MySQL) has certain constraints in place.

1. When you query data in an HTAP instance, the names of databases, tables, views, users, and roles are case sensitive, but the names of columns and partitions are case insensitive.

2. Tables in a GaussDB(for MySQL) database can be synchronized only when primary keys are defined for the tables.

3. Some DDL statements executed on GaussDB(for MySQL) instances cannot be synchronized to HTAP instances, which may cause synchronization failures or data inconsistencies.

   The DDL statements that can be synchronized and cannot be synchronized are as follows:

   – DDL statements that can be synchronized

Table 16-1 DDL statements that can be synchronized

DDL Name	SQL Example
Creating a table	CREATE TABLE tbl_name (c_id int not null, c_d_id integer not null, primary key (c_id));
Dropping a table	DROP TABLE tbl_name;
Renaming a table	RENAME TABLE tbl_name to new_tbl_name; ALTER TABLE tbl_name RENAME TO new_tbl_name;
Clearing table data	TRUNCATE TABLE tbl_name;
Altering table comments	ALTER TABLE tbl_name COMMENT='test';
Adding a column (non-primary key column)	ALTER TABLE tbl_name ADD c_varchar varchar(2000) AFTER c_tinytext;

DDL Name	SQL Example
Deleting a column (non-primary key column)	ALTER TABLE tbl_name DROP c_vchar;
Changing the type and sequence of a column (non-primary key column)	ALTER TABLE tbl_name CHANGE c_vchar c_vchar varchar(2000) default 'test' AFTER c_tinytext; **CAUTION** The column name and default value cannot be changed. ALTER TABLE tbl_name MODIFY c_vchar varchar(2100) default 'test' AFTER c_tinytext; **CAUTION** The default value cannot be changed.

- DDL statements that cannot be synchronized

  After a database synchronization task, only tables and data can be synchronized. Operations for databases, tablespaces, indexes, foreign keys, functions, stored procedures, triggers, views, partitions (DELETE operations), primary keys (INSERT/DELETE/ALTER operations), transactions, users, roles, permissions, and events cannot be synchronized.

  Table 16-2 list index-related operations that cannot be synchronized.

  Table 16-2 Index-related operations that cannot be synchronized

DDL Name	SQL Example
Adding an index	ALTER TABLE tbl_name ADD INDEX name;
Renaming an index	ALTER TABLE tbl_name RENAME INDEX old_index_name TO new_index_name;
Dropping an index	DROP INDEX name ON table;
Adding a full-text index	CREATE FULLTEXT INDEX name ON table(column);
Adding a spatial index	ALTER TABLE geom ADD SPATIAL INDEX(g);
Modifying the type of an index	ALTER TABLE tbl_name DROP INDEX i1, ADD INDEX i1(key_part,...) USING BTREE;
Adding an index constraint	ALTER TABLE tbl_name ADD CONSTRAINT UNIQUE USING BTREE (column); ALTER TABLE tbl_name ADD CONSTRAINT UNIQUE USING HASH(column);

  Table 16-3 list partitioned table-related operations that cannot be synchronized

Table 16-3 Partitioned table-related operations that cannot be synchronized

DDL Name	SQL Example
Analyzing a table partition	ALTER TABLE {db}.tp ANALYZE PARTITION p0;
Checking a table partition	ALTER TABLE {db}.tp CHECK PARTITION p0;
Optimizing a table partition	ALTER TABLE {db}.tp OPTIMIZE PARTITION p0;
Re-building a table partition	ALTER TABLE {db}.tp REBUILD PARTITION p0;
Repairing a table partition	ALTER TABLE {db}.tp REPAIR PARTITION p0;
Creating a database	CREATE DATABASE ddl_test_2;
Modifying a row format	ALTER TABLE tbl_name ROW_FORMAT = row_format;
Setting persistent table statistics	ALTER TABLE tbl_name STATS_PERSISTENT=0, STATS_SAMPLE_PAGES=20,STATS_AUTO_RECALC =1, ALGORITHM=INPLACE, LOCK=NONE;
Setting a table character set	ALTER TABLE tbl_name CHARACTER SET = charset_name;
Converting a table character set	ALTER TABLE tbl_name CONVERT TO CHARACTER SET charset_name;
Optimizing a table	OPTIMIZE TABLE tbl_name;
Rebuilding a table using the FORCE option	ALTER TABLE tbl_name FORCE;
Rebuilding a table without data	ALTER TABLE tbl_name ENGINE=InnoDB;
Renaming a tablespace	ALTER TABLESPACE tablespace_name RENAME TO new_tablespace_name;
Adding a table partition	ALTER TABLE {db}.tp ADD PARTITION (PARTITION p3 VALUES LESS THAN (2006));
Setting the default character set and verification rules for a table	ALTER TABLE tbl_name DEFAULT CHARACTER SET = utf8 COLLATE = utf8_general_ci;

Table creation statements cannot contain CHECK or table options.

During data synchronization, operations in **Table 16-4** may cause data inconsistency between HTAP instances and GaussDB(for MySQL) instances. You should avoid these operations.

They do not affect data query and analysis on HTAP instances.

Table 16-4 DDL operations that result in data inconsistency

DDL Name	SQL Example
Deleting a primary key	ALTER TABLE tbl_name DROP PRIMARY KEY;
Adding a primary key	ALTER TABLE {db}.t1 ADD PRIMARY KEY (id);
Adding a primary key and deleting a primary key	ALTER TABLE tbl_name DROP PRIMARY KEY, ADD PRIMARY KEY (column);
Setting a primary key to NULL	ALTER TABLE tbl_name MODIFY COLUMN key_column_name data_type;
Changing the type of a primary key	ALTER TABLE tbl_name MODIFY COLUMN key_column_name data_type not null;
Adding a foreign key	ALTER TABLE tbl1 ADD CONSTRAINT fk_name FOREIGN KEY index (col1)REFERENCES tbl2(col2) referential_actions;
Deleting a foreign key	ALTER TABLE tbl DROP FOREIGN KEY fk_name;
Adding a column **NOTE** Common columns can be added. If columns contain the following default values, they cannot be added. ● Double quotation marks (") ● Functions, character strings, and identifiers that cannot be found in HTAP instances	ALTER TABLE tbl_name ADD COLUMN column_name column_definition c VARCHAR(10) DEFAULT (CONCAT('1', '2'));

DDL Name	SQL Example
Setting the default value of a column **NOTE** If columns contain the following default values, you cannot reset default values for the columns. • Double quotation marks (") • Functions, character strings, and identifiers that cannot be found in HTAP instances	ALTER TABLE tbl_name ALTER COLUMN col SET DEFAULT literal;
Changing NULL in tables to NOT NULL	ALTER TABLE tbl_name MODIFY COLUMN column_name data_type NOT NULL;
Changing the column name and type at the same time	ALTER TABLE t1 CHANGE b b1 VARCHAR(100);
Changing the name of a column	ALTER TABLE t1 RENAME COLUMN a TO b;
Creating a table without a primary key	ALTER TABLE t1 ADD COLUMN (c2 INT GENERATED ALWAYS AS (c1 + 1)STORED);
Adding a STORED derived column	ALTER TABLE {db}.t1 ADD COLUMN (st2 INT GENERATED ALWAYS AS (c2 + 2)STORED), ALGORITHM=COPY;
Adding a VIRTUAL derived column	ALTER TABLE t1 ADD COLUMN (c2 INT GENERATED ALWAYS AS (c1 + 1)VIRTUAL);
Dropping a table partition	ALTER TABLE {db}.tp DROP PARTITION p4;
Discarding a table partition	ALTER TABLE {db}.tp DISCARD PARTITION p2 TABLESPACE;
Importing a table partition	ALTER TABLE {db}.tp IMPORT PARTITION p2 TABLESPACE;
Truncating a table partition	ALTER TABLE {db}.tp TRUNCATE PARTITION p2;
Truncating a partitioned table	TRUNCATE {db}.tp;

DDL Name	SQL Example
Coalescing table partitions	ALTER TABLE {db}.tp_hash COALESCE PARTITION 2;
Reorganizing table partitions	ALTER TABLE {db}.tp REORGANIZE PARTITION p0,p1,p2,p3 INTO (…);
Exchanging table partitions	ALTER TABLE {db}.tp EXCHANGE PARTITION p0 WITH TABLE {db}.tp2;
Removing a table partition	ALTER TABLE {db}.tp REMOVE PARTITIONING;
Using a REPLACE clause	CREATE OR REPLACE TABLE;

4. The names of the databases and tables to be synchronized cannot contain Chinese characters.

5. To improve performance, you can use the following methods to optimize queries:
   – Simplify SQL statements by reducing invalid calculations, deleting unused fields, and avoiding SELECT.
   – Instead of querying all columns, delete those that are unnecessary.

6. Tables to be synchronized use the OLAP engine and primary key model by default.

7. After connecting to a standard HTAP instance, run the following command to view the databases synchronized from GaussDB(for MySQL) to the standard HTAP instance and the synchronization status.

   As this command queries the binlog information in GaussDB(for MySQL), it may consume the hourly query quota. You are advised to run this command at most once a minute.
   ```
   show sync job;
   ```

16.2 Connecting to an HTAP Instance for Complex OLAP Queries

You can let an application directly connect to an HTAP instance to enable complex OLAP queries.

Procedure

Step 1: Buy a Standard HTAP Instance

Step 2: Synchronize GaussDB(for MySQL) Data to the Standard HTAP Instance

Step 3: Connect to the HTAP Instance for OLAP Queries

Prerequisites

● Parameters have been configured for a GaussDB(for MySQL) instance according to the following table.

Table 16-5 Parameter description

Parameter	Value	How to Modify
default_authentication_plugin	mysql_native_password	**Modifying Parameters of a DB Instance**
binlog_expire_logs_seconds	86400 **NOTE** It is recommended that the binlog retention period be greater than one day. 86,400s = 60 (seconds) x 60 (minutes) x 24 (hours). This prevents incremental replication failures caused by a short binlog retention period.	**Modifying Parameters of a DB Instance**
log_bin **NOTE** To use this parameter, ensure that the GaussDB(for MySQL) kernel version is earlier than 2.0.45.230900.	ON	**How Do I Enable and View Binlog of My GaussDB(for MySQL) Instance?**
rds_global_sql_log_bin **NOTE** To use this parameter, ensure that the GaussDB(for MySQL) kernel version is 2.0.45.230900 or later.	ON	**How Do I Enable and View Binlog of My GaussDB(for MySQL) Instance?**
binlog_format	ROW	Run the **SHOW VARIABLES;** command to check the parameter value. If you need to change the parameter value, contact customer service.
binlog_row_image	FULL	Run the **SHOW VARIABLES;** command to check the parameter value. If you need to change the parameter value, contact customer service.

Parameter	Value	How to Modify
log_bin_use_v1_row_ev ents	OFF	Run the **SHOW VARIABLES;** command to check the parameter value. If you need to change the parameter value, contact customer service.

- Databases and tables have been created for the GaussDB(for MySQL) instance.

Step 1: Buy a Standard HTAP Instance

1. **Log in to the management console.**

2. Click ⊙ in the upper left corner and select a region and project.

3. Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

4. On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

5. In the navigation pane, choose **HTAP Analysis**. On the displayed page, click **Create HTAP Instance**.

6. In the **DB Instance Information** area, check the current GaussDB(for MySQL) instance information.

Figure 16-3 Checking the GaussDB(for MySQL) instance information

7. Set parameters for the HTAP instance.

Figure 16-4 Creating a standard HTAP instance

Table 16-6 Parameter description

Parameter	Description
Billing Mode	Select **Pay-per-use**.
HTAP Instance Type	Select **Single** or **Cluster**. ● **Single**: There is only one FE node and one BE node. It is used only for function experience and testing and does not ensure **SLA**. ● **Cluster**: There are at least three FE or BE nodes and at most 10 FE or BE nodes.

Parameter	Description
Storage Type	Select **Extreme SSD** or **Ultra-high I/O**. ● **Extreme SSD**: uses a 25GE network and RDMA to provide you with up to 1 million random read/write performance per disk and low latency per channel. ● **Ultra-high I/O**: uses multi-disk striping to balance I/O loads among multiple disks, improving read/write bandwidth. The maximum throughput is 1.7 GB/s.
AZ Type	Only single-AZ is available.
AZ	Select an AZ as needed.
Time Zone	Select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.
Instance Specifications	Only general-enhanced is available.
Backend Node Specifications	Select the BE node specifications. The BE nodes are for data storage and SQL computing.
Backend Node Storage (GB)	Select the storage for BE nodes. The default storage is 50 GB and can be expanded to up to 32,000 GB.
Backend Nodes	● A single-node instance has only one BE node. ● A cluster instance has 3 to 10 BE nodes. You can apply for a maximum of 10 nodes at a time.
Frontend Node Specifications	Select the FE node specifications. The FE nodes manage metadata, manage client connections, and plan and schedule queries.
Frontend Node Storage (GB)	Select the storage for FE nodes. The default storage is 50 GB and can be expanded to up to 1,000 GB.
Frontend Nodes	● A single-node instance has only one FE node. ● A cluster instance has 3 to 10 FE nodes. You can apply for a maximum of 10 nodes at a time.
Administrator	The default username is **root**.
Administrator Password	The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.). Enter a strong password and periodically change it to improve security and defend against threats such as brute force cracking attempts.

Parameter	Description
Confirm Password	Enter the administrator password again.

8. After configuration, click **Next**.

9. Confirm the configuration and click **Submit**.

10. On the HTAP instance list page, view and manage the HTAP instance.

Step 2: Synchronize GaussDB(for MySQL) Data to the Standard HTAP Instance

1. On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

2. In the navigation pane, choose **HTAP Analysis**.

3. Click the name of an HTAP instance to access the **Basic Information** page.

4. In the navigation pane, choose **Data Synchronization**. On the displayed page, click **Create Synchronization Task**.

5. Configure required parameters.

Figure 16-5 Creating a synchronization task

– Currently, the databases whose name is Chinese cannot be synchronized. The destination database and task name cannot contain Chinese characters, and the destination database name must contain at least three characters.

– **Synchronize Read Replica Data**: Select **Yes**. You need to select a read replica. Full data is synchronized from the selected read replica, preventing query load on the primary node during a full synchronization. If there is only one read replica, this node is selected by default. During a full synchronization, ensure that the read replica is available, or the

synchronization will fail and you will need to perform the synchronization again.

- **Synchronization Task Name**: The name can contain 3 to 128 characters. Only letters, digits, underscores (_) are allowed.

- **Destination Database**: The name can contain 3 to 128 characters. Only letters, digits, underscores (_) are allowed.

- **Database to be Synchronized**: Select a database that the data will be synchronized to from the drop-down list. You can modify the database parameters as required.

Figure 16-6 Setting databases to be synchronized

- **Synchronization Scope**: Select **All Tables** or **Some Tables**.

- **Blacklist and Whitelist**: If **Synchronization Scope** is set to **Some Tables**, you need to configure tables for the blacklist or whitelist. Set the blacklist and whitelist for the selected tables.

  ☐ NOTE

  - You can set either a blacklist or a whitelist. If you select the whitelist, only the tables in the whitelist are synchronized.

  - The tables to be synchronized must contain primary keys or a non-empty unique key, or they cannot be synchronized to the HTAP instance.

  - Extra disk space may be used during backend data combination and query. You are advised to reserve 50% of the disk space for the system.

  - When setting the table blacklist or whitelist, you can enter multiple tables in the search box at a time. The tables can be separated by commas (,), spaces, or line breaks (\n). After entering multiple tables, you need to click 🔍 . These tables will be selected by default and displayed in the **Selected Table** area.

- **Configure Table Operations**: Enable or disable it as required.

  ▪ If you select **Enabled**:

1) Select a synchronized table on the left and perform operations on its columns. The operations include order by, key columns, distributed by, partition by, data_model, buckets, replication_num, and enable_persistent_index. Multiple operations are separated by semicolons (**;**).

For details about the syntax, see **Table 16-7**.

Table 16-7 Operation syntax

Operation Type	Syntax
order by	order by (column1, column2) or order by column1,column2
key columns	key columns (column1, column2) or key columns column1,column2
distributed by	distributed by (column1, column2) buckets 3 **NOTE** **buckets** is optional. If it is not set, the default value is used.
partition by	There are expression partitions and list partitions. For details, see the partition syntax example.
data_model	Specifies the table type. The value can be primary key, duplicate key, or unique key. Syntax: data_model=primary key, data_model=duplicate key, or data_model=unique key
replication_num	replication_num=3 **NOTE** The value cannot exceed the number of BE nodes, or the verification fails.
enable_persistent_index	Specifies whether to make the index persistent. Syntax: enable_persistent_index=true or enable_persistent_index=false
Combined scenario	data_model=duplicate key;key columns column1, column2;

Partition syntax example:

You only need to set a partition expression (time function expression or lis expression) when creating a table. During data import, an HTAP instance automatically creates partitions based on the data and the rule defined in the partition expression.

Partitioning based on a time function expression: If data is often queried and managed based on a continuous date range,

you only need to specify a partition column of the date type (DATE or DATETIME) and a partition granularity (year, month, day, or hour) in the time function expression. An HTAP instance automatically creates partitions and sets the start time and end time of the partitions based on the imported data and partition expression.

Syntax:

```
PARTITION BY expression
…
[ PROPERTIES( 'partition_live_number' = 'xxx' ) ]

expression ::=
    { date_trunc ( <time_unit> , <partition_column> ) |
      time_slice ( <partition_column> , INTERVAL <N> <time_unit> [ ,
boundary ] ) }
```

Table 16-8 Parameter description

Parameter	Mandatory	Description
expression	Yes	Currently, only the **date_trunc** and **time_slice** functions are supported. If you use **time_slice**, you do not need to configure the **boundary** parameter because this parameter can only be set to **floor** by default.
time_unit	Yes	Partition granularity. Currently, the value can only be **hour**, **day**, **month**, or **year**. If the partition granularity is **hour**, the partition columns can only be of the DATETIME type.

Parameter	Mandatory	Description
partition_column	Yes	Partition column. ● Only the date type (DATE or DATETIME) is supported. If **date_trunc** is used, the partition column can be of the DATE or DATETIME type. If **time_slice** is used, the partition column can only be of the DATETIME type. The value of the partition column can be **NULL**. ● If the partition column is of the DATE type, the value range is from **0000-01-01** to **9999-12-31**. If the partition column is of the DATETIME type, the value range is from **0000-01-01 01:01:01** to **9999-12-31 23:59:59**. ● Currently, only one partition column can be specified.

Example: If you often query data by day, you can use the partition expression **date_trunc ()**, set the partition column to **event_day**, and set the partition granularity to **day** during table creation. In this way, data is automatically partitioned based on dates when being imported. Data of the same day is stored in the same partition. Partition pruning can significantly improve queries.

```
CREATE TABLE site_access1 (
    event_day DATETIME NOT NULL,
    site_id INT DEFAULT '10',
    city_code VARCHAR(100),
    user_name VARCHAR(32) DEFAULT '',
    pv BIGINT DEFAULT '0'
)
DUPLICATE KEY(event_day, site_id, city_code, user_name)
PARTITION BY date_trunc('day', event_day)
DISTRIBUTED BY HASH(event_day, site_id);
```

Partitioning based on the column expression: If you often query and manage data based on enumerated values, you only need to specify the column representing the type as the partition column. An HTAP instance automatically divides and creates partitions based on the partition column value of the imported data.

Syntax:

```
PARTITION BY expression
...
```

```
[ PROPERTIES( 'partition_live_number' = 'xxx' ) ]

expression ::=
    ( <partition_columns> )

partition_columns ::=
    <column>, [ <column> [,...] ]
```

Table 16-9 Parameter description

Parameter	Mandatory	Description
partition_columns	Yes	Partition columns. ● The value can be a Character (BINARY is not supported), Date, Integer, or Boolean value. The value cannot be **NULL**. ● After the import, a partition automatically created can contain only one value of each partition column. If multiple values of each partition column need to be contained, use list partitioning.

Example: If you often query the equipment room billing details by date range and city, you can use a partition expression to specify the date and city as the partition columns when creating a table. In this way, data of the same date and city is grouped into the same partition, and partition pruning can be used to significantly accelerate queries.

```
CREATE TABLE t_recharge_detail1 (
    id bigint,
    user_id bigint,
    recharge_money decimal(32,2),
    city varchar(20) not null,
    dt varchar(20) not null
)
DUPLICATE KEY(id)
PARTITION BY (dt,city)
DISTRIBUTED BY HASH(`id`);
```

List partitioning

Data is partitioned based on a list of enumerated values that you explicitly define. You need to explicitly list the enumerated values contained in each list partition, and the values do not need to be consecutive.

List partitioning is suitable for storing columns where there are a small number of enumerated values and querying and managing data based on the enumerated values. For example, a column indicates a geographical location, status, or category. Each value of a column represents an independent category. Data is partitioned based on the enumerated values of columns

to improve query performance and data management. List partitioning is especially suitable for scenarios where a partition needs to contain multiple values of each partition column. For example, the **city** column in a table indicates the city that an individual is from, and you often query and manage data by state and city. You can use the **city** column as the partition column for list partitioning when creating a table, and specify that data of multiple cities in the same state is stored in the same partition **PARTITION pCalifornia VALUES IN ("Los Angeles","San Francisco","San Diego")**, this feature accelerates queries and data management.

📖 NOTE

Partitions must be created during table creation. Partitions cannot be automatically created during data import. If the table does not contain the partitions corresponding to the data, an error is reported.

Syntax:

```
PARTITION BY LIST (partition_columns)(
    PARTITION <partition_name> VALUES IN (value_list)
    [, ...]
)

partition_columns::=
    <column> [,<column> [, ...] ]

value_list ::=
    value_item [, value_item [, ...] ]

value_item ::=
    { <value> | ( <value> [, <value>, [, ...] ] ) }
```

Table 16-10 Parameter description

Parameter	Mandatory	Description
partition_columns	Yes	Partition columns. The value can be a Character (except BINARY), Date (DATE and DATETIME), Integer, or Boolean value. The value cannot be **NULL**.
partition_name	Yes	Partition name. You are advised to set proper partition names to distinguish data categories in different partitions.
value_list	Yes	List of enumerated values of partition columns in a partition.

Example 1: If you often query the equipment room billing details by state or city, you can specify the **city** column as the partition

column and specify that the cities in each partition belong to the same state. In this way, you can quickly query data of a specific state or city and manage data by state or city.

```
CREATE TABLE t_recharge_detail2 (
    id bigint,
    user_id bigint,
    recharge_money decimal(32,2),
    city varchar(20) not null,
    dt varchar(20) not null
)
DUPLICATE KEY(id)
PARTITION BY LIST (city) (
PARTITION pCalifornia VALUES IN ("Los Angeles","San Francisco","San Diego"), --: These cities belong to the same state.
    PARTITION pTexas VALUES IN ("Houston","Dallas","Austin")
)
DISTRIBUTED BY HASH(`id`);
```

Example 2: If you often query the equipment room billing details by date range and state or city, you can specify the date and city as the partition columns when creating a table. In this way, data of a specific date and a specific state or city is grouped into the same partition, to accelerate queries and data management.

```
CREATE TABLE t_recharge_detail4 (
    id bigint,
    user_id bigint,
    recharge_money decimal(32,2),
    city varchar(20) not null,
    dt varchar(20) not null
) ENGINE=OLAP
DUPLICATE KEY(id)
PARTITION BY LIST (dt,city) (
  PARTITION p202204_California VALUES IN (
      ("2022-04-01", "Los Angeles"),
      ("2022-04-01", "San Francisco"),
      ("2022-04-02", "Los Angeles"),
      ("2022-04-02", "San Francisco")
   ),
  PARTITION p202204_Texas VALUES IN (
      ("2022-04-01", "Houston"),
      ("2022-04-01", "Dallas"),
      ("2022-04-02", "Houston"),
      ("2022-04-02", "Dallas")
   )
)
DISTRIBUTED BY HASH(`id`);
```

2) After entering the statement for performing column operations on the table, click **Verify** on the right of the area.

- If you select **Disabled**, go to **6**.
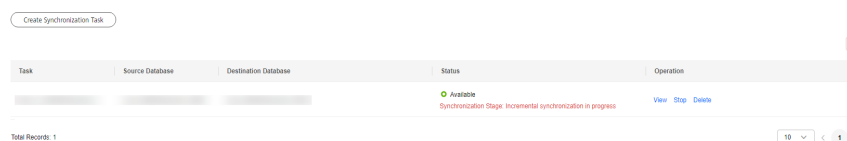
6. After the settings are complete, click **Create Synchronization Task**.

7. Confirm the settings and click **Sync Now**.

📖 **NOTE**

If you click **Previous** on the page or click [<] in the upper left corner of the page to return to the data synchronization page, a synchronization task will be generated. The status of the task is **Synchronization Stage: Waiting for synchronization**. To start the task, click **Synchronize** in the **Operation** column.

8. Click **Back to Synchronization List** to return to the data synchronization page. View details about the task name, source database, destination database, status, and operations.

Figure 16-7 Viewing task status

📖 **NOTE**

If the status of a task is **Synchronization Stage: Incremental synchronization in progress**, the data synchronization is complete.

Step 3: Connect to the HTAP Instance for OLAP Queries

For details about how to connect to a standard HTAP instance and perform OLAP queries, see **Connecting to a Standard HTAP Instance Through JDBC**.

16.3 Connecting to a Standard HTAP Instance

16.3.1 Connecting to a Standard HTAP Instance Through DAS

Data Admin Service (DAS) enables you to manage DB instances on a web-based console, simplifying database management and improving working efficiency.

By default, you have remote login permissions. It is recommended that you use DAS to connect to HTAP instances because this connection method is more secure and convenient than other methods.

Procedure

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

Step 2 Click [◎] in the upper left corner and select a region and project.

Step 3 Click [≡] in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 In the instance list, locate an HTAP instance and click **Log In** in the **Operation** column.

Figure 16-8 Logging in to a standard HTAP instance
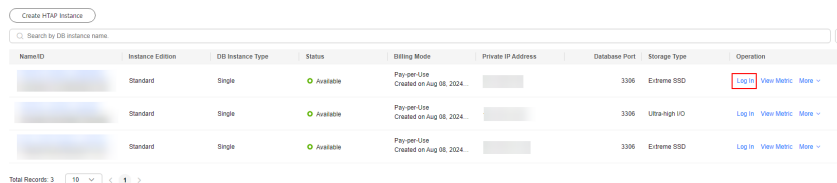
Step 7 Select the node to be logged in to, enter the database username and password, and click **Test Connection**.

Step 8 After the connection test is successful, click **Log In** to access your database.

----End

16.3.2 Connecting to a Standard HTAP Instance Through JDBC

You can connect to a standard HTAP instance through JDBC.

Precautions

Currently, HTAP instances only support the UTF-8 character set.

Prerequisites

- You are familiar with:
  - Computer basics
  - Java
  - JDBC knowledge
- You have downloaded the official **JDBC driver** for MySQL or MirrorDB.
- You have created a standard HTAP instance.
- The following dependency has been added to the **pom.xml** file.
  ```
  <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.47</version>
  </dependency>
  ```
- You can use the following command to connect to an HTAP instance through JDBC:

  jdbc:mysql://<instance_ip>:<instance_port>/<database_name>

Parameter	Description
<instance_ip>	IP address of the FE node in the HTAP instance. If a proxy is installed, use the IP address of the proxy.
<instance_port>	HTAP instance port. The default value is 3306.

Parameter	Description
<database_name>	Database name used for connecting to the instance.

Sample Code

Code example (Java code for connecting to an HTAP database):

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
public class JDBCTest {
    static final String IP = "*.*.*.*"; //IP address of the instance
    static final String USER = "***"; //Username
    static final String PASS = "***"; //Password
    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        String url = "jdbc:clickhouse://" + IP + ":8123";
        try {

            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(url, USER, PASS);
            stmt = conn.createStatement();
            String sql = "show databases;";
            ResultSet rs = stmt.executeQuery(sql);
            int columns = rs.getMetaData().getColumnCount();
            for (int i = 1; i <= columns; i++) {
                System.out.print(rs.getMetaData().getColumnName(i));
                System.out.print("\t");
            }
            while (rs.next()) {
                System.out.println();
                for (int i = 1; i <= columns; i++) {
                    System.out.print(rs.getObject(i));
                    System.out.print("\t");
                }
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // release resource ....
        }
    }
}
```

16.4 Standard HTAP Instance Management

16.4.1 Rebooting a Standard HTAP Instance

Scenarios

You may need to reboot an HTAP instance for maintenance reasons.

Constraints

- You can reboot an HTAP instance only when it is available or abnormal. When some operations such as creating a task, changing specifications, scaling up storage, and upgrading a minor version, are being performed on an HTAP instance, the instance cannot be rebooted.
- It takes about 1 to 2 minutes to reboot an HTAP instance. During the reboot, the instance is unavailable. Rebooting an HTAP instance will clear its cache. To prevent traffic congestion during peak hours, you are advised to reboot it during off-peak hours.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click **Reboot** in the **Operation** column.

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

----End

16.4.2 Rebooting a Node of a Standard HTAP Instance

Scenarios

You may need to reboot a node of an HTAP instance for maintenance reasons.

Constraints

- You can reboot a node only when it is available or abnormal. When some operations such as creating a task, changing specifications, scaling up storage, and upgrading a minor version, are being performed on a node, the node cannot be rebooted.
- It takes about 1 to 2 minutes to reboot a node of an HTAP instance. During the reboot, the instance is unavailable. Rebooting a node will clear its cache. To prevent traffic congestion during peak hours, you are advised to reboot the node during off-peak hours.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**. Locate an HTAP instance and click its name to access the **Basic Information** page.

Step 6 Locate an HTAP node and click **Reboot** in the **Operation** column.

Step 7 In the displayed dialog box, click **OK** to reboot the node. It takes about 1 to 2 minutes.

----End

16.4.3 Deleting a Standard HTAP Instance

Scenarios

You can delete any unused HTAP instances to release resources.

Precautions

- HTAP instances cannot be deleted when operations are being performed on them.
- Deleted HTAP instances cannot be recovered. Exercise caution when performing this operation.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click **Delete** in the **Operation** column.

Step 7 In the displayed dialog box, enter **DELETE** and click **OK**.

Figure 16-9 Deleting an HTAP instance
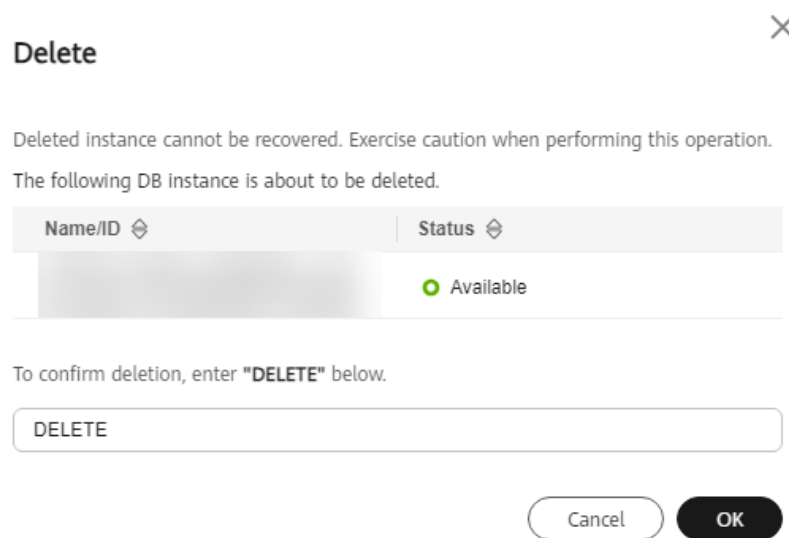
----End

16.5 Standard HTAP Account Management

Standard HTAP instances use the following methods to manage accounts:

- After GaussDB(for MySQL) data is synchronized to a standard HTAP instance, accounts cannot be synchronized. You need to manually create database accounts on the HTAP instance.

- You can create databases, tables, and accounts for your HTAP instances as needed.

This section describes how to create an account, reset the password, modify account permissions, and delete an account on the GaussDB(for MySQL) console.

System Accounts

To provide O&M services, the system automatically creates system accounts when you create HTAP instances, but these system accounts are not available to you.

- **rdsAdmin**: a management account with superuser permissions, which is used to query and modify instance information, rectify faults, migrate data, and restore data.

- **rdsMetric**: an account used for metric monitoring. This account is used by watchdog to collect database status data.

> **NOTICE**
>
> Deleting, renaming, and changing passwords or permissions for these accounts will cause the instance to run abnormally. Exercise caution when performing these operations.

Creating a Database Account

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Click the name of an HTAP instance to access the **Basic Information** page.

Step 7 In the navigation pane, choose **Accounts**. On the displayed page, click **Create Account**.

Step 8 In the displayed dialog box, set the required parameters.

Figure 16-10 Creating a database account
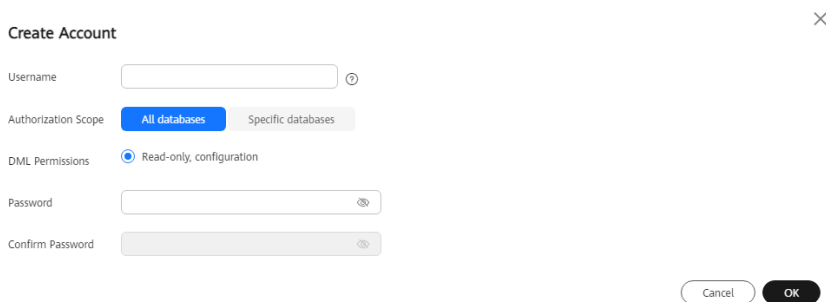
Table 16-11 Parameter description

Parameter	Description
Username	Contains 2 to 32 characters. It must start with a lowercase letter and end with a lowercase letter or digit. Only lowercase letters, digits, and underscores (_) are allowed.
Authorization Scope	• **All databases** • **Specific databases** **Database Not Authorized**: When creating an account, do not select any database in this area. The created account cannot perform operations on any database. To learn how to grant the required permissions for a particular database, see **Modifying Account Permissions**. **Database Authorized**: The databases selected in the **Database Not Authorized** area are displayed.
DML Permissions	The permissions include read-only, read/write, read and configuration, and read/write and configuration.

Parameter	Description
Password	● Contains 8 to 32 characters. ● Contains at least three of the following types of characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,). ● Cannot be the username or the username backwards.
Confirm Password	Must be the same as the new password.

Step 9 Click **OK**.

Step 10 In the account list, view the account information, including the username, authorized databases, and DML permissions.

> 📖 **NOTE**
>
> You can **reset account passwords**, **change account permissions**, or **delete accounts**.

----End

Resetting a Password

Step 1 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 2 In the navigation pane, choose **HTAP Analysis**.

Step 3 Click the name of an HTAP instance to access the **Basic Information** page.

Step 4 In the navigation pane, choose **Accounts**. On the displayed page, locate an account and click **Reset Password** in the **Operation** column.

Step 5 In the displayed dialog box, enter a new password, confirm the password, and click **OK**.

----End

Modifying Account Permissions

> 📖 **NOTE**
>
> If you delete a database somewhere other than on the HTAP console, permissions granted specifically for the database are not automatically deleted. They must be deleted manually.

Step 1 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 2 In the navigation pane, choose **HTAP Analysis**.

Step 3 Click the name of an HTAP instance to access the **Basic Information** page.

Step 4 In the navigation pane, choose **Accounts**. On the displayed page, locate an account and click **Change Permission** in the **Operation** column.

Step 5 In the displayed dialog box, modify permissions as required and click **OK**.

----End

Deleting an Account

Step 1 On the **Instances** page, locate a GaussDB(for MySQL) instance and click its name to access the **Basic Information** page.

Step 2 In the navigation pane, choose **HTAP Analysis**.

Step 3 Click the name of an HTAP instance to access the **Basic Information** page.

Step 4 In the navigation pane, choose **Accounts**. On the displayed page, locate an account and click **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, confirm the information and click **OK**.

----End

16.6 Syntax and Data Type Mappings between HTAP and GaussDB(for MySQL) Instances

When data of GaussDB(for MySQL) instances is synchronized to HTAP instances, the data types will be converted. For details, see **Table 16-12**.

Table 16-12 Data type conversion

Data Type	GaussDB(for MySQL) Data Type	HTAP Data Type
NUMERTIC	TINYINT	TINYINT
	TINYINT UNSIGNED	SMALLINT
	SMALLINT	SMALLINT
	SMALLINT UNSIGNED	INT
	MEDIUMINT	INT
	INTEGER/INT	INT
	INTEGER/INT UNSIGNED	BIGINT
	BIGINT	BIGINT
	BIGINT UNSIGNED	LARGEINT
	DECIMAL/NEMERIC	DECIMAL
	FLOAT	FLOAT
	DOUBLE/REAL	DOUBLE
	BIT	STRING
DATE TIME	DATE	DATE
	DATETIME	DATETIME

Data Type	GaussDB(for MySQL) Data Type	HTAP Data Type
	TIMESTAMP	DATETIME
	TIME	STRING
	YEAR	INT
STRING	CHAR	CHAR/VARCHAR
	VARCHAR	VARCHAR
	BINARY	VARBINARY
	VARBINARY	VARBINARY
	BLOB	VARBINARY
	TEXT	STRING
	ENUM	STRING
	SET	STRING
SPATIAL	GEOMETRY	STRING
	POINT	STRING
	LINESTRING	STRING
	POLYGON	STRING
	MULTIPOINT	STRING
	MULTILINESTRING	STRING
	MULTIPOLYGON	STRING
	GEOMETRYCOLLECTION	STRING
JSON	JSON	JSON

17 RegionlessDB Clusters

17.1 RegionlessDB Cluster Overview

A RegionlessDB cluster consists of multiple GaussDB(for MySQL) instances in different regions around the world. Currently, a RegionlessDB cluster consists of one primary instance (in the primary region) and up to five standby instances (in standby regions). Data is synchronized between primary and standby instances, providing nearby access and regional DR capabilities.

Figure 17-1 RegionlessDB cluster principle
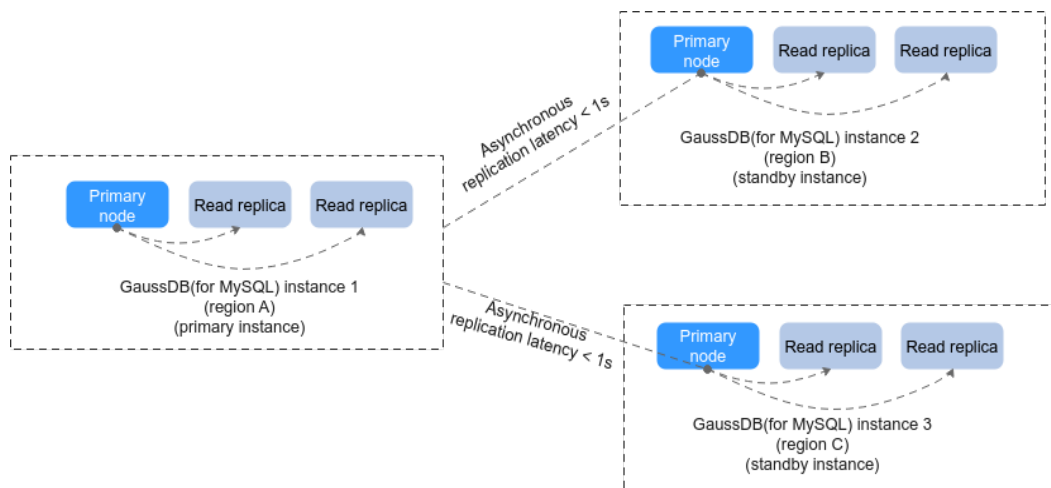
Scenarios

- Remote multi-active deployment

  Data is synchronized among instances in a RegionlessDB cluster. For lower network latency and quicker resource access, you can select the instance nearest to your workloads.

- Remote disaster recovery

  If there is a region-level fault on the primary instance, workloads can be switched to a standby instance for remote DR.

Architecture

Figure 17-2 Architecture
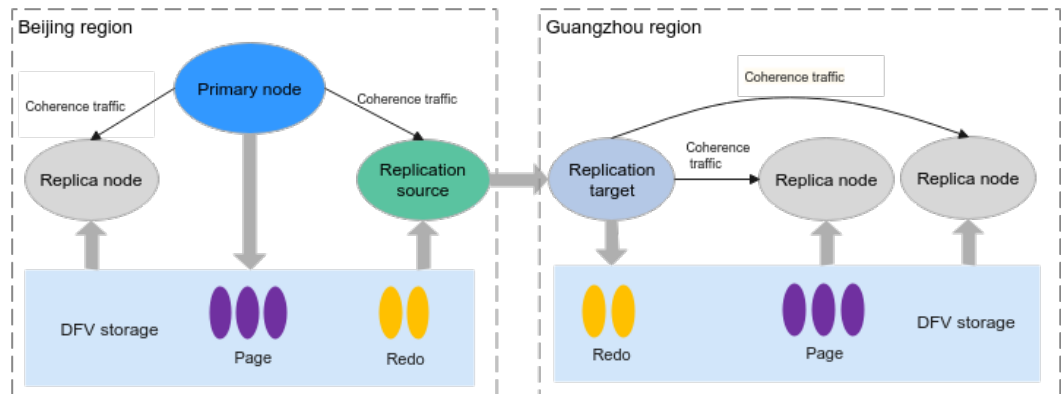
- Cross-region deployment is supported. Redo logs generated in the primary instance are synchronized to a standby instance and written to DFV storage. Pages required for database access are replayed. For details, see **Figure 17-2**. (Data is synchronized based on the replication node **Source** of the primary instance and the replication node **Target** of the standby instance.)

- In the primary instance, the read replica obtains required redo logs and pages from DFV storage through the primary node. In the standby instance, the read replica obtains required redo logs and pages from DFV storage through the replication node **Target**.

Advantages

- Global deployment and nearby data access

  Instances in a RegionlessDB cluster are from different regions around the world. Data generated by the primary instance can be directly read from the nearest standby instance.

- Low latency of cross-region replication

  Redo logs are directly and uninterruptedly read from the DFV storage for asynchronous replication. The replication latency is less than 1 second thanks to high-throughput parallel data synchronization.

- No downtime for the primary node during data synchronization

  The replication node of the primary instance reads data from different nodes in the DFV storage in parallel for synchronization. This means that the primary node does not need to directly synchronize data to the standby instances. Instead, it only needs to update the location information of redo logs in the storage to the replication node of the primary instance. In this way, workloads on the primary node are not affected.

- Too many read replicas

  There are up to five standby instances in a cluster, and each standby instance supports up to 15 read replicas.

  ☐ **NOTE**

  When you are creating a DB instance, a maximum of 10 read replicas can be created at a time.

- Region-level disaster recovery

  If there is a region-level fault on the primary instance, workloads can be quickly switched to a standby instance for remote DR, achieving an RPO in minutes and an RTO in seconds.

  📖 NOTE

  - If you need to use quick DR, contact customer service.
  - Recovery Point Objective (RPO): the maximum data loss amount tolerated by the system.
  - Recovery Time Objective (RTO): the maximum service interruption duration tolerated by the system. It refers to the requirement for the recovery duration of an information system failure or service function failure caused by a disaster.

Constraints

- Before using this feature, you need to obtain the data security compliance requirements of the local region and evaluate the compliance with related laws and regulations.

- RegionlessDB is in the open beta test (OBT) phase. To use this function, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

- To enable communication between regions, you need to create a Virtual Private Network (VPN) in advance. For details about how to create a VPN, see **Configuring Enterprise Edition S2C VPN to Connect an On-premises Data Center to a VPC**.

- Only pay-per-use instances can be created.

- The kernel version must be 2.0.46.231000 or later, and the primary instance must be a new instance.

- The instances in a RegionlessDB cluster cannot use 192.168.0.0/16 as their subnet CIDR block.

- The subnet CIDR blocks of the primary and standby instances in different regions must be different.

- When a standby instance is created, data needs to be synchronized from the primary instance. The time required depends on how much data there is.

- The primary instance in a RegionlessDB cluster cannot be restored to the original instance, and other instances cannot be restored to any instance in a RegionlessDB cluster.

- If you create proxy instances or HTAP instances for a GaussDB(for MySQL) instance, the GaussDB(for MySQL) instance cannot be used as an instance in a RegionlessDB cluster. Delete the proxy instances or HTAP instances first.

- The primary instance does not support the following operations:
  - Changing a database port
  - Changing a private IP address
  - Creating an HTAP instance
  - Creating a proxy instance

- The standby instance does not support the following operations:
  - Resetting a password

- – Creating and restoring a backup

- – Creating an account

- – Authorizing an account

- – Creating a proxy instance

- – Creating an HTAP instance

- – Promoting a read replica to the primary node

- – Changing a database port

- – Changing a private IP address

- – Modifying auto scaling policies

- Data across regions is synchronized through a network. The VPN bandwidth must be greater than the write bandwidth of the primary instance in a RegionlessDB cluster.

- In large-scale DDL scenarios, the replication latency may fluctuate for more than 1 second.

- RegionlessDB clusters do not support OpenAPIs.

- A RegionlessDB cluster consists of one primary instance (in the primary region) and up to five standby instances (in standby regions). The primary instance processes read and write requests and the standby instances process read-only requests. **Table 17-1** lists the maximum specifications supported by a RegionlessDB cluster.

Table 17-1 Specifications

Description	Primary Instance	Standby Instance
Max. Instances	1	5
Max. Read/Write Nodes per Instance	1	0
Max. Read-only Nodes per Instance	15	15

☐ **NOTE**

When you are creating a DB instance, a maximum of 10 read replicas can be created at a time.

17.2 Using a RegionlessDB Cluster for Remote Multi-Active DR

Scenarios

If your workloads are deployed in multiple regions, you can create a RegionlessDB cluster to access databases from the nearest region. As shown in **Figure 17-3**, a RegionlessDB cluster contain a primary instance and two standby instances. Read requests are sent to a standby instance in the nearest region, and write requests

are automatically forwarded from the nearest region to the primary instance. After data is written to the primary instance, the data is synchronized to all standby instances, reducing the cross-region network latency.

Figure 17-3 Remote multi-active principle
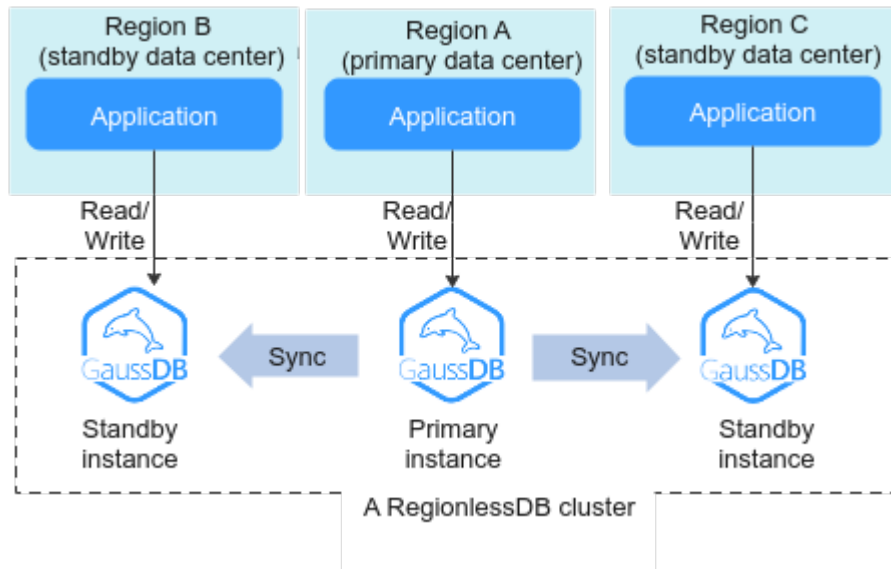
Constraints

For details, see **Constraints**.

Step 1: Create a RegionlessDB Cluster

1. **Log in to the management console.**

2. Click ⊙ in the upper left corner and select a region and project.

3. Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

4. On the **RegionlessDB** page, click **Create RegionlessDB** in the upper right corner.

   Figure 17-4 Creating a RegionlessDB cluster
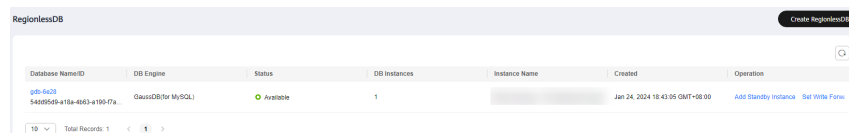
5. In the **Create RegionlessDB** dialog box, configure **RegionlessDB Name**, **Primary Instance Region**, and **Primary Instance**.

Figure 17-5 Configuring the RegionlessDB cluster information

Table 17-2 Parameter description

Parameter	Description
RegionlessDB Name	The name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
Primary Instance Region	Select a region where the primary instance is located.
Primary Instance	Select an existing DB instance as the primary instance of the RegionlessDB cluster.

6. Click **OK**.

7. After the primary instance is created, view and manage it.

   During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the primary instance is **Available**, you can use the instance.

Step 2: Add a Standby Instance

1. On the **RegionlessDB** page, locate the RegionlessDB cluster.

2. Click **Add Standby Instance** in the **Operation** column.

Figure 17-6 Adding a standby instance

3. On the displayed page, configure related parameters.

Table 17-3 Basic information

Parameter	Description
Region	Region where the standby instance is deployed. **NOTICE** Products in different regions cannot communicate with each other through a private network. After a DB instance is purchased, the region cannot be changed.
Creation Method	Create new
DB Instance Name	The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0
Kernel Version	Kernel version of the standby instance. The kernel version must be 2.0.46.231000 or later. For details about the updates in each minor kernel version, see **Kernel Version Release History**. **NOTE** To configure the kernel version, contact customer service.
DB Instance Type	Only **Primary/Standby** can be selected. There are 2 to 10 read replicas in a primary/standby instance in the RegionlessDB cluster.
Storage Type	Shared
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. ● **Single AZ**: The primary node and read replicas are deployed in the same AZ. ● **Multi-AZ**: The primary node and read replicas are deployed in different AZs to ensure high reliability.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.

Parameter	Description
Instance Specifications	For details about the specifications supported by GaussDB(for MySQL), see **Instance Specifications**. GaussDB(for MySQL) is a cloud-native database that uses the shared storage. To ensure service stability in high read/write pressure, the system controls the read/write peaks of DB instances based on instance specifications. For details about how to select specifications, see **Performance White Paper**.
CPU Architecture	The CPU architecture can be x86 or Kunpeng. Under a CPU architecture, you need to select vCPUs and memory of the instance.
Nodes	All nodes of the standby instance are read replicas. You can apply for a maximum of 10 read replicas at a time for a pay-per-use instance. After an instance is created, you can add read replicas as required. Up to 15 read replicas can be created for a standby instance in a cluster.
Storage	Storage will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.

VPC	• A dedicated virtual network in which your GaussDB(for MySQL) instance is located. It isolates networks for different workloads. You can select an existing VPC or create a VPC. For details about how to create a VPC, see **Creating a VPC**.
	If no VPC is available, GaussDB(for MySQL) allocates a VPC to you by default.
	NOTICE
	    • Ensure that the VPC selected for the standby instance is connected to the VPC selected for the primary instance through a VPN.
	    • After a GaussDB(for MySQL) instance is created, the VPC cannot be changed.
	• A subnet provides dedicated network resources that are logically isolated from other networks for network security. A private IP address is automatically assigned when you create a DB instance. You can also enter an idle private IP address in the subnet CIDR block.

Parameter	Description
Security Group	It can enhance security by controlling access to GaussDB(for MySQL) from other services. When you select a security group, you must ensure that it allows the client to access instances. If no security group is available or has been created, GaussDB(for MySQL) allocates a security group to you by default. **NOTE** ● To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. ● Configure private network security group rules to ensure that the primary and standby instances in a cluster can communicate with each other.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. You can modify the instance parameters as required after the instance is created. **NOTICE** If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not applied. Instead, the default values are used. innodb_buffer_pool_size innodb_log_buffer_size max_connections innodb_buffer_pool_instances innodb_page_cleaners innodb_parallel_read_threads innodb_read_io_threads innodb_write_io_threads threadpool_size After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters in a Parameter Template**.
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Managing Tags**.

> **NOTE**
>
> The instance password and table name case sensitivity are the same as those of the primary instance. You do not need to set them separately.

4. Click **Next**.

5. Confirm the information and click **Submit**.

6. Go to the **Instances** page to view and manage the instance.

   During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the instance is **Available**, you can use the instance.

   > **NOTE**
   >
   > If there is a large amount of data in the primary instance, it may take a long time to complete a full backup during standby instance creation.

Step 3: Enable Write Forwarding

In normal cases, after a RegionlessDB cluster is created, the primary instance receives and processes read and write requests, and standby instances receive only read requests. After write forwarding is enabled, standby instances can receive write requests and then forward them to the primary instance for processing. After data is written to the primary instance, the data is synchronized to all standby instances. Write forwarding simplifies the data write process. You can directly connect a database service through a standby instance's IP address to perform read and write operations. In addition, consistency is ensured and the nearby read is not affected.

NOTICE

- Write forwarding is only available when the transaction isolation level of the standby instances is RR.

- In the current version, WARNING and RECORD information cannot be displayed when a standby instance forwards write requests.

- In the current version, SQL requests that are being executed cannot be interrupted when a standby instance forwards write requests.

- When write forwarding is enabled, user **_@gdb_WriteForward@_** is created. Do not modify or delete the user, or write forwarding cannot run properly.

- The following commands are supported for write forwarding:
    - SQLCOM_UPDATE
    - SQLCOM_INSERT
    - SQLCOM_DELETE
    - SQLCOM_INSERT_SELECT
    - SQLCOM_REPLACE
    - SQLCOM_REPLACE_SELECT
    - SQLCOM_DELETE_MULTI
    - SQLCOM_UPDATE_MULTI
    - SQLCOM_ROLLBACK

    If an unsupported command is executed, the following error information is displayed.
    ```
    ERROR xxx (yyy): This version of MySQL doesn't yet support 'operation with write forwarding'.
    ```
    operation indicates the operation type that is not supported.

- The following scenarios are not supported:
    - There are SELECT FOR UPDATE statements.
    - There are EXPLAIN write forwarding statements.
    - The statements for write forwarding contain SET VARIABLE.
    - SAVEPOINT is not supported when write forwarding is enabled.
    - Write forwarding is not supported in XA transactions.
    - Currently, START TRANSACTION READ WRITE is not supported. You can directly use START TRANSACTION to test write forwarding.
    - Write forwarding is not supported in stored procedures.
    - When write forwarding is enabled, temporary tables cannot be created. To create temporary tables, disable write forwarding temporarily.

- For commands that can be implicitly committed, if write forwarding is not supported, the transactions corresponding to the current node and primary node are automatically committed.

- For the global consistency level, before accessing data for the first time, each transaction needs to use a connection in the session pool to obtain a data point (LSN) from the primary node. If no sessions are available, the command for reading data may fail.

- If there is a connection error when a user uses a session for write forwarding and the user is in a multi-statement transaction, the server proactively closes
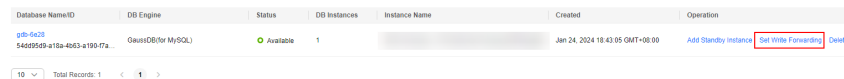
the connections to the client and the primary node, ensuring that the client can detect the error.

- The versions of the primary and standby instances must be the latest.

- Write operations are finally forwarded to and processed by the primary node. If a temporary table with the same name exists in the given database of the primary and read replicas, the data on the primary node is used.

- If there is a primary/standby switchover or failover for a standby instance in a RegionlessDB cluster, the write forwarding parameters (**rds_open_write_forwarding** and **rds_write_forward_read_consistency**) are restored to the default values.

Step 1 On the **RegionlessDB** page, locate the RegionlessDB cluster.

Step 2 Click **Set Write Forwarding** in the **Operation** column to create a write forwarding account.

Figure 17-7 Creating a write forwarding account

☐ NOTE

> The system automatically creates an internal account (**_@gdb_WriteForward@_**) so that write requests can be forwarded to the primary instance for processing. You cannot modify or delete the internal account, or write forwarding will be affected.

Step 3 In the **Set Write Forwarding** dialog box, confirm the information and click **OK**.

Figure 17-8 Setting write forwarding
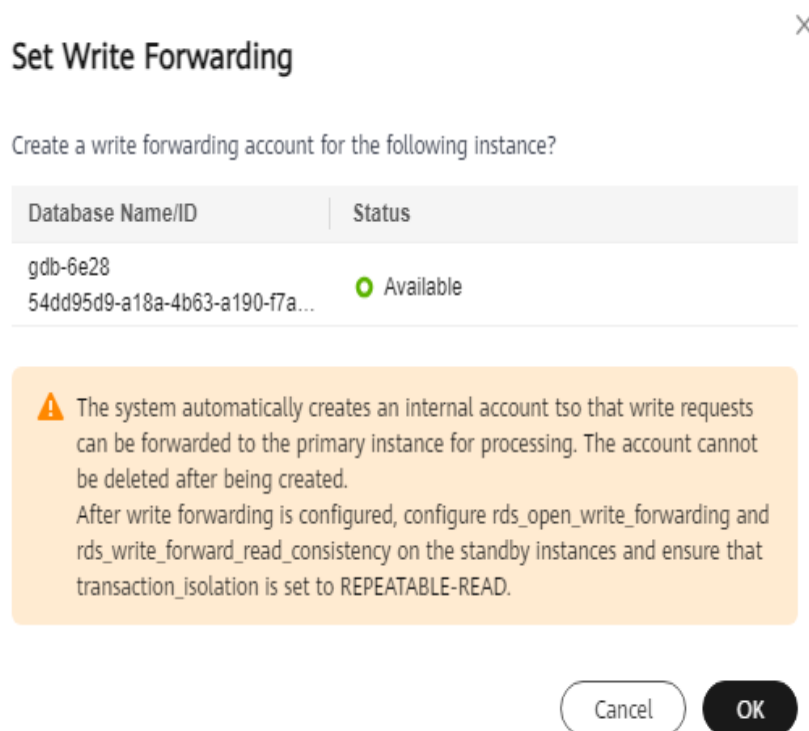
Step 4 On the **Instances** page, click the name of the standby instance in the RegionlessDB cluster.

Step 5 In the navigation pane, choose **Parameters**.

Step 6 Search for **rds_open_write_forwarding** in the upper right corner of the **Parameters** page and change its value to **ON**.

Step 7 Click **Save** in the upper left corner to enable write forwarding.

Step 8 Search for **rds_write_forward_read_consistency** in the upper right corner of the **Parameters** page and change the read consistency level of write forwarding.

You can modify the parameters to set the read consistency range. For details, see **Table 17-4**.

Table 17-4 Parameter description

Parameter	Description
NONE	Write forwarding is disabled.
EVENTUAL	Results of write operations are not visible until the write operations are performed on the primary instance. The query does not wait for data synchronization between primary and standby instances to complete, so data that is not updated may be read.

Parameter	Description
SESSION	All queries executed by a standby instance with write forwarding enabled see the results of all data writes performed in this session. The queries wait for the results of forwarded write operations to be replicated.
GLOBAL	A session can view all committed changes of all sessions and instances in a RegionlessDB cluster. The query may wait for a certain period, which is related to the replication latency.

📖 **NOTE**

- If read consistency is required, you are advised to set the consistency level to SESSION. The consistency level GLOBAL will cause a large extra cost for all read requests. For example, if any client is used to connect to GaussDB(for MySQL) and the GLOBAL level is used, the time for accessing the MySQL command line is prolonged.

- The read consistency level in write forwarding cannot be changed to SESSION in a transaction.

- Before enabling write forwarding, ensure that the transaction isolation levels of standby instances are RR.

- When write forwarding is enabled, the transaction isolation level of the current session cannot be changed.

- The read consistency level cannot be changed in a transaction.

Step 9 Click **Save** in the upper left corner.

----End

Step 4: Connect to the RegionlessDB Cluster for Service Management

After a RegionlessDB cluster is created, no unified connection address is provided. The primary and standby instances in the RegionlessDB cluster provide independent connection addresses. You can use the nearest primary or standby instance based on the service access region to connect to the RegionlessDB cluster. The RegionlessDB cluster automatically forwards write requests to the primary instance for processing and read requests to the instance of the nearest region for processing.

Example:

1. Connect to the primary instance and write data to the database.
   ```
   mysql> CREATE DATABASE mydatabase;
   mysql> CREATE TABLE orders (order_id INT PRIMARY KEY, customer_name VARCHAR(255),
   order_date DATE);
   mysql> INSERT INTO orders (order_id, customer_name, order_date) VALUES (1, 'UserA', '2023-12-18'),
   (2, 'UserB', '2023-12-17'), (3, 'UserC', '2023-12-16');
   ```

2. Use the nearest standby instance to access the database and query the data written in **1**.
   ```
   mysql> select * from mydatabase.orders;
   +----------+---------------+------------+
   | order_id | customer_name | order_date |
   +----------+---------------+------------+
   |        1 | UserA         | 2023-12-18 |
   ```

```
|    2 | UserB        | 2023-12-17 |
|    3 | UserC        | 2023-12-16 |
+----------+---------------+------------+
```

3. Connect to database through the primary instance and run the following SQL statements to create the **mydatabase** database and **orders** table.

```
mysql> CREATE DATABASE mydatabase;
Query OK, 1 row affected (0.00 sec)
mysql> USE mydatabase;
Database changed
mysql> CREATE TABLE orders (order_id INT PRIMARY KEY, customer_name VARCHAR(255),
order_date DATE);
Query OK, 0 rows affected (0.01 sec)
```

4. Connect to the database from a standby instance, run the following SQL statements to write three data records to the **orders** table, and query the written data.

```
mysql> INSERT INTO orders (order_id, customer_name, order_date) VALUES (1, 'UserA', '2023-12-18'),
(2, 'UserB', '2023-12-17'), (3, 'UserC', '2023-12-16');
Query OK, 3 rows affected (0.00 sec)
mysql> SELECT * FROM mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|    1 | UserA        | 2023-12-18 |
|    2 | UserB        | 2023-12-17 |
|    3 | UserC        | 2023-12-16 |
+----------+---------------+------------+
3 rows in set (0.01 sec)
```

5. Connect to the database through the primary instance and run the following SQL statements to query the data inserted by the standby instance in **4**.

```
mysql> SELECT * FROM mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|    1 | UserA        | 2023-12-18 |
|    2 | UserB        | 2023-12-17 |
|    3 | UserC        | 2023-12-16 |
+----------+---------------+------------+
3 rows in set (0.00 sec)
```

17.3 Using a RegionlessDB Cluster for Remote DR

Scenarios

If there is a region-level fault on the primary instance, workloads can be switched to a standby instance for remote DR.

As shown in **Figure 17-9**, a RegionlessDB cluster contains a primary instance deployed across two AZs and a standby instance deployed in a single AZ. If the primary AZ of the primary instance is faulty, workloads are preferentially switched to the standby AZ. If both the primary and standby AZs of the primary instance are faulty, workloads are switched to the standby instance.

Figure 17-9 Remote DR principle
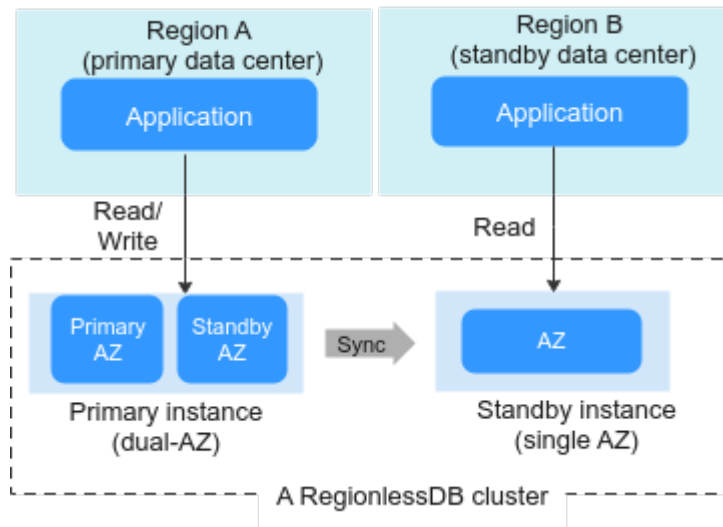
Constraints

For details, see **Constraints**.

Step 1: Create a RegionlessDB Cluster

1. **Log in to the management console.**

2. Click ⊙ in the upper left corner and select a region and project.

3. Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

4. On the **RegionlessDB** page, click **Create RegionlessDB** in the upper right corner.

    Figure 17-10 Creating a RegionlessDB cluster
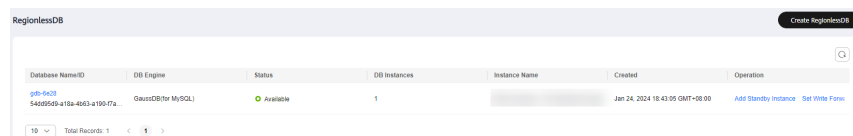
5. In the **Create RegionlessDB** dialog box, configure **RegionlessDB Name**, **Primary Instance Region**, and **Primary Instance**.

Figure 17-11 Configuring the RegionlessDB cluster information

Table 17-5 Parameter description

Parameter	Description
RegionlessDB Name	The name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
Primary Instance Region	Select a region where the primary instance is located.
Primary Instance	Select an existing DB instance as the primary instance of the RegionlessDB cluster.

6. Click **OK**.

7. After the primary instance is created, view and manage it.

   During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the primary instance is **Available**, you can use the instance.

Step 2: Add a Standby Instance

1. On the **RegionlessDB** page, locate the RegionlessDB cluster.

2. Click **Add Standby Instance** in the **Operation** column.

Figure 17-12 Adding a standby instance

3. On the displayed page, configure related parameters.

Table 17-6 Basic information

Parameter	Description
Region	Region where the standby instance is deployed. **NOTICE** Products in different regions cannot communicate with each other through a private network. After a DB instance is purchased, the region cannot be changed.
Creation Method	Create new
DB Instance Name	The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0
Kernel Version	Kernel version of the standby instance. The kernel version must be 2.0.46.231000 or later. For details about the updates in each minor kernel version, see **Kernel Version Release History**. **NOTE** To configure the kernel version, contact customer service.
DB Instance Type	Only **Primary/Standby** can be selected. There are 2 to 10 read replicas in a primary/standby instance in the RegionlessDB cluster.
Storage Type	Shared
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. ● **Single AZ**: The primary node and read replicas are deployed in the same AZ. ● **Multi-AZ**: The primary node and read replicas are deployed in different AZs to ensure high reliability.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.

Parameter	Description
Instance Specifications	For details about the specifications supported by GaussDB(for MySQL), see **Instance Specifications**. GaussDB(for MySQL) is a cloud-native database that uses the shared storage. To ensure service stability in high read/write pressure, the system controls the read/write peaks of DB instances based on instance specifications. For details about how to select specifications, see **Performance White Paper**.
CPU Architecture	The CPU architecture can be x86 or Kunpeng. Under a CPU architecture, you need to select vCPUs and memory of the instance.
Nodes	All nodes of the standby instance are read replicas. You can apply for a maximum of 10 read replicas at a time for a pay-per-use instance. After an instance is created, you can add read replicas as required. Up to 15 read replicas can be created for a standby instance in a cluster.
Storage	Storage will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
VPC	● A dedicated virtual network in which your GaussDB(for MySQL) instance is located. It isolates networks for different workloads. You can select an existing VPC or create a VPC. For details about how to create a VPC, see **Creating a VPC**. If no VPC is available, GaussDB(for MySQL) allocates a VPC to you by default. **NOTICE** 　● Ensure that the VPC selected for the standby instance is connected to the VPC selected for the primary instance through a VPN. 　● After a GaussDB(for MySQL) instance is created, the VPC cannot be changed. ● A subnet provides dedicated network resources that are logically isolated from other networks for network security. A private IP address is automatically assigned when you create a DB instance. You can also enter an idle private IP address in the subnet CIDR block.

Parameter	Description
Security Group	It can enhance security by controlling access to GaussDB(for MySQL) from other services. When you select a security group, you must ensure that it allows the client to access instances. If no security group is available or has been created, GaussDB(for MySQL) allocates a security group to you by default. **NOTE** ● To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. ● Configure private network security group rules to ensure that the primary and standby instances in a cluster can communicate with each other.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. You can modify the instance parameters as required after the instance is created. **NOTICE** If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not applied. Instead, the default values are used. innodb_buffer_pool_size innodb_log_buffer_size max_connections innodb_buffer_pool_instances innodb_page_cleaners innodb_parallel_read_threads innodb_read_io_threads innodb_write_io_threads threadpool_size After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters in a Parameter Template**.
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Managing Tags**.

☐ NOTE

The instance password and table name case sensitivity are the same as those of the primary instance. You do not need to set them separately.

4. Click **Next**.
5. Confirm the information and click **Submit**.
6. Go to the **Instances** page to view and manage the instance.

During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the instance is **Available**, you can use the instance.

☐ NOTE

If there is a large amount of data in the primary instance, it may take a long time to complete a full backup during standby instance creation.

Step 3: Connect to a Standby Instance for Workload Management

Select the nearest standby instance to access the database based on the workload access area.

For example, use a standby instance to access the database and query data.

```
mysql> select * from mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|        1 | UserA         | 2023-12-18 |
|        2 | UserB         | 2023-12-17 |
|        3 | UserC         | 2023-12-16 |
+----------+---------------+------------+
```

17.4 Performing a Primary/Standby Switchover or Failover in a RegionlessDB Cluster

A RegionlessDB cluster consists of multiple GaussDB(for MySQL) instances in different regions around the world. The cluster in each region inherits the original same-region HA feature. A RegionlessDB cluster provides cross-region primary/standby switchover and failover capabilities.

Primary/Standby Switchovers

You can promote any standby instance to primary. After the switchover, the original primary instance will be added back to the cluster as a standby instance.

Before a switchover, all data on the primary instance will be synchronized to other standby instances, ensuring no data loss.

Failovers

If the primary instance in a RegionlessDB cluster fails and cannot be restored, usually due to a regional outage, a failover is triggered to promote the standby

instance with the latest data from all available standby instances to the primary instance.

A failover may result in some data loss, depending on the replication latency between the primary and standby instances during the failover.

Generally, a failover can be complete within several minutes. However, after a failover is performed, the original primary instance needs to be rebuilt as a standby instance before being added back to the entire cluster. The rebuilding process may take dozens of minutes to several hours, depending on the data volume and network conditions between regions.

Other Operations and Checks

The primary and standby instances are independent GaussDB(for MySQL) instances. During a primary/standby switchover or failover, the configurations between the primary and standby instances are not exchanged. To prevent performance and compatibility issues caused by different configuration parameters, you are advised to check for any differences in configuration items between the primary and standby instances after a primary/standby switchover or failover.

- Check the read/write addresses of your application.

  During a primary/standby switchover or failover, the read/write addresses of instances are not exchanged. You need to check whether the read/write address of your application is as expected.

  During a failover, you can configure an application to use the read/write address of the new primary instance. After the faulty instance is rebuilt, reconfigure the read/write address of the application.

- Check the write forwarding configurations of standby instances. For details, see **Step 3: Enable Write Forwarding**.

- Check the configurations of a parameter template. For details, see **Modifying Parameters of a DB Instance**.

- Configure monitoring alarms. For details, see **Monitoring and Alarms**.

NOTE

- The current version does not support primary/standby switchovers.
- If a failover is required, contact customer service.

17.5 Removing a Standby Instance from a RegionlessDB Cluster

You can remove a standby instance from a RegionlessDB cluster.

Constraints

For details, see **Constraints**.

Procedure

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

Step 2 Click ⦿ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **RegionlessDB** page, locate the RegionlessDB cluster.

Step 5 Click the name of the cluster to view its details.

Step 6 In the instance list area, locate a standby instance and click **Remove** in the **Operation** column.

Figure 17-13 Accessing the instance list page

Name/ID	Region	Status	Node Type	Instance Specifications	Created	Operation
rds-taurus 0608f1ae6bdf44b949866c4d0b...			Primary Instance	gaussdb.mysql.large.x86.4	Jan 24, 2024 18:43:05 GMT+08:00	View Metric
gauss-6994 a6d3cfcf3d12484e9e2a5a8393...			Standby Instance	gaussdb.mysql.large.x86.normal.2	Feb 26, 2024 14:30:16 GMT+08:00	View Metric Remove

10 ▼ Total Records: 2 ‹ 1 ›

Step 7 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 8 In the displayed dialog box, enter **DELETE** in the text box and click **OK**.

Figure 17-14 Removing a standby instance from RegionlessDB

Remove from RegionlessDB ✕

Selected for removing from the RegionlessDB: **1**

After being removed, the instance cannot be added to the RegionlessDB as a standby instance.

Name/ID	Status
gauss-6994 a6d3cfcf3d12484e9e2a5a8393...	

To confirm deletion, enter **"DELETE"** below.

[DELETE]

Cancel   OK

To view the detailed progress and result of the task, go to the **Task Center** page.

📖 NOTE

- Only standby instances can be removed from a RegionlessDB cluster.
- After a standby instance is removed from a RegionlessDB cluster, data of the primary instance will not be synchronized to the standby instance.
- After a standby instance is removed from a RegionlessDB cluster, the standby instance will be permanently deleted. Exercise caution when performing this operation.

----End

17.6 Deleting a RegionlessDB Cluster

You can delete a RegionlessDB cluster.

Constraints

- Before deleting a RegionlessDB cluster, ensure that all standby instances have been removed from it. For details about how to remove a standby instance, see **Removing a Standby Instance from a RegionlessDB Cluster**.
- For more constraints, see **Constraints**.

Procedure

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **RegionlessDB**.

Step 5 Locate a cluster and click **Delete** in the **Operation** column.

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

For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 7 In the **Delete RegionlessDB** dialog box, select or deselect **Delete instances in RegionlessDB** as required, enter **DELETE** in the text box, and click **OK**.

Figure 17-15 Deleting a RegionlessDB cluster
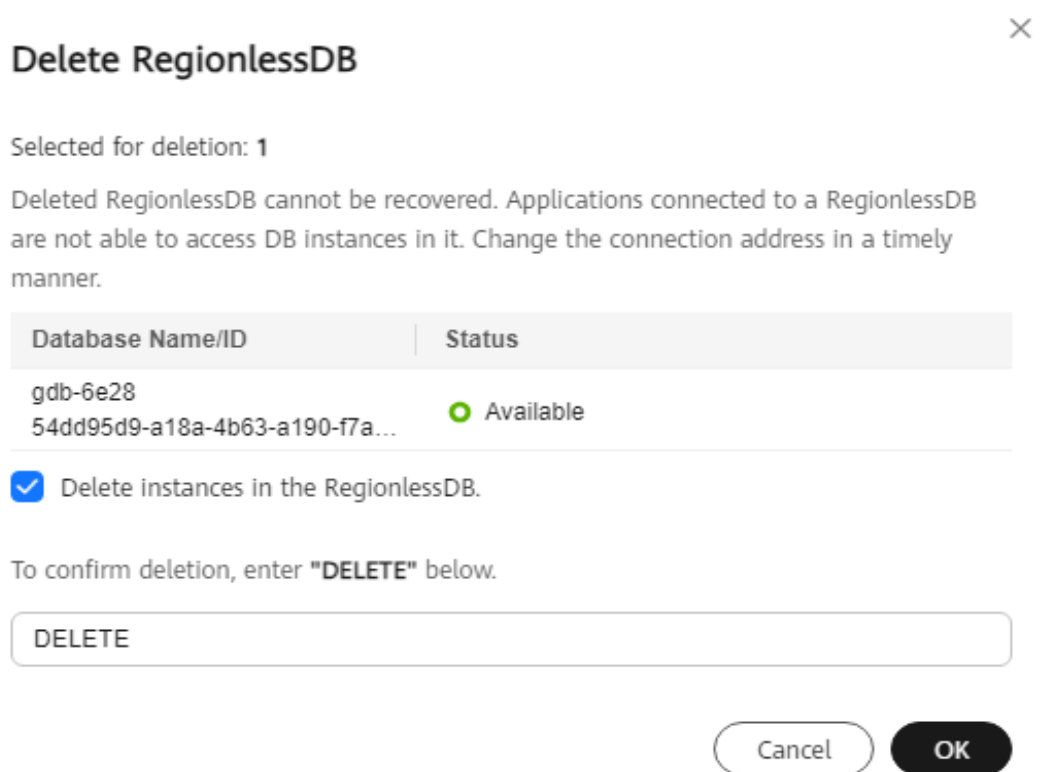
Step 8 Refresh the RegionlessDB cluster list later to confirm that the deletion was successful.

To view the detailed progress and result of the task, go to the **Task Center** page.

----End

17.7 Viewing the Replication Latency and Traffic of a RegionlessDB Cluster

After a RegionlessDB cluster is created, you can monitor the database status and performance based on related metrics.

Method 1: Viewing Metrics on the Console

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

Step 2 Click ⊙ in the upper left corner and select the region and project of the primary instance.

Step 3 Click ☰ and select **Cloud Eye** under **Management & Governance**.

Step 4 In the navigation pane, choose **Cloud Service Monitoring** > **GaussDB(for MySQL)**.

Step 5 Click ⌄ in the front of a RegionlessDB cluster. Locate a standby instance and click **View Metric** in the **Operation** column.

- Cloud Eye can monitor performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 24 hours or last 7 days.

Figure 17-16 Viewing RegionlessDB metrics
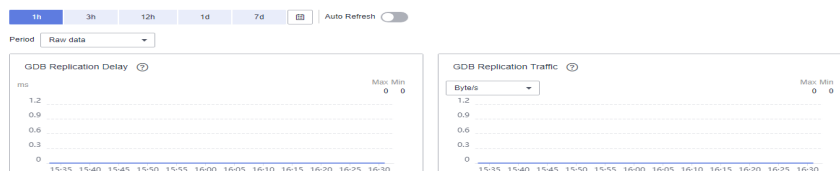
- For details about metrics supported by RegionlessDB clusters, see **Table 17-7**.

Table 17-7 RegionlessDB cluster metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gdb_replication_latency	GDB Replication Delay	Data replication latency of the measured object	≥0 ms	RegionlessDB standby instances	1 minute
gdb_replication_capacity	GDB Replication Traffic	Data replication traffic of the measured object	≥0 bytes /s	RegionlessDB standby instances	1 minute

📖 **NOTE**

For details about the metrics of primary and standby instances, see section 4.17.3

----End

Method 2: Viewing Metrics Using SQL Commands

Use a MySQL client tool to connect to the GaussDB(for MySQL) instance and run the following command to query the RegionlessDB status:

```
mysql> select * from information_schema.global_db_status;
```

Figure 17-17 Querying the RegionlessDB status
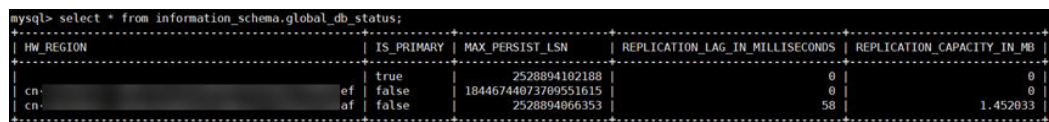
In the command output, each row indicates an instance in the RegionlessDB cluster (the first row indicates the primary instance and other rows indicate the standby instances). For details about the parameters contained in each row, see **Table 17-8**.

Table 17-8 Parameter description

Parameter	Description
HW_REGION	Region code of the standby instance. The first row in the table is the primary instance, and the region code of the primary instance is an empty string.
IS_PRIMARY	Whether the instance is the primary instance. **true**: it is the primary instance. **false**: It is the standby instance.
MAX_PERSIST_LSN	Maximum LSN of the current redo logs of the instance that have been persisted to the shared storage.
REPLICATION_LAG_IN_ MILLISECONDS	Latency from the time when data is written to the primary instance to the time when data can be read from the standby instance, in ms. The replication latency of the primary instance is 0.
REPLICATION_CAPACITY _IN_MB	Throughput of data replication from the primary instance to a standby instance, in MB/s. The replication throughput of the primary instance is 0.

18 Monitoring and Alarms

18.1 GaussDB(for MySQL) Metrics

Function

You can monitor the status of your instances using Cloud Eye. The namespaces, descriptions, and dimensions of metrics of instances can be reported to Cloud Eye.

The monitoring interval can be 1 minute, 1 second, or 5 seconds. The default monitoring interval is 1 minute. To enable Monitoring by Seconds, contact customer service to apply for the required permissions.

Namespace

SYS.GAUSSDB

DB Instance Metrics

Table 18-1 DB instance metrics

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100%	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_ mysql00 2_mem_ util	Memo ry Usage	Memory usage of the monitored object	0– 100%	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_ mysql00 4_bytes_i n	Netwo rk Input Throug hput	Incoming traffic in bytes per second	≥0 bytes/ s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_ mysql00 5_bytes_ out	Netwo rk Output Throug hput	Outgoing traffic in bytes per second	≥0 bytes/ s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_ mysql00 6_conn_c ount	Total Conne ctions	Total number of connections that attempt to connect to the GaussDB(for MySQL) server	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_ mysql00 7_conn_a ctive_cou nt	Curren t Active Conne ctions	Number of active connections	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_ mysql00 8_qps	QPS	Query times of SQL statements (including DDL, DML, SHOW, SET statements and storage procedures) per second	≥0 times/ s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_ mysql00 9_tps	TPS	Execution times of submitted and rollback transactions per second	≥0 times/ s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql010_innodb_buf_usage	Buffer Pool Usage	Ratio of used pages to total pages in the InnoDB buffer	0-1	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0-1	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to all data in the InnoDB buffer	0–100%	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥0 bytes/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql014_innodb_writes	InnoDB Write Throughput	Bytes written to pages by InnoDB per second. GaussDB(for MySQL) only writes data to temporary tables.	≥0 bytes/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql017_innodb_log_write_req_count	InnoDB Log Write Request Frequency	Number of InnoDB log write requests per second	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql019_innodb_log_writes	InnoDB Log Writes	Number of physical writes to the InnoDB redo log file	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql020_temp_tbl_count	Temporary Tables	Number of temporary tables automatically created on disks when GaussDB(for MySQL) statements are executed	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥0 counts /s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥0 counts /s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥0 counts /s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥0 counts /s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥0 counts /s	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0 GB-128 TB	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql060_rx_errors	Error Rate of Received Packets	Ratio of the number of error packets to the total number of received packets during the monitoring period	0–100%	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql061_rx_dropped	Loss Rate of Received Packets	Ratio of the number of lost packets to the total number of received packets during the monitoring period	0–100%	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql062_tx_errors	Error Rate of Sent Packets	Ratio of the number of error packets to the total number of sent packets during the monitoring period	0–100%	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql063_tx_dropped	Loss Rate of Sent Packets	Ratio of the number of lost packets to the total number of sent packets during the monitoring period	0–100%	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql072_conn_usage	Connection Usage	Percent of used GaussDB(for MySQL) connections to the total number of connections	0–100%	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql074_slow_queries	Slow Query Logs	Number of GaussDB(for MySQL) slow query logs generated per minute	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql077_replication_delay	Replication Delay	Delay between the primary node and read replicas **NOTE** This metric is used only for read replicas.	≥ 0s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql104_dfv_write_delay	Storage Write Delay	Average delay of writing data to the storage layer in a specified period	≥0 ms	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql105_dfv_read_delay	Storage Read Delay	Average delay of reading data from the storage layer in a specified period	≥0 ms	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql106_innodb_row_lock_current_waits	InnoDB Row Locks	Number of row locks being waited by operations on the InnoDB table **NOTE** If there are DDL statements, long transactions, or slow SQL statements, the number of row locks being waited may increase.	≥0 locks/ s	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql107_comdml_ins_and_ins_sel_count	INSERT and INSERT_SELECT Statements per Second	Number of INSERT and INSERT_SELECT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql108_com_commit_count	COMMIT Statements per Second	Number of COMMIT statements executed per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql109_com_rollback_count	ROLLBACK Statements per Second	Number of ROLLBACK statements executed per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql110_innodb_bufpool_reads	InnoDB Storage Layer Read Requests per Second	Number of times that InnoDB reads data from the storage layer per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql111_innodb_bufpool_read_requests	InnoDB Read Requests per Second	Number of InnoDB read requests per second	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql114_innodb_bufpool_read_ahead	InnoDB Bufpool Read Ahead	Number of pages read into the InnoDB buffer pool by the read-ahead background thread	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql115_innodb_bufpool_read_ahead_evicted	InnoDB Bufpool Read Ahead Evicted	Number of pages read into the InnoDB buffer pool by the read-ahead background thread that were subsequently evicted without having been accessed by queries	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql116_innodb_bufpool_read_ahead_rnd	InnoDB Bufpool Read Ahead Rnd	Number of random read-aheads initiated by InnoDB	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql117_innodb_pages_read	InnoDB Pages Read	Number of pages read from the InnoDB buffer pool by operations on InnoDB tables	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql118_innodb_pages_written	InnoDB Pages Written	Number of pages written by operations on InnoDB tables	≥0 counts	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql119_disk_used_ratio	Disk Usage	Disk usage of the monitored object	0–100%	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql120_innodb_buffer_pool_bytes_data	Total Bytes of Buffer Pool	Total number of bytes in the InnoDB buffer pool containing data	≥0 bytes	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql121_innodb_row_lock_time	Row Lock Time	Total time spent in acquiring row locks for InnoDB tables	≥0 ms	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql122_innodb_row_lock_waits	Row Lock Waits	Number of times operations on InnoDB tables had to wait for a row lock	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql123_sort_range	Sorts Using Ranges	Number of sorts that were done using ranges	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql124_sort_rows	Sorted Rows	Number of sorted rows	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql125_sort_scan	Sorts by Scanning Tables	Number of sorts that were done by scanning tables.	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql126_table_open_cache_hits	Hits for Open Tables Cache Lookups	Number of hits for open tables cache lookups	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql127_table_open_cache_misses	Misses for Open Tables Cache Lookups	Number of misses for open tables cache lookups	≥0 counts /min	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql128_long_trx_count	Long-Running Transactions	Number of long transactions that are not closed	≥0 counts	GaussDB(for MySQL) instance nodes	150s
gaussdb_mysql342_iostat_iops_write	I/O Write IOPS	I/O write IOPS	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql344_iostat_iops_read	I/O Read IOPS	I/O read IOPS	≥0 counts/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql346_iostat_throughput_write	I/O Write Bandwidth	Disk write bandwidth per second	≥0 bytes/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql348_iostat_throughput_read	I/O Read Bandwidth	Disk read bandwidth per second	≥0 bytes/s	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql371_taurus_binlog_total_file_counts	Binlog Files	Number of GaussDB(for MySQL) binlog files	≥0	GaussDB(for MySQL) instance nodes	5 minutes
gaussdb_mysql378_create_temp_tbl_per_min	Temporary Tables Created per Minute	Number of temporary tables automatically created on disks per minute when GaussDB(for MySQL) statements are executed	≥0 counts/min	GaussDB(for MySQL) instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql386_undo_spaces_trx_count	Existing Transactions in Undo Space	Number of transactions that are not cleared in the undo space.	≥0	GaussDB(for MySQL) instance nodes	30s
gaussdb_mysql348_taurus_throttle_slice_num	Write Traffic Control	Whether write traffic control is triggered for a DB instance. If its value is greater than 0, write traffic control is triggered. Its value indicates the number of slices whose traffic is limited.	≥0	GaussDB(for MySQL) instance nodes	1 minute
gaussdb_mysql339_taurus_sal_flow_control_instance_read_page_throttle	Read Traffic Control	Whether read traffic control is triggered for a DB instance. If its value is greater than 0, read traffic control is triggered. Its value indicates the number of read pages whose traffic is limited.	≥0	GaussDB(for MySQL) instance nodes	1 minute

Proxy Instance Metrics

Table 18-2 Proxy instance metrics

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds_proxy_frontend_connections	Frontend Connections	Number of connections between applications and the proxy	≥ 0 counts	Proxy instance nodes	1 minute
rds_proxy_backend_connections	Backend Connections	Number of connections between the proxy and GaussDB(for MySQL) databases	≥ 0 counts	Proxy instance nodes	1 minute
rds_proxy_average_response_time	Average Response Time	Average response time	≥0 ms	Proxy instance nodes	1 minute
rds_proxy_query_per_seconds	QPS	Query times of SQL statements	≥ 0 counts	Proxy instance nodes	1 minute
rds_proxy_read_query_proportions	Read Proportion	Proportion of read requests to total requests	0–100%	Proxy instance nodes	1 minute
rds_proxy_write_query_proportions	Write Proportion	Proportion of write requests to total requests	0–100%	Proxy instance nodes	1 minute
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100%	Proxy instance nodes	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0–100%	Proxy instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥0 bytes/s	Proxy instance nodes	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥0 bytes/s	Proxy instance nodes	1 minute
rds_proxy_frontend_connection_creation	Front-End Connections Created per Second	Number of connections created per second between the database proxy and applications	≥ 0 counts	Proxy instance nodes	1 minute
rds_proxy_multi_statement_query	Multi-Statement Queries per Second	Number of multi-statements executed in transactions per second	≥ 0 counts	Proxy instance nodes	1 minute
rds_proxy_transaction_query	Transaction Queries per Second	Number of SELECT statements executed in transactions per second	≥ 0 counts	Proxy instance nodes	1 minute

Dimension

Table 18-3 Metric dimension

Key	Value
gaussdb_mysql_instance_id	GaussDB(for MySQL) instance ID
gaussdb_mysql_node_id	GaussDB(for MySQL) instance node ID
dbproxy_instance_id	Proxy instance ID
dbproxy_node_id	Proxy node ID

18.2 Viewing Monitoring Metrics

18.2.1 Viewing DB Instance Metrics

Scenarios

Cloud Eye monitors status of your DB instances. You can view the metrics of DB instances on the management console. With these metrics, you can identify periods of high resource usage. You can also check error logs or slow query logs to optimize database performance.

Prerequisites

- DB instances are running properly.

  Metrics of the DB instances that are faulty or have been deleted cannot be displayed on the Cloud Eye console, but you can view them after the DB instances are rebooted or become available.

  📖 **NOTE**

  If a DB instance has been faulty for 24 hours, Cloud Eye assumes that the instance no longer exists and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the instance.

- DB instances have kept running properly for about 10 minutes.

  For a newly created DB instance, you need to wait for a while before viewing its metrics.

Viewing DB Instance Metrics

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click ••• and choose **View Metric**.

Figure 18-1 Viewing metrics on the **Basic Information** page
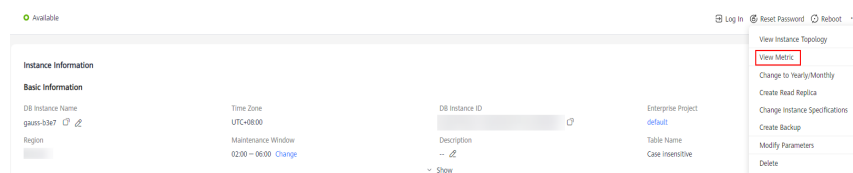
To view metrics of a node, locate the node in the **Node List** area and click **View Metrics** in the **Operation** column.

Step 5 On the displayed Cloud Eye page, view metrics.

You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 1 day, or last 7 days.

----End

Viewing Real-Time DB Instance Metrics

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Advanced O&M**.

Step 6 Under **Real-Time Monitoring**, view real-time monitoring data such as CPU usage, memory usage, SELECT statements per second, DELETE statements per second, and INSERT statements per second.

You can also click **View details** to view more metrics on the Cloud Eye console.

----End

18.2.2 Viewing Proxy Instance Metrics

This section describes how to view proxy instance metrics.

Prerequisites

The GaussDB(for MySQL) instance is running properly.

Read/write splitting has been enabled for the GaussDB(for MySQL) instance. For details, see **How to Use a Proxy Instance to Enable Read/Write Splitting**.

Procedure

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 Choose **Database Proxy** in the navigation pane, locate a proxy instance, and click **View Metrics** in the **Operation** column.

You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 1 day, or last 7 days.

----End

18.3 Configuring Monitoring by Seconds

GaussDB(for MySQL) supports Monitoring by Seconds. You can set the monitoring interval to 1 second or 5 seconds to view the metric values.

Billing

GaussDB(for MySQL) provides monitoring every 60 seconds for free, but you are billed for Monitoring by Seconds. Its pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.

Table 18-4 Price details

Region	Monitoring Interval	Pay-per-Use (USD/Hour)
CN East-Shanghai1, CN North-Beijing4, CN South-Guangzhou, CN Southwest-Guiyang1, CN North-Ulanqab1, and CN South-Guangzhou-InvitationOnly	1s	0.024
	5s	0.012
AP-Singapore, AP-Jakarta, RU-Moscow2, CN-Hong Kong, AP-Bangkok, and TR-Istanbul	1s	0.032
	5s	0.016
LA-Sao Paulo1	1s	0.054
	5s	0.027

Enabling Monitoring by Seconds

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click **Performance**.

Step 7 In the upper part of the page, click **Enable Monitoring by Seconds**.

Step 8 In the displayed dialog box, click ⬭ next to **Monitoring by Seconds**, select a collection interval, and click **OK**.

> After you enable this function, monitoring data will be reported and displayed by the second after about five minutes.

Step 9 In the navigation pane, click **Advanced O&M** > **Real-Time Monitoring** to view metric data.

- View the current data collection period in the upper part of the page.
- Monitoring by Seconds supports the following metrics: CPU usage, memory usage, SELECT statements per second, DELETE statements per second, and INSERT statements per second. You can click **View details** to view more metrics. For details about the metrics, see **GaussDB(for MySQL) Metrics**.
- If you need to change the collection period, see **Modifying Collection Interval**.

----End

Disabling Monitoring by Seconds

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 3 Click **Performance**.

Step 4 In the upper part of the page, click **Enable Monitoring by Seconds**.

Step 5 In the displayed dialog box, click 🔵 next to **Monitoring by Seconds** and click **OK**.

> After you disable this function, monitoring data will be reported and displayed by the minute after about five minutes.

----End

Modifying Collection Interval

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 3 Click **Performance**.

Step 4 In the upper part of the page, click **Enable Monitoring by Seconds**.

Step 5 Select the monitoring interval and click **OK**.

> Monitoring data will be reported based on the new collection interval about 5 minutes later.

Figure 18-2 Modifying the collection interval

----End
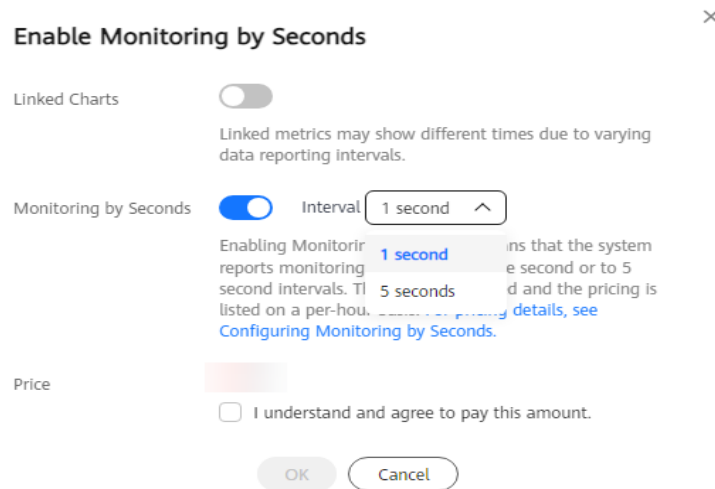
APIs

- **Configuring the Monitoring By Seconds Function**
- **Querying the Configuration of Monitoring by Seconds**

18.4 Configuring Alarm Rules

18.4.1 Creating an Alarm Rule for a DB Instance

Scenarios

You can create alarm rules for a DB instance to configure the monitored objects and notification policies and then stay aware of the DB instance status.

The following parameters can be configured: alarm rule names, services, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Creating an Alarm Rule for a DB Instance

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

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3 In the navigation pane, choose **Cloud Service Monitoring**.

Step 4 Click the **GaussDB(for MySQL)** dashboard.

Step 5 In the instance list, locate the target instance and choose **More** > **Create Alarm Rule** in the **Operation** column.

Step 6 On the displayed page, configure parameters as needed. For details, see **Creating an Alarm Rule**.

1. Configure the alarm rule name and description.

Figure 18-3 Configuring the alarm rule name and description
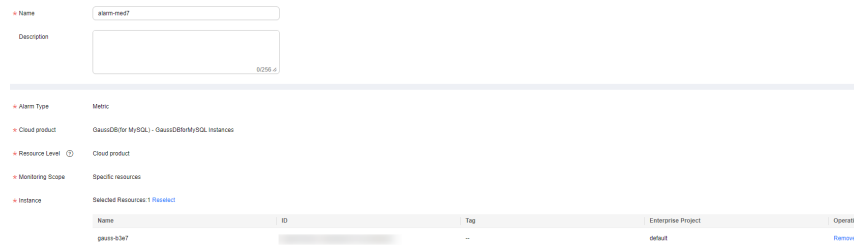
Table 18-5 Name and **Description**

Parameter	Description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed. Example value: **alarm-b6al**
Description	(Optional) Provides supplementary information about the alarm rule.

2. Configure alarm rule parameters.

Figure 18-4 Configuring an alarm rule
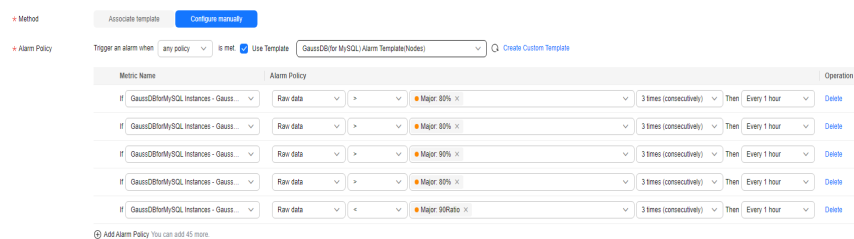
Table 18-6 Alarm rule parameters

Parameter	Description
Method	Select an associated template, use an existing template or create a custom template as required. – Modifying the template will also modify its associated alarm rules. – If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.

Parameter	Description
Template	Select the template to be used. You can select a default alarm template or create a custom template. For details about how to create a custom template, see **Creating a Custom Template**
Alarm Policy	Specifies the policy for triggering an alarm. A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm will be triggered.

3. Configure alarm notification parameters.

Figure 18-5 Alarm Notification

Table 18-7 Alarm Notification parameters

Parameter	Description
Alarm Notification	Specifies whether to send notifications to users over different protocols, such as SMS, email, voice notification, HTTP, HTTPS, FunctionGraph (function), FunctionGraph (workflow), WeCom chatbot, DingTalk chatbot, Lark chatbot, and WeLink chatbot.
Notification Type	The following three options are available: – **Notification policies**: Flexible alarm notifications by severity and more notification channels are provided. – **Notification groups**: Configure notification templates on Cloud Eye. – **Topic subscriptions**: Configure notification templates on SMN.
Notification Policies	If **Notification policies** is selected for **Notification Recipient**, you need to select one or more notification policies. You can specify the notification group, window, template, and other parameters in a notification policy.

Parameter	Description
Notification Group	If **Notification groups** is selected for **Notification Recipient**, select the notification groups to which alarm notifications will be sent.
Notification Object	Specifies the object to which alarm notifications will be sent. You can select the account contact or a topic name. This parameter is available only if **Topic subscriptions** is selected for **Notification Recipient**. – **Account contact** is the mobile phone number and email address of the registered account. – **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 Template	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. You can select an existing template or create a new one.
Notification Window	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. Specifies the time window during which Cloud Eye sends notifications. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within this window.
Trigger Condition	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. Specifies the condition for triggering the alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.

4. Configure the enterprise project and tag.

Figure 18-6 Advanced Settings
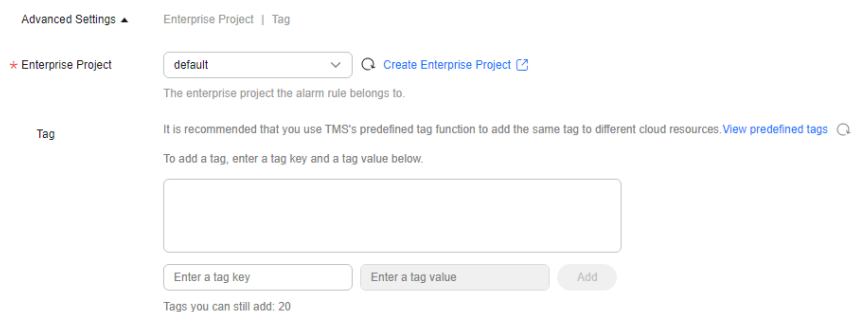
Table 18-8 Enterprise Project and **Tag**

Parameter	Description
Enterprise Project	Specifies the enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule. For details about how to create an enterprise project, see **Creating an Enterprise Project**.
Tag	Adding tags helps you better identify and manage your DB instances.

Step 7 Click **Create**.

----End

Creating an Alarm Rule for a Metric

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ≡ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using either of the following methods:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click ⋯ and choose **View Metric**.

- In the **Node List** area of the **Basic Information** page, locate a node and click **View Metrics** in the **Operation** column.

Step 5 Locate the metric that you want to create an alarm for and click ⊞ in the upper right corner of the metric.

Figure 18-7 Creating an alarm rule for a metric
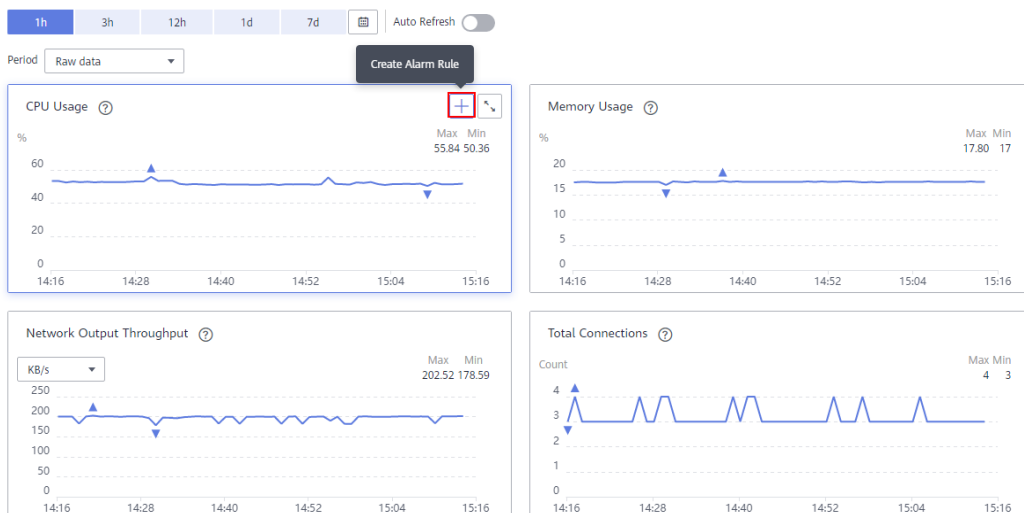
Step 6 On the **Create Alarm Rule** page, configure parameters as needed. For details, see **Creating an Alarm Rule**.

1. Configure the alarm rule name and description.

   Figure 18-8 Configuring the alarm rule name and description
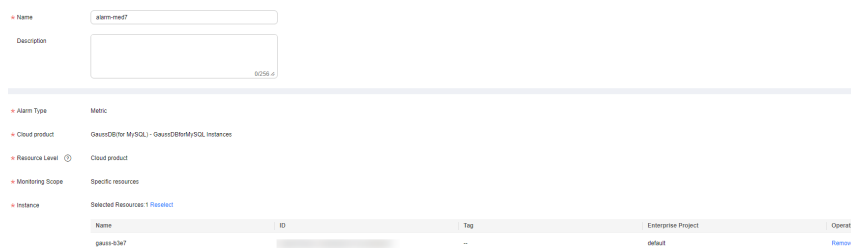
   Table 18-9 Name and **Description**

Parameter	Description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed. Example value: **alarm-b6al**
Description	(Optional) Provides supplementary information about the alarm rule.

2. Configure alarm rule parameters.

   Figure 18-9 Configuring alarm rule parameters
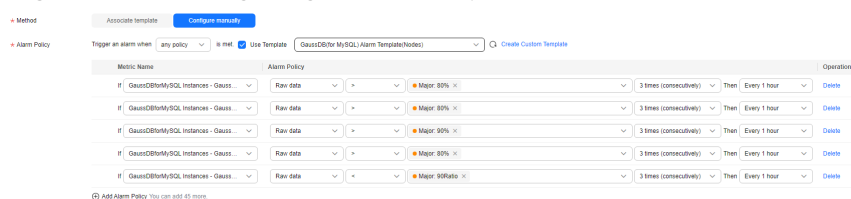
Table 18-10 Alarm rule parameters

Parameter	Description
Method	The default value is **Configure manually**. – Modifying the template will also modify its associated alarm rules. – If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.
Alarm Policy	Specifies the policy for triggering an alarm. A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm will be triggered.

3. Configure alarm notification parameters.

Figure 18-10 Alarm Notification
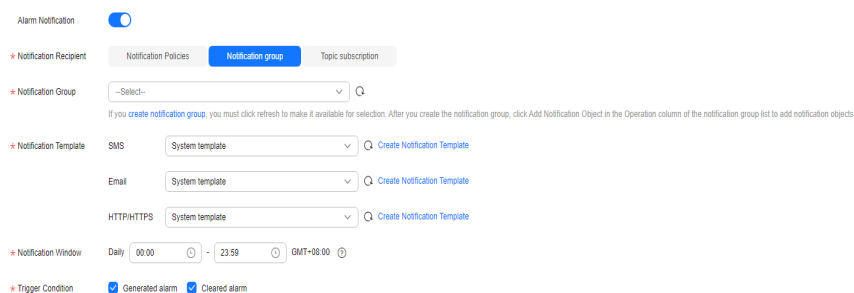
Table 18-11 Alarm Notification parameters

Parameter	Description
Alarm Notification	Specifies whether to send notifications to users over different protocols, such as SMS, email, voice notification, HTTP, HTTPS, FunctionGraph (function), FunctionGraph (workflow), WeCom chatbot, DingTalk chatbot, Lark chatbot, and WeLink chatbot.
Notification Type	The following three options are available: – **Notification policies**: Flexible alarm notifications by severity and more notification channels are provided. – **Notification groups**: Configure notification templates on Cloud Eye. – **Topic subscriptions**: Configure notification templates on SMN.

Parameter	Description
Notification Policies	If **Notification policies** is selected for **Notification Recipient**, you need to select one or more notification policies. You can specify the notification group, window, template, and other parameters in a notification policy.
Notification Group	If **Notification groups** is selected for **Notification Recipient**, select the notification groups to which alarm notifications will be sent.
Notification Object	Specifies the object to which alarm notifications will be sent. You can select the account contact or a topic name. This parameter is available only if **Topic subscriptions** is selected for **Notification Recipient**. – **Account contact** is the mobile phone number and email address of the registered account. – **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 Template	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. You can select an existing template or create a new one.
Notification Window	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. Specifies the time window during which Cloud Eye sends notifications. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within this window.
Trigger Condition	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. Specifies the condition for triggering the alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.

4. Configure the enterprise project and tag.

Figure 18-11 Advanced Settings

Advanced Settings ▲   Enterprise Project | Tag

* Enterprise Project   default   Q   Create Enterprise Project [↗]

The enterprise project the alarm rule belongs to.

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

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

Enter a tag key   Enter a tag value   Add

Tags you can still add: 20

Table 18-12 Enterprise Project and **Tag**

Parameter	Description
Enterprise Project	Specifies the enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule. For details about how to create an enterprise project, see **Creating an Enterprise Project**.
Tag	Adding tags helps you better identify and manage your DB instances.

Step 7   Click **Create**.

----End

18.4.2 Creating an Alarm Rule for a Proxy Instance

Scenarios

You can create alarm rules for a proxy instance to configure the monitored objects and notification policies and then stay aware of the proxy instance status.

The following parameters can be configured: alarm rule names, alarm thresholds, monitoring period, and whether to send notifications.

Procedure

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

Step 2   Click ☰ in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3   In the navigation pane, choose **Cloud Service Monitoring**.

Step 4   On the displayed page, click the **Database Proxy Service** dashboard. In the instance list, locate the target instance and choose **More** > **Create Alarm Rule** in the **Operation** column.

Step 5 On the displayed page, configure parameters as needed. For details, see **Creating an Alarm Rule**.

1. Configure the alarm rule name and description.

Figure 18-12 Configuring the alarm rule name and description
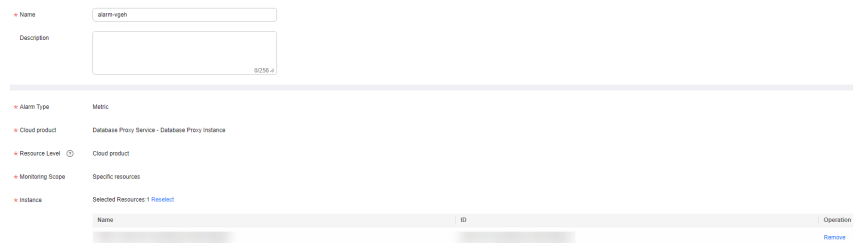
Table 18-13 Name and **Description**

Parameter	Description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed. Example value: **alarm-b6al**
Description	(Optional) Provides supplementary information about the alarm rule.

2. Configure alarm rule parameters.

Figure 18-13 Configuring alarm rule parameters
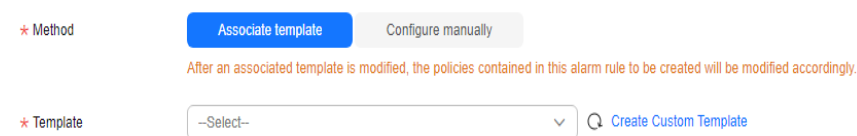
Table 18-14 Alarm rule parameters

Parameter	Description
Method	Select an associated template, use an existing template or create a custom template as required. – Modifying the template will also modify its associated alarm rules. – If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.

Parameter	Description
Template	Select the template to be used. You can select a default alarm template or create a custom template. For details about how to create a custom template, see **Creating a Custom Template**
Alarm Policy	Specifies the policy for triggering an alarm. A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm will be triggered.

3. Configure alarm notification parameters.

Figure 18-14 Alarm Notification

Table 18-15 Alarm Notification parameters

Parameter	Description
Alarm Notification	Specifies whether to send notifications to users over different protocols, such as SMS, email, voice notification, HTTP, HTTPS, FunctionGraph (function), FunctionGraph (workflow), WeCom chatbot, DingTalk chatbot, Lark chatbot, and WeLink chatbot.
Notification Type	The following three options are available: – **Notification policies**: Flexible alarm notifications by severity and more notification channels are provided. – **Notification groups**: Configure notification templates on Cloud Eye. – **Topic subscriptions**: Configure notification templates on SMN.
Notification Policies	If **Notification policies** is selected for **Notification Recipient**, you need to select one or more notification policies. You can specify the notification group, window, template, and other parameters in a notification policy.

Parameter	Description
Notification Group	If **Notification groups** is selected for **Notification Recipient**, select the notification groups to which alarm notifications will be sent.
Notification Object	Specifies the object to which alarm notifications will be sent. You can select the account contact or a topic name. This parameter is available only if **Topic subscriptions** is selected for **Notification Recipient**. – **Account contact** is the mobile phone number and email address of the registered account. – **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 Template	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. You can select an existing template or create a new one.
Notification Window	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. Specifies the time window during which Cloud Eye sends notifications. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within this window.
Trigger Condition	This parameter is only available if **Notification groups** or **Topic subscriptions** is selected for **Notification Recipient**. Specifies the condition for triggering the alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.

4. Configure the enterprise project and tag.

Figure 18-15 Advanced Settings

Table 18-16 Enterprise Project and **Tag**

Parameter	Description
Enterprise Project	Specifies the enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.
	For details about how to create an enterprise project, see **Creating an Enterprise Project**.
Tag	Adding tags helps you better identify and manage your DB instances.

Step 6 Click **Create**.

----End

18.5 Event Monitoring

18.5.1 Introducing Event Monitoring

Event monitoring provides reporting, query, and alarm functions for event data. You can create alarm rules for both system events and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on GaussDB(for MySQL) that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, such as deleting a read replica or changing instance specifications.

Event monitoring provides an API for reporting custom events (abnormal events or important change events) generated by services to Cloud Eye.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see **Events Supported by Event Monitoring**.

18.5.2 Viewing Event Monitoring Data

Scenarios

In event monitoring, you can query system events that are automatically reported to Cloud Eye and custom events reported to Cloud Eye through the API. You can create alarm rules for both system events and custom events. When specific events occur, Cloud Eye generates alarms for you.

Event monitoring is enabled by default.

You can view the event monitoring data.

Procedure

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

Step 2 Click ⚲ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using either of the following methods:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click ⋯ and choose **View Metric**.

- In the **Node List** area of the **Basic Information** page, locate a node and click **View Metrics** in the **Operation** column.

Step 5 Click ‹ to return to the Cloud Eye console.

Step 6 In the navigation pane, choose **Event Monitoring**.

On the displayed **Event Monitoring** page, all system events of the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view events generated in different periods.

Step 7 Locate an event and click **View Event** in the **Operation** column to view details about a specific event.

----End

18.5.3 Creating Alarm Rules for Event Monitoring

Scenarios

You can create alarm rules for event monitoring.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3 In the navigation pane, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the displayed page, configure parameters as needed.

Table 18-17 Parameter description

Parameter	Description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create an enterprise project.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Event Type	Specifies the event type of the metric corresponding to the alarm rule.
Event Source	Specifies the service the event is generated for. Example value: **GaussDB(for MySQL)**
Monitoring Scope	Specifies the monitoring scope for event monitoring.
Method	Specifies the event creation method.
Alarm Policy	Events indicate the instantaneous operations users performed on system resources, such as login and logout. For details about events supported by Event Monitoring, see **Events Supported by Event Monitoring**. You can select a trigger mode and alarm severity as needed.

Click [toggle] to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 18-18 Alarm notification parameters

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text message, or through HTTP/HTTPS request to servers.
Notification Object	Specifies the object an alarm notification is to be sent to. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.

Parameter	Description
Validity Period	Cloud Eye sends notifications only within the validity period specified in the alarm rule. If you set **Validity Period** to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification.

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

----End

18.5.4 Events Supported by Event Monitoring

Table 18-19 GaussDB(for MySQL)

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
GaussDB(for MySQL)	Incremental backup failure	TaurusIncrementalBackupInstanceFailed	Major	The network between the instance and the management plane or the OBS is disconnected, or the backup environment created for the instance is abnormal.	Submit a service ticket.	Backup jobs fail.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Read replica creation failure	addReadonlyNodesFailed	Major	The quota is insufficient or underlying resources are exhausted.	Check the read replica quota. Release resources and create read replicas again.	Read replicas fail to be created.
	DB instance creation failure	createInstanceFailed	Major	The quota is insufficient or underlying resources are exhausted.	Check the instance quota. Release resources and create instances again.	Instances fail to be created.
	Read replica promotion failure	activeStandBySwitchFailed	Major	The read replica fails to be promoted to the primary node due to network or server failures. The original primary node takes over services quickly.	Submit a service ticket.	The read replica fails to be promoted to the primary node.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Instance specifications change failure	flavorAlteration Failed	Major	The quota is insufficient or underlying resources are exhausted.	Submit a service ticket.	Instance specifications fail to be changed.
	Faulty DB instance	TaurusInstance RunningStatus Abnormal	Major	The instance process is faulty or the network between the instance and the DFV storage is disconnected.	Submit a service ticket.	Services may be affected.
	DB instance recovered	TaurusInstance RunningStatus Recovered	Major	The instance is recovered.	Observe the service running status.	None.
	Faulty node	TaurusNodeRu nningStatusAb normal	Major	The node process is faulty or the network between the node and the DFV storage is disconnected.	Observe the instance and service running statuses.	A read replica may be promoted to the primary node.
	Node recovered	TaurusNodeRu nningStatusRec overed	Major	The node is recovered.	View the node running status.	None.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Read replica deletion failure	TaurusDeleteReadOnlyNodeFailed	Major	The network between the management plane and the read replica is disconnected or the VM fails to be deleted from IaaS.	Submit a service ticket.	Read replicas fail to be deleted.
	Password reset failure	TaurusResetInstancePasswordFailed	Major	The network between the management plane and the instance is disconnected or the instance is abnormal.	Check the instance status and try again. If the fault persists, submit a service ticket.	Passwords fail to be reset for instances.
	DB instance reboot failure	TaurusRestartInstanceFailed	Major	The network between the management plane and the instance is disconnected or the instance is abnormal.	Check the instance status and try again. If the fault persists, submit a service ticket.	Instances fail to be rebooted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Restoration to new DB instance failure	TaurusRestoreToNewInstanceFailed	Major	The instance quota is insufficient, underlying resources are exhausted, or the data restoration logic is incorrect.	If the new instance fails to be created, check the instance quota, release resources, and try to restore to a new instance again. In other cases, submit a service ticket.	Backup data fails to be restored to new instances.
	EIP binding failure	TaurusBindEIPToInstanceFailed	Major	The binding task fails.	Submit a service ticket.	EIPs fail to be bound to instances.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	EIP unbinding failure	TaurusUnbindEIPFromInstanceFailed	Major	The unbinding task fails.	Submit a service ticket.	EIPs fail to be unbound from instances.
	Parameter modification failure	TaurusUpdateInstanceParameterFailed	Major	The network between the management plane and the instance is disconnected or the instance is abnormal.	Check the instance status and try again. If the fault persists, submit a service ticket.	Instance parameters fail to be modified.
	Parameter template application failure	TaurusApplyParameterGroupToInstanceFailed	Major	The network between the management plane and instances is disconnected or the instances are abnormal.	Check the instance status and try again. If the fault persists, submit a service ticket.	Parameter templates fail to be applied to instances.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Full backup failure	TaurusBackupInstanceFailed	Major	The network between the instance and the management plane or the OBS is disconnected, or the backup environment created for the instance is abnormal.	Submit a service ticket.	Backup jobs fail.
	Read replica promotion	TaurusActiveStandbySwitched	Major	When the primary node is faulty, a read replica is promoted to the primary node.	Check the instance status. If the fault persists, submit a service ticket.	Services are intermittently interrupted.
	Instance read-only	NodeReadonlyMode	Major	The instance supports only query operations.	Submit a service ticket.	After the instance becomes read-only, write requests cannot be processed.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Instance read/write	NodeReadWriteMode	Major	The instance can process both write and read requests.	Submit a service ticket.	None.
	Instance DR switchover	DisasterSwitchOver	Major	If an instance is faulty and unavailable, a switchover is performed to ensure that the instance continues to provide services.	Contact technical support.	The database connection is intermittently interrupted. The DR instance is promoted to primary to provide services.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Database process restarted	TaurusDatabaseProcessRestarted	Major	The database process is stopped due to insufficient memory or high load.	Log in to the Cloud Eye console. Check whether the memory usage increases sharply or the CPU usage is too high for a long time. You can increase the specifications or optimize the service logic.	When the database process is suspended, workloads on the node are interrupted. In this case, the HA service automatically restarts the database process and attempts to recover the workloads.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Connection failure between proxy instance and DB instance	proxy_connection_failure_to_db	Major	The database proxy failed to establish a new connection with the primary node of a DB instance, and it may fail to establish a new connection with a read replica. The DB instance or proxy instance is overloaded, or the network between the them is abnormal.	Change values of related parameters based on metrics (Connections, Active Connections, and CPU Usage) of the DB instance and proxy instance. If the metrics are normal, submit a service ticket.	Service requests accessed through the proxy instance are interrupted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Connection failure between database proxy and read replica	proxy_connection_failure_to_replica	General	The proxy instance failed to establish a new connection with a read replica. The read replica is overloaded, or the network between the proxy instance and read replica is abnormal.	Change values of related parameters based on metrics (Connections, Active Connections, and CPU Usage) of the read replica. If the metrics are normal, submit a service ticket.	Read requests accessed through the proxy instance are interrupted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Proxy instance access to DB instance failure	proxy_connection_failure_cause_security_group	Major	No rules in the security group allow the proxy instance to access the DB instance.	Add the proxy instance addresss to the rules of the security group.	Service requests accessed through the proxy instance are interrupted.

19 Logs and Auditing

19.1 Configuring Log Reporting

You can view database-level logs on the **Logs** page, including error logs and slow SQL query logs.

Scenarios

If you enable log reporting for your DB instance, new logs generated for the instance will be uploaded to **Log Tank Service (LTS)** for management.

Precautions

- You will be billed for this function.
- Ensure that there are available LTS log groups and log streams in the same region as your DB instance.
- Error logs and slow query logs cannot share the same log stream.
- You can bind a new structuring template to an error log stream or slow log query stream, but once selected, the log stream type cannot be changed.
- If a structuring template has been bound to a log stream, ensure that the template type is the same as the log type when you select the log stream. For example, if an error log template has been bound to a log stream, the log stream cannot be used for slow query logs.

Enabling Log Reporting
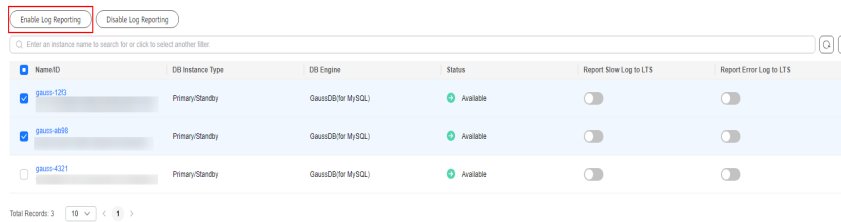
Step 1 **Log in to the management console.**

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **Log Reporting**.

Step 5 Select one or more instances and click **Enable Log Reporting**.

**Figure 19-1** Enabling log reporting for multiple instances

Step 6 In the displayed dialog box, select a log group and log stream, and click **OK**.

☐ NOTE

- Error logs and slow query logs cannot share the same log stream.
- Log reporting cannot be enabled immediately. There is a delay of about 10 minutes.
- You can only enable either error log reporting to LTS or slow log reporting to LTS.
- Audit logs record all requests sent to your DB instance and are stored in LTS.

Figure 19-2 Enabling log reporting
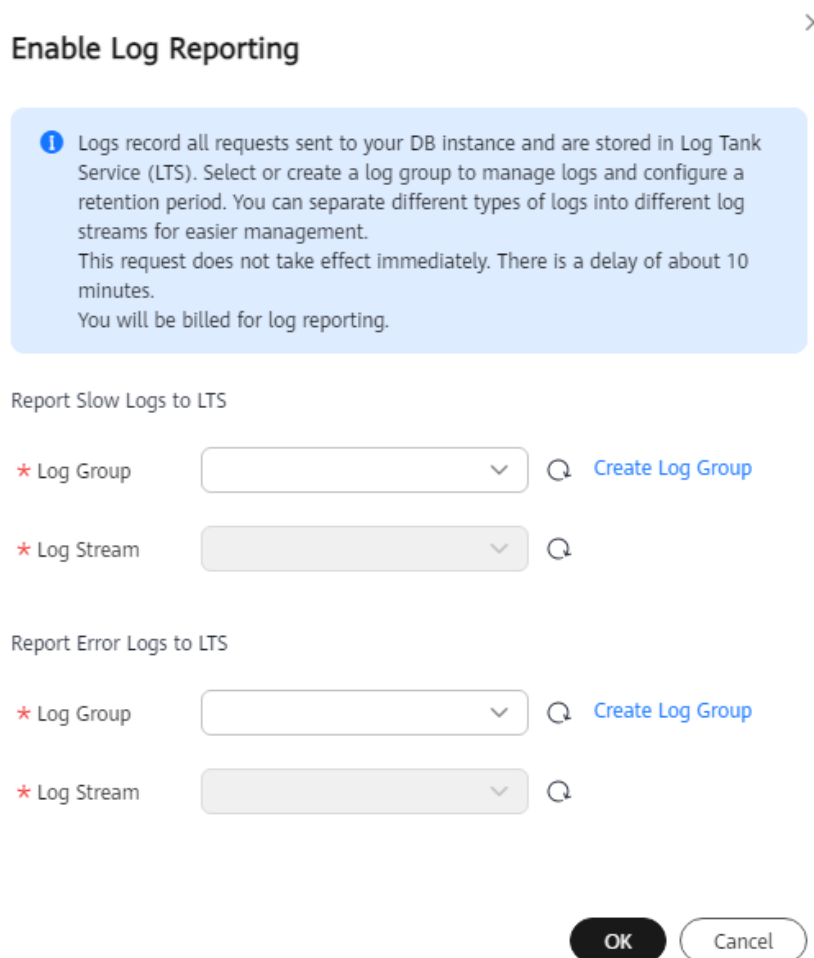
----End

Disabling Log Reporting

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

Step 2 Click ⬚ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **Log Reporting**.

Step 5 Disable log reporting in either of the following ways:

📖 NOTE

- If log reporting is disabled, logs generated for the DB instance will not be reported to LTS.
- This request is not applied immediately. There is a delay of about 10 minutes.

- Disabling log reporting for multiple instances

  a. Select one or more instances and click **Disable Log Reporting**.

  b. In the displayed dialog box, click **OK**.

  Figure 19-3 Disabling log reporting
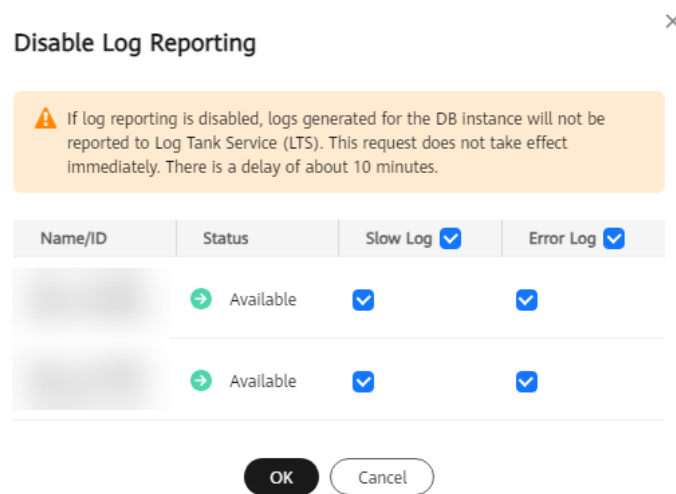
- Disabling log reporting for a single instance

  a. Locate an instance and click 🔵 in the **Report Error Logs to LTS** or **Report Slow Logs to LTS** column.

  b. In the displayed dialog box, click **Yes**.
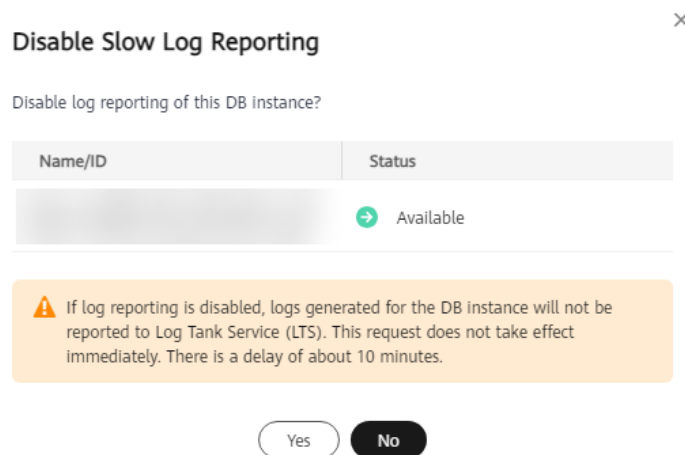
Figure 19-4 Disabling slow log reporting

Disable Slow Log Reporting ✕

Disable log reporting of this DB instance?

Name/ID	Status
	➡ Available

⚠ If log reporting is disabled, logs generated for the DB instance will not be reported to Log Tank Service (LTS). This request does not take effect immediately. There is a delay of about 10 minutes.

Yes   No

Figure 19-5 Disabling error log reporting

Disable Error Log Reporting ✕

Disable log reporting of this DB instance?

Name/ID	Status
	➡ Available

⚠ If log reporting is disabled, logs generated for the DB instance will not be reported to Log Tank Service (LTS). This request does not take effect immediately. There is a delay of about 10 minutes.

Yes   No
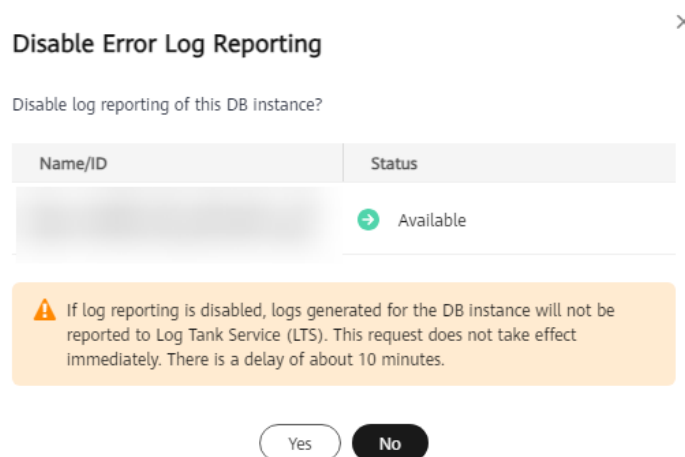
----End

19.2 Managing Error Logs of a DB Instance

Error logs contain logs generated while the database is running. They can help you analyze problems with the database.

Viewing Log Details

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

Step 2 Click ⌖ in the upper left corner and select a region and project.

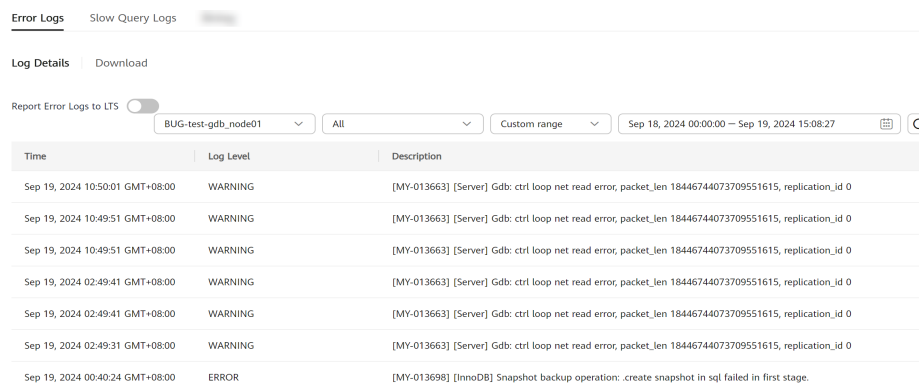**Step 3** Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Logs**.

Step 6 On the **Error Logs** page, view error logs of different nodes, at different log levels, and within a specified time range.

Click the drop-down list in the upper right corner, and select a node name and a log level as needed.

The levels of error logs include ALL, INFO, WARNING, ERROR, FATAL and NOTE.

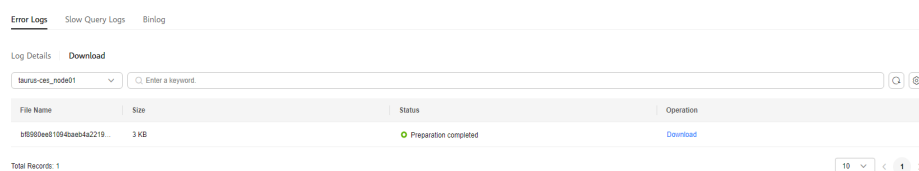Click 🗓 and specify a time period.

Figure 19-6 Viewing error logs

----**End**

Downloading an Error Log

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **Logs**.

Step 3 On the **Error Logs** tab, click **Download**. Locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 19-7 Downloading an error log

● The system automatically loads the download preparation tasks. The loading duration is determined by the log file size and network environment.

  – When the log is being prepared for download, the log status is **Preparing**.

  – When the log is ready for download, the log status is **Preparation completed**.

– If the preparation for download fails, the log status is **Abnormal**.

Logs in the **Preparing** or **Abnormal** state cannot be downloaded.

- Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.

- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

- You can select the logs to be downloaded by node.

----End

Reporting Error Logs to LTS

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Logs**.

Step 6 On the **Error Logs** tab, click next to **Report Error Logs to LTS**.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 19-8 Reporting error logs to LTS

Report Error Logs to LTS

> Logs record all requests sent to your DB instance and are stored in Log Tank Service (LTS).
> This request does not take effect immediately. There is a delay of about 10 minutes.
> You will be billed for log reporting. See LTS pricing details.

* Log Group   [∨]   C   View Log Groups

* Log Stream   [∨]   C

OK   Cancel

----End

APIs

19.3 Managing Slow Query Logs of a DB Instance

Scenarios

Slow query logs record statements that exceed **long_query_time** (10 seconds by default). You can view log details and statistics to identify statements that are executing slowly and optimize the statements.

GaussDB(for MySQL) supports the following statement types:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- ALTER
- DROP

Parameter Description

Table 19-1 Parameters related to slow queries

Parameter	Description
long_query_time	Specifies how many seconds an SQL query has to take to be recorded in slow query logs. The default value is 10s. You are advised to set this parameter to 1s. The lock wait time is not calculated into the query time.
log_queries_not_using _indexes	Specifies whether to record the slow query that without indexes. The default value is **OFF**.
log_throttle_queries_n ot_using_indexes	Specifies the SQL statement that can be written to the slow query log every minute. The default value is **0**.

Viewing Slow Query Log Details

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

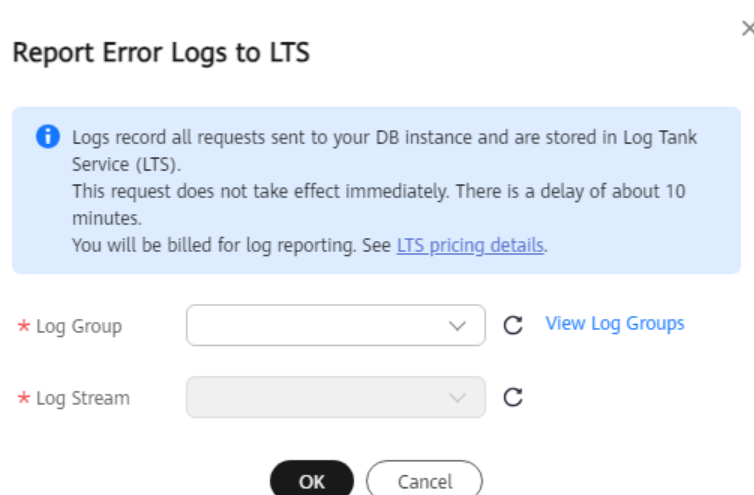**Step 3** Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Logs**.

Step 6 On the **Slow Query Logs** page, view the slow query log details.

You can view slow query logs of different nodes and SQL statement types in a given database.

Supported SQL statement types include SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, and DROP.
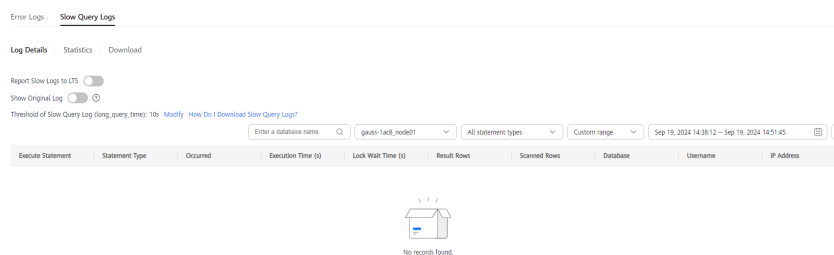
You can also view slow query logs in a specified time period by clicking ⊞ and specifying a time period.

Figure 19-9 Viewing slow query logs

----End

Enabling Show Original Log

☐☐ NOTE

By default, SQL statements are displayed anonymously. If **Show Original Log** is enabled, SQL statements in the logs will be displayed in plaintext.

Logs displayed in plaintext will be automatically deleted 30 days later. If a DB instance is deleted, its related logs will also be deleted.

Step 1 On the **Instances** page, click the instance name.
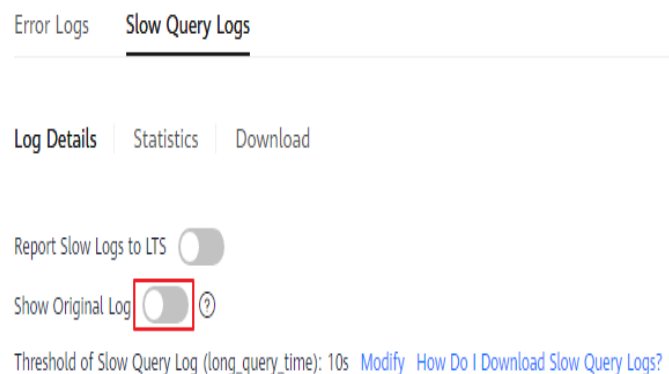
Step 2 In the navigation pane, choose **Logs**.

Step 3 On the **Slow Query Logs** tab, click ⬤ next to **Show Original Log**.

Figure 19-10 Enabling Show Original Log

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

----End

Viewing Statistics

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **Logs**. On the **Slow Query Logs** tab, click **Statistics** to view details.

Figure 19-11 Statistics

☐ NOTE

- On the **Statistics** page, only one of the SQL statements of the same type is displayed as an example. For example, if two select sleep(N) statements, **select sleep(1)** and **select sleep(2)**, are executed in sequence, only **select sleep(N)** will be displayed.

- However, if **Show Original Log** is enabled, all of the slow SQL statements are displayed. For example, if **select sleep(1)** and **select sleep(2)** are executed in sequence, both of them will be displayed.

- **No. and Ratio of SQL Executions** indicates the ratio of the slow executions to the total executions of the SQL statement.

- On the **Statistics** page, only the latest 5,000 slow SQL statements within a specified period are analyzed.

- You can filter slow log statistics by database name (which cannot contain any special characters), statement type, or time period. The database name supports only exact search.

- If any database name in the slow log statistics contains special characters such as < > ', the special characters will be escaped.

----End

Downloading a Slow Query Log

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **Logs**.

Step 3 On the **Slow Query Logs** tab, click **Download**. Locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 19-12 Downloading a slow query log

- The system automatically loads the download preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** state cannot be downloaded.

- Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.

- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

- You can select the logs to be downloaded by node.

----End
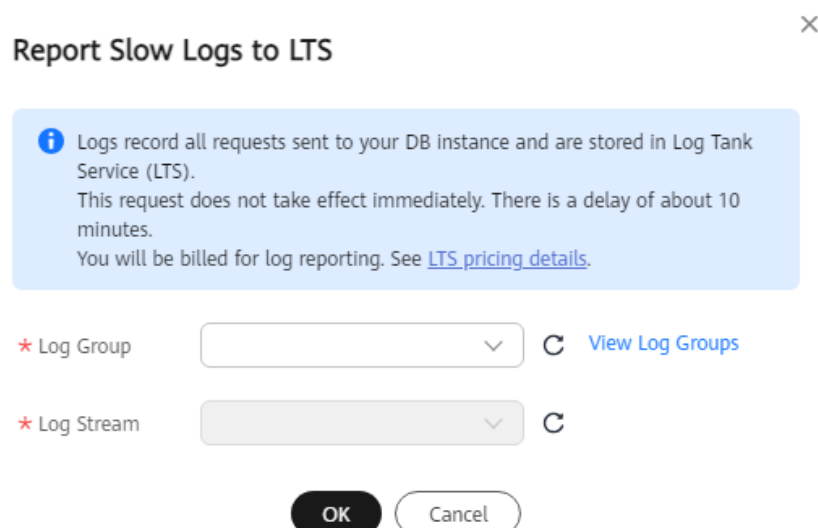
Reporting Slow Logs to LTS

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **Logs**.

Step 3 On the **Slow Query Logs** page, click ⬤ next to **Report Slow Logs to LTS**.

Step 4 Select an LTS log group and log stream and click **OK**.

Figure 19-13 Reporting slow logs to LTS

----End

APIs

[Querying Slow Query Logs](#)

19.4 Configuring SQL Explorer for a DB Instance

Enabling SQL Explorer will allow GaussDB(for MySQL) to store all SQL statement logs for analysis.

You can enable SQL Explorer on the **DAS** console.

Constraints

SQL Explorer cannot record all data. It has the following constraints:

- Some data cannot be recorded if a buffer overrun occurs.
- If the size of an SQL statement exceeds the value of **rds_sql_tracer_max_record_size**, the statement is not recorded by default.

> **NOTE**
>
> **rds_sql_tracer_max_record_size** controls the maximum size of an SQL statement. To change its value, see **Modifying Parameters of a DB Instance**.

19.5 Interconnection with CTS

19.5.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to GaussDB(for MySQL) for further query, audit, and backtracking. **Table 19-2** lists the supported operations.

Table 19-2 GaussDB(for MySQL) operations recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance	instance	createInstance
Creating a read replica	instance	addNodes
Deleting a read replica	instance	deleteNode
Rebooting a DB instance	instance	restartInstance
Changing a database port	instance	changeInstancePort
Changing a security group	instance	modifySecurityGroup
Promoting a read replica to the primary node	instance	instanceSwitchOver
Binding or unbinding an EIP	instance	setOrResetPublicIP

Operation	Resource Type	Trace Name
Deleting a DB instance	instance	deleteInstance
Renaming a DB instance	instance	renameInstance
Changing a failover priority	instance	modifyPriority
Creating a database	instance	createDatabase
Creating a database account	instance	createDatabaseUser
Resetting a password	instance	resetPassword
Deleting a database	instance	dropDatabase
Deleting a database account	instance	dropDatabaseUser
Changing the password of a database user	instance	modifyDatabaseUserPwd
Restoring data to a new DB instance	instance	restoreInstance
Enabling read/write splitting	instance	openProxy
Disabling read/write splitting	instance	closeProxy
Assigning read weights	instance	setProxyWeight
Changing the CPU and memory specifications of an instance	instance	resizeFlavorOrVolume
Configuring monitoring by seconds	instance	openSecondExtend
Upgrading a minor version	instance	upgradeVersion
Adding a tag	instance	addInstanceTags
Authorizing database user permissions	instance	grantDatabaseUser
Revoking database user permissions	instance	revokeDatabaseUser
Applying for a private domain name	instance	createDnsName
Modifying a private domain name	instance	modifyDnsName

Operation	Resource Type	Trace Name
Changing the routing policy of a proxy instance	instance	modifyProxyRouteMode
Changing the port of a proxy instance	instance	modifyProxyPort
Applying for a private domain name for a database proxy instance	instance	proxyCreateDns
Changing a private domain name for a database proxy instance	instance	modifyProxyDnsName
Deleting a private domain name for a database proxy instance	instance	deleteProxyDnsName
Deleting database proxy nodes	instance	reduceProxy
Creating a backup	backup	createManualSnapshot
Configuring an automated backup policy	backup	setBackupPolicy
Deleting a backup	backup	deleteManualSnapshot
Creating a parameter template	parameterGroup	createParameterGroup
Modifying parameters in a parameter template	parameterGroup	updateParameterGroup
Deleting a parameter template	parameterGroup	deleteParameterGroup
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Comparing parameter templates	parameterGroup	compareParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup

19.5.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

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

☐ NOTE

Before using CTS, you need to enable it. For details, see **Enabling CTS**.

Procedure

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

Step 2 Click ⦿ in the upper left corner and select a region and project.

Step 3 In the upper left corner of the page, click ☰ and choose **Management & Governance** > **Cloud Trace Service**.

Step 4 In the navigation pane, choose **Trace List**.

Step 5 Filter conditions to query traces.

Table 19-3 Filtering criteria

Filtering Criteria	Description
Time Range	In the upper right corner, choose **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.
Trace Type	Select **Management** or **Data** ● Management traces record details about creating, configuring, and deleting cloud service resources in your tenant account. ● Data traces record operations on data, such as data upload and download. **NOTE** – If you select **Data** for **Trace Type**, you can only filter traces by tracker. – The trace list does not record queries.
Trace Source	Select a trace source as needed.
Resource Type	Select a resource type as needed.
Search By	If you select **Resource ID** for **Search By**, you need to enter a resource ID.
Operator	Select a specific operator from the drop-down list.

Filtering Criteria	Description
Trace Status	Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.

Step 6 View the events that meet the search criteria.

Step 7 Click an event name. Details about the event are displayed in the dialog box on the right.

Step 8 Click **Export** in the upper left corner of the list. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to the traces.

For details about key fields in the trace structure, see **Trace Structure** and **Example Traces** in *Cloud Trace Service User Guide*.

----End

20 Task Center

20.1 Viewing a Task

You can view the progresses and results of instant and scheduled tasks on the **Task Center** page.
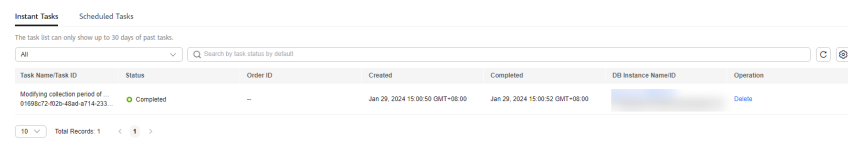
Viewing an Instant Task

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **Task Center**. On the displayed **Instant Tasks** tab page, locate the target task and view its details.

Figure 20-1 Viewing an instant task

- Identify a task based on the task name/task ID, order ID, and instance name/ID, and view the task creation time and end time.
- Click the **All** drop-down list box in the upper part to view the task execution progress and status in a specified period. The default period is all time. The task list shows tasks that have been executed in the past 30 days.
- Click the filter box in the upper part to query the desired instant tasks by task name and task status.
  - Task status: **Running**, **Completed**, and **Failed**

– Task name:

- Creating a GaussDB(for MySQL) instance

- Creating a GaussDB(for MySQL) read replica

- Rebooting a GaussDB(for MySQL) instance

- Changing a GaussDB(for MySQL) instance port

- Promoting a GaussDB(for MySQL) read replica to the primary node

- Binding an EIP to a GaussDB(for MySQL) instance

- Unbinding an EIP from a GaussDB(for MySQL) instance

- Changing the instance name for a GaussDB(for MySQL) instance

- Changing a security group for a GaussDB(for MySQL) instance

- Deleting a GaussDB(for MySQL) instance

- Deleting a GaussDB(for MySQL) read replica

- Changing the specifications of a GaussDB(for MySQL) instance

- Restoring to a new GaussDB(for MySQL) instance

- Changing private IP address

- Modifying collection period of Monitoring by Seconds

- Adding database proxy nodes

- Deleting database proxy nodes

- Enabling database proxy

- Disabling database proxy

- Changing IP address of a proxy instance

- Changing proxy instance specifications

- Enabling or disabling SSL

- Changing consistency level of a proxy instance

- Changing read weights of nodes

- Restoring to an existing DB instance

- Restoring tables to a point in time

- Creating a database

- Deleting a database

- Creating a database account

- Deleting a database account

- Changing the password of a database user

- Changing the host IP address of a database user

- Authorizing database user permissions

- Deleting database user permissions

- Rebooting a node

- Changing read/write splitting address

- Changing a node name

- Increasing specifications of a serverless instance

- Decreasing specifications of a serverless instance

- Changing the port of a proxy instance

- Applying for a private domain name for a proxy instance

- Changing the private domain name of a proxy instance

- Deleting the private domain name of a proxy instance

- Changing the routing policy of a proxy instance

- Enabling or disabling SSL for a proxy instance

- Applying for a private domain name for the DB instance

- Changing the private domain name of the DB instance

- Creating the primary instance for a RegionlessDB cluster

- Creating standby instances for a RegionlessDB cluster

- Deleting a RegionlessDB cluster

- Setting write forwarding for a RegionlessDB cluster

- Modifying the remarks of a GaussDB(for MySQL) database

- Modifying the remarks of a GaussDB(for MySQL) database user

----End

Viewing a Scheduled Task
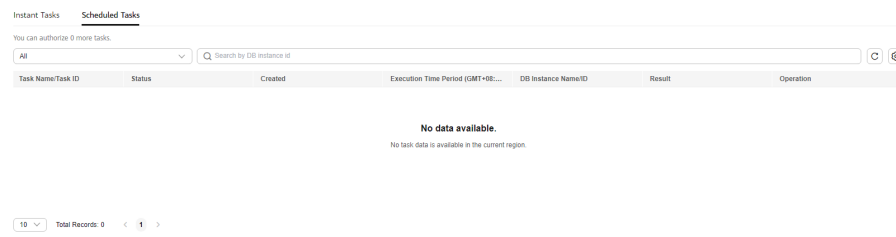
Step 1 **Log in to the management console.**

Step 2 Click ⊚ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **Task Center**. On the **Scheduled Tasks** tab page, view the task progress and results.

Figure 20-2 Viewing a scheduled task

- To identify the task, you can use the instance name/ID or enter the instance ID in the search box in the upper right corner.
- You can enter the instance ID or task status in the search box to determine the desired task and view the task creation time and execution time.

  Task status: **Running**, **Completed**, **Failed**, **Canceled**, **To be executed**, and **To be authorized**.

- Click the **All** drop-down list box in the upper part to view the task execution progress and status in a specified period. The default period is all time.

----End

APIs

- **Obtaining Information About a Task with a Specified ID**
- **Obtaining Instant Tasks**
- **Obtaining Scheduled Tasks**

20.2 Deleting a Task Record

You can delete the task records that no longer need to be displayed.

Constraints

- Deleted task records cannot be recovered. Exercise caution when performing this operation.
- Deleting task records will not delete instances or terminate tasks in progress.

Deleting an Instant Task Record

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 In the navigation pane, choose **Task Center**. Locate the target task on the displayed **Instant Tasks** tab page.

Step 5 Click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

You can delete instant task records with the following statuses:

- Completed
- Failed

----End

Deleting a Scheduled Task Record

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page, choose **Databases** > **GaussDB(for MySQL)**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the task record to be deleted and check whether the task record status is **To be executed** or **To be authorized**.

- If yes, go to **Step 5**.
- If no, go to **Step 6**.

Step 5 Click **Cancel** in the **Operation** column. In the displayed dialog box, click **OK**. Then, click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

You can delete scheduled task records with the following statuses:

- Completed
- Failed
- Canceled
- To be authorized

----End

APIs

- **Canceling a Scheduled Task**
- **Deleting a Task Record**

21 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, and other cloud services manage their own tags.

- You are advised to configure predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Each instance can have up to 20 tags.

Adding a Tag

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **GaussDB(for MySQL)**.

Step 4 On the **Instances** page, click the instance name.

Step 5 Choose **Tags** in the navigation pane and click **Add Tag**. In the displayed dialog box, enter a tag key and value, and click **OK**.

Figure 21-1 Adding a tag

- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with all instances except the current one.
- The tag key must be unique and must consist of 1 to 36 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- The tag value can be empty or consist of 1 to 43 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

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

----End

Editing a Tag

Step 1 On the **Instances** page, click the instance name.

Step 2 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.
- The tag value can be empty or consist of 1 to 43 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

Step 3 View and manage the tag on the **Tags** page.

----End

Deleting a Tag

Step 1 On the **Instances** page, click the instance name.

Step 2 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 3 View that the tag is no longer displayed on the **Tags** page.

----End

APIs

- **Querying Resource Tags**
- **Querying Project Tags**
- **Adding or Deleting Tags in Batches**

22 Quota Management

Scenarios

Quotas put limits on the quantities and capacities of resources available to users, for example, the maximum number of GaussDB(for MySQL) instances that you can create.

If a quota cannot meet your needs, apply for a higher quota.
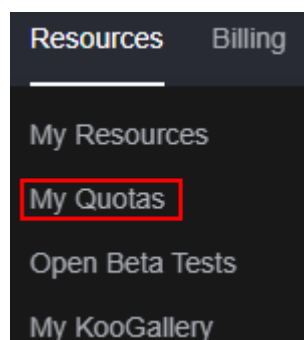
Viewing Quotas

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

Step 2 Click in the upper left corner and select a region and project.

Step 3 Choose **Resources** > **My Quotas** in the upper right corner of the page.

The **Quota** page is displayed.

Figure 22-1 Viewing quotas

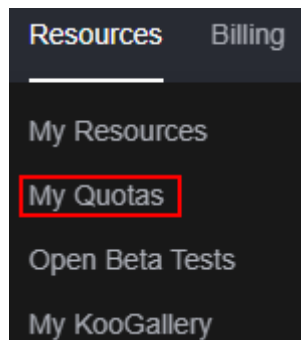**Step 4** View the used and total quotas of each type of resources.

----End

Increasing Quotas

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

Step 2 Click ⊙ in the upper left corner and select a region and project.

Step 3 In the upper right corner of the console page, choose **Resources** > **My Quotas**.

Figure 22-2 Viewing quotas

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 22-3 Increasing quotas

Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for quota adjustment.

Step 6 After all necessary parameters are configured, select the agreement and click **Submit**.

----End

APIs

- **Querying the Instance Quotas of a Tenant**
- **Querying the Resource Quotas of a Specified Enterprise Project**
- **Configuring Resource Quotas for a Specified Enterprise Project**
- **Modifying the Resource Quotas of a Specified Enterprise Project**