GaussDB(for MySQL)

Getting Started

Issue 02

Date 2023-04-18

Huawei Technologies Co., Ltd.

Address:	Huawei Industrial Base Bantian, Longgang Shenzhen 518129 People's Republic of China
Website:	https://www.huawei.com
Email:	support@huawei.com

Contents

1 Operation Guide

You can create and connect to DB instances on the console.

Flowchart

Figure 1-1 Flowchart
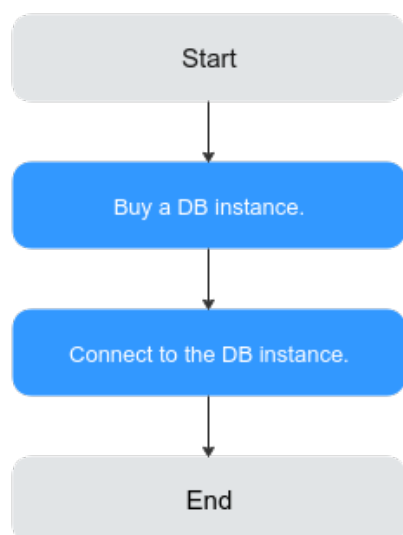
Procedure

Table 1-1 Related operations and documents

Related Operation	Document
Buying a DB instance	**Step 1: Buy a DB Instance**
Connecting to the DB instance	**Step 2: Connect to the DB Instance**

2 Step 1: Buy 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.

Procedure

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

Step 2 On the displayed page, select **Pay-per-use** for **Billing Mode**, configure required information, and click **Next**.

Table 2-1 Basic information

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	A region where the DB instance is located.
DB Instance Name	A name starts with a letter and consists of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0
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.

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.

Table 2-2 Specifications and storage

Parameter	Description
Instance Specifications	Different instance specifications support different numbers of database connections and maximum IOPS. For details, see **Instance Specifications**.
CPU Architecture	The CPU architecture can be x86. Under a CPU architecture, you need to select vCPUs and memory of a DB instance.
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 pay-per-use instance at a time.
Storage Space (GB)	Contains the system overhead required for inode, reserved block, and database operation. Pay-per-use instances: You do not need to specify storage, which will be scaled dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.

Table 2-3 Network

Parameter	Description
VPC	Indicates a dedicated virtual network in which your DB instances are located. It isolates networks for different services. You can select an existing VPC or create a VPC. If no VPC is available, GaussDB(for MySQL) allocates a VPC to you by default. **NOTICE** After a DB instance is created, the VPC cannot be changed.
Subnet	Improves network security by providing dedicated network resources that are logically isolated from other networks. 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. **NOTE** Currently, GaussDB(for MySQL) does not support IPv6.

Parameter	Description
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.

Table 2-4 Database configuration

Parameter	Description
Administrator	The default login name for the database is **root**.
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. Keep this password secure. If lost, the system cannot retrieve it.
Confirm Password	Must be the same as **Administrator Password**.

Table 2-5 Parameter template

Parameter	Description
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"

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**.

Table 2-6 Tags

Parameter	Description
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. A maximum of 20 tags can be added for each instance.

Table 2-7 Batch instance creation

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

Step 3 Confirm the settings for the pay-per-use instance.

- 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 instance status becomes **Available**, the instance can be used.
- 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.

  ☐ NOTE

  To ensure data and instance security, change the database port immediately after the instance is created.

----End

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.

Procedure

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

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

Table 2-8 Basic information

Parameter	Description
Region	A region where the DB instance is located.
DB Instance Name	A name starts with a letter and consists of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	GaussDB(for MySQL)
DB Engine Version	MySQL 8.0
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.
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.

Table 2-9 Specifications and storage

Parameter	Description
Instance Specifications	Different instance specifications support different numbers of database connections and maximum IOPS. For details, see **Instance Specifications**.
CPU Architecture	The CPU architecture can be x86. Under a CPU architecture, you need to select vCPUs and memory of a DB instance.
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.

Parameter	Description
Storage Space (GB)	Contains the system overhead required for inode, reserved block, and database operation. Yearly/monthly instances: 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.

Table 2-10 Network

Parameter	Description
VPC	Indicates a dedicated virtual network in which your DB instances are located. It isolates networks for different services. You can select an existing VPC or create a VPC. If no VPC is available, GaussDB(for MySQL) allocates a VPC to you by default. **NOTICE** After a DB instance is created, the VPC cannot be changed.
Subnet	Improves network security by providing dedicated network resources that are logically isolated from other networks. 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. **NOTE** Currently, GaussDB(for MySQL) does not support IPv6.
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.

Table 2-11 Database configuration

Parameter	Description
Administrator	The default login name for the database is **root**.
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. Keep this password secure. If lost, the system cannot retrieve it.

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

Table 2-12 Parameter template

Parameter	Description
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"
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**.

Table 2-13 Tags

Parameter	Description
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. A maximum of 20 tags can be added for each instance.

Table 2-14 Purchase period (yearly/monthly instances)

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.

Table 2-15 Batch instance creation

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

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 instance status becomes **Available**, the instance can be used.
- 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.

  ☐ NOTE

    To ensure data and instance security, change the database port immediately after the instance is created.

----End

3 Step 2: Connect to the DB Instance

3.1 Overview

Table 3-1 Connection methods

Connect Through	Connection Address	Description	Comments
DAS	Not required	DAS enables you to manage 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 networ k	EIP	If you cannot access the 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.

 NOTE

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

3.2 Connecting to a DB Instance Using DAS (Recommended)

Procedure

Step 1 Log in to the management console.

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

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

Step 4 On the **Instances** page, locate the instance you want to log in and click **Log In** in the **Operation** column.

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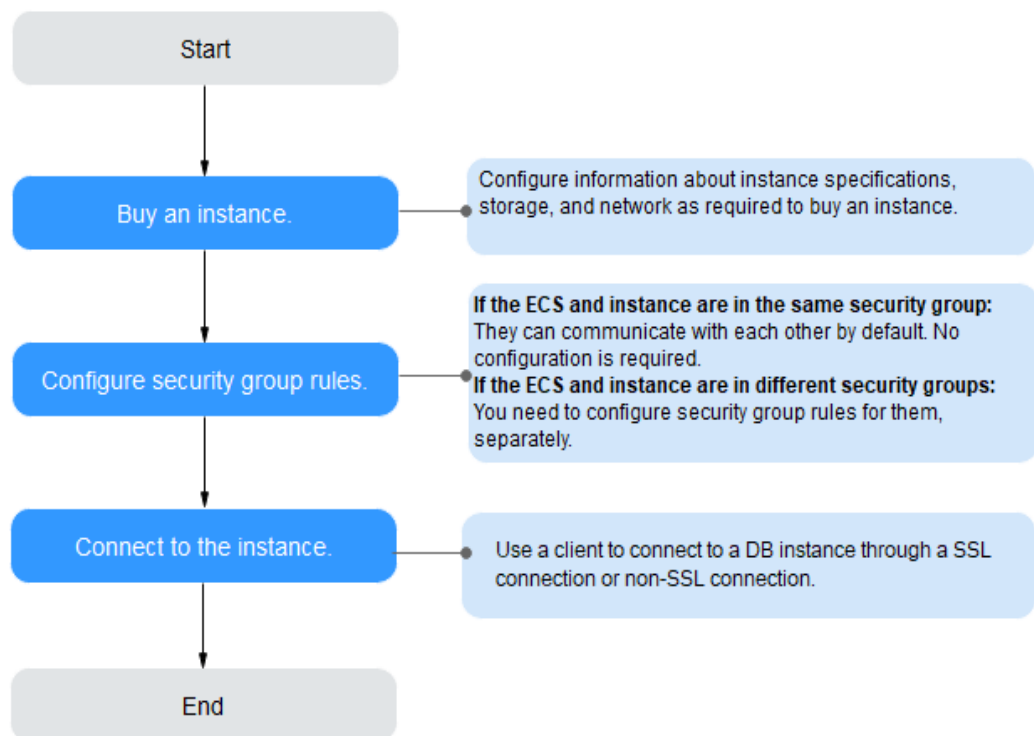
----End

3.3 Connecting to a DB Instance over a Private Network

3.3.1 Process

Figure 3-1 illustrates the process of connecting to an instance over a private network.

Figure 3-1 Connecting to an instance over a private network

3.3.2 Connecting to a DB Instance from a Linux ECS

You can connect to your DB instance using a Linux ECS with a MySQL client installed over a private network.

1. **Purchasing an ECS**
2. **Querying the Private IP Address of the DB Instance to Be Connected**
3. **Testing Connectivity and Installing a MySQL Client**
4. **Using CLI 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, go to **Step 3**.

- If there is a Windows ECS, see **Connecting to a DB Instance from a Windows ECS**.

- If no ECS is available, go to **Step 2**.

Step 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 GaussDB(for MySQL) DB instance for mutual communication.

For details about how to purchase a Linux ECS, see "Purchasing an ECS" in *Elastic Cloud Server User Guide*.

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

Step 4 On the **Basic Information** page of the GaussDB(for MySQL) DB instance, view the region and VPC of the DB instance.

Step 5 Check whether the ECS and GaussDB(for MySQL) DB instance are in the same region and VPC.

- If yes, go to **Querying the Private IP Address of the DB Instance to Be Connected**.

- If no, purchase another ECS or DB instance. If the ECS and DB instance are in different regions, they cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.

- If the ECS and DB instance are in different VPCs, change the VPC of the ECS to that of the DB instance. For details, see "Changing a VPC" in the *Elastic Cloud Server User Guide*.

----End

Querying the Private IP Address of the DB Instance to Be Connected

Step 1 Log in to the management console.

Step 2 On the **Instances** page, click the instance name to go to the **Basic Information** page.

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

----End

Testing Connectivity and Installing a MySQL Client

Step 1 Log in to the ECS. For details, see "Login Using VNC" in the *Elastic Cloud Server User Guide*.

Step 2 On the ECS, check whether the private IP address and database port of the DB instance (obtained in **Querying the Private IP Address of the DB Instance to Be Connected**) can be connected.

telnet *private IP address 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 available.
- If no, check the security group rules.
  - If **Destination** is not **0.0.0.0/0** and **Protocol & Port** is not **All** on the **Outbound Rules** page of the ECS, add the private IP address and port of the DB instance to the outbound rules.
  - To view the inbound rules of the GaussDB(for MySQL) security group, add the private IP address and port of the ECS to the inbound rules. For details, see **Configuring Security Group Rules**.

Step 3 Download the MySQL client installation package for Linux locally. A MySQL client running a version later than that of the DB instance is recommended.

Find the **link** to the required version on the download page. The mysql-community-client-8.0.21-1.el6.x86_64 is used as an example.
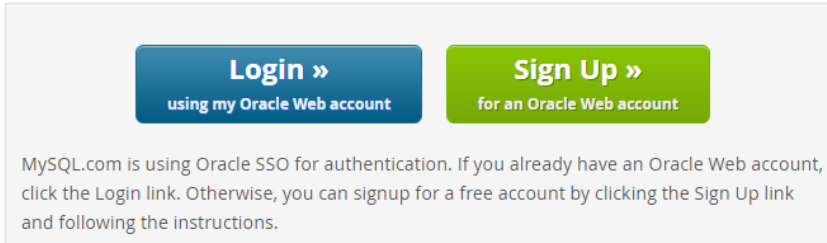
Figure 3-2 Downloading a MySQL client

Step 4 Upload the installation package to the ECS.

Step 5 You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

Step 6 Run the following command to install the MySQL client:

rpm -ivh *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. Example:

  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, you can add the **nodeps** parameter to the command and install the client again. Example:

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

----End

Using CLI to Connect to a DB Instance

In Linux, use a MySQL client to connect to your instance through **an SSL connection** or **non-SSL connection**. The SSL connection encrypts data and is more secure.

- SSL Connection (Recommended)

  a. Log in to the management console.

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

  c. In the **DB Instance Information** area, check whether SSL is enabled next to the **SSL** field.

    - If SSL is enabled, go to **d**.

    - If SSL is disabled, click ⬤▬. In the displayed dialog box, click **Yes** to enable SSL. Then go to **d**.

  d. Click ⬇ next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

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

  f. 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 3-2 Parameter description

Parameter	Description
<host>	Private IP address obtained in **Step 3**.
<port>	Database port obtained in **Step 3**. The default value is 3306.

Parameter	Description
<userName>	Administrator account **root**.
<caName>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

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

Enter password:

- Non-SSL Connection

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

    mysql -h <host> **-P** <port> **-u** <userName> **-p**

    Example:

    mysql -h 192.168.0.16 -P 3306 -u root -p

    Table 3-3 Parameter description

Parameter	Description
<host>	Private IP address obtained in **Step 3**.
<port>	Database port obtained in **Step 3**. The default value is 3306.
<userName>	Administrator account **root**.

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

    Enter password:

    Figure 3-3 Connection succeeded

3.3.3 Connecting to a DB Instance from a Windows ECS

You can connect to your DB instance using a Windows ECS with a database client (for example, MySQL-Front) installed over a private IP address.

1. **Purchasing an ECS**

2. **Querying the Private IP Address of the DB Instance to Be Connected**

3. **Testing Connectivity and Installing MySQL-Front**

4. **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 from a Linux ECS**.
- 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. The ECS must be in the same region, VPC, and security group as the GaussDB(for MySQL) DB instance for mutual communication.

For details about how to purchase a Windows ECS, see "Purchasing an ECS" in *Elastic Cloud Server User Guide*.

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

Step 4 On the **Basic Information** page of the GaussDB(for MySQL) DB instance, view the region and VPC of the DB instance.

Step 5 Check whether the ECS and GaussDB(for MySQL) DB instance are in the same region and VPC.

- If yes, go to **Querying the Private IP Address of the DB Instance to Be Connected**.
- If no, purchase another ECS or DB instance. If the ECS and DB instance are in different regions, they cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.
- If the ECS and DB instance are in different VPCs, change the VPC of the ECS to that of the DB instance. For details, see "Changing a VPC" in the *Elastic Cloud Server User Guide*.

----End

Querying the Private IP Address of the DB Instance to Be Connected

Step 1 Log in to the management console.

Step 2 On the **Instances** page, click the instance name to go to the **Basic Information** page.

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

----End

Testing Connectivity and Installing MySQL-Front

Step 1 Log in to the ECS. For details, see "Login Using VNC" in the *Elastic Cloud Server User Guide*.

Step 2 On the ECS, check whether the private IP address and database port of the DB instance (obtained in **Querying the Private IP Address of the DB Instance to Be Connected**) can be connected.

telnet *private IP address 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 available.
- If no, check the security group rules.
  - If **Destination** is not **0.0.0.0/0** and **Protocol & Port** is not **All** on the **Outbound Rules** page of the ECS, add the private IP address and port of the DB instance to the outbound rules.
  - To view the inbound rules of the GaussDB(for MySQL) security group, add the private IP address and port of the ECS to the inbound rules. For details, see **Configuring Security Group Rules**.

Step 3 Open a browser, and download and install the MySQL-Front tool on the ECS.
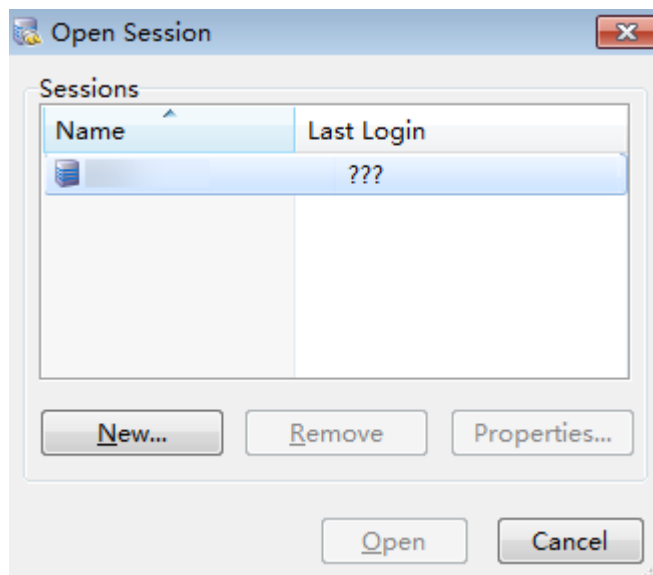
----End

Using MySQL-Front to Connect to a DB Instance
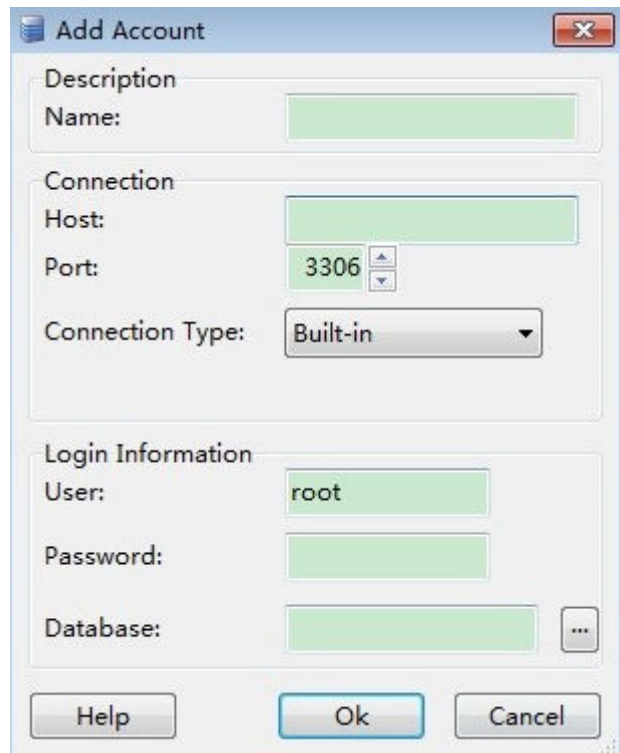
Step 1 Start MySQL-Front.

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

Figure 3-4 Connection management

Step 3 Enter the information of the DB instance to be connected and click **Ok**.

Figure 3-5 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 have created in **Step 3** and click **Open**. If the connection information is correct, the DB instance is successfully connected.

Figure 3-6 Opening a session
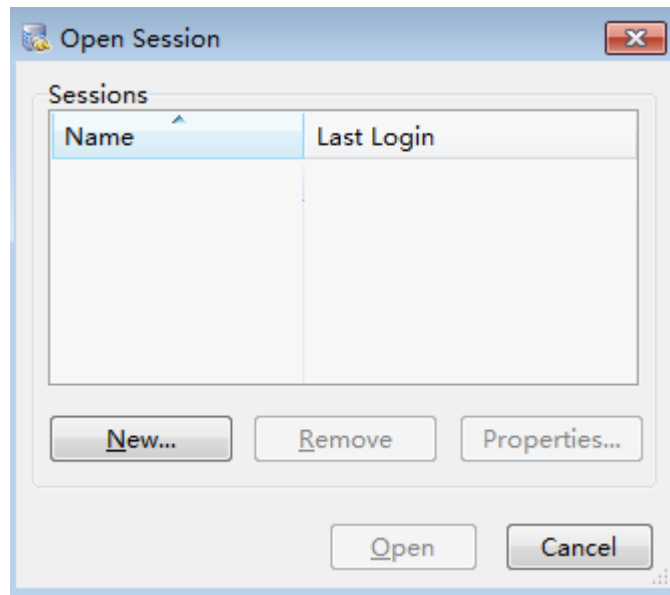
----End

3.3.4 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and instances that have the same security requirements and are mutually trusted in a VPC. To ensure database security and reliability, you need to configure security group rules to allow specific IP addresses and ports to access instances.

Check whether the ECS and 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 instance, respectively.
  - Instance: Configure an inbound rule for the security group to which the 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.

You can configure an inbound rule for an instance.

Precautions

The default security group rule allows all outbound data packets. If an ECS and an instance are in the same security group, they can access each other. When a security group is created, you can configure security group rules to control access to and from 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.

- To access an 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.

If you use **0.0.0.0/0**, all IP addresses can access instances associated with the security group.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **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 on the **Basic Information** page, click the security group name next to the **Security Group** field.

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

You can click ⊕ to add more inbound rules.

Table 3-5 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)

| Parameter	Description	Example Value
	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

3.4 Connecting to a DB Instance over a Public Network

3.4.1 Process

Figure 3-7 illustrates the process of connecting to an instance over a public network.

Figure 3-7 Connecting to an instance over a public network
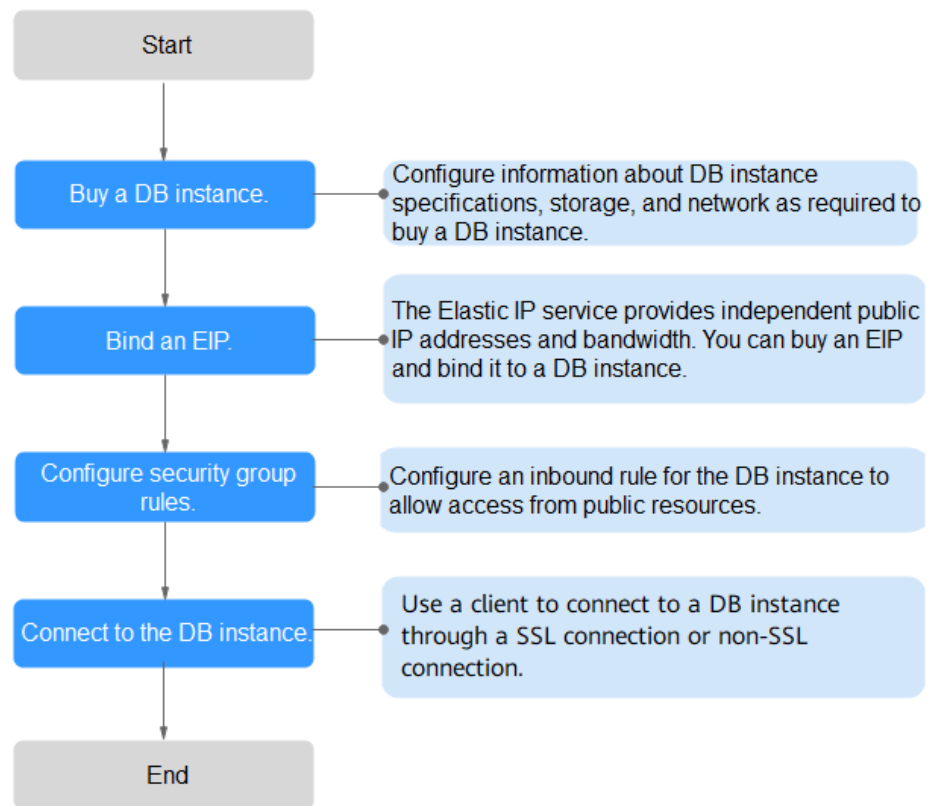
3.4.2 Binding an EIP

Scenarios

You can bind an EIP to a DB instance for public accessibility and can unbind the EIP from the instance if needed.

Precautions

- Public accessibility reduces the security of instances. 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.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **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** in the **Public IP Address (EIP)** field.

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 **Public IP Address (EIP)** field of the **Network Information** area, view the EIP that was bound.

----End

3.4.3 Connecting to a DB Instance from a Linux ECS

If your DB instance and ECS are not in the same region or VPC, you can connect to your DB instance by using either of ways:

- A Linux ECS with a MySQL client installed through an EIP
- A local PC with a MySQL client installed through an EIP

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

1. **Purchasing an ECS**
2. **Binding an EIP**
3. **Querying the EIP of the DB Instance to Be Connected**
4. **Testing Connectivity and Installing a MySQL Client**
5. **Using CLI 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, go to **Step 3**.
- If there is a Windows ECS, see **Connecting to a DB Instance from a Windows ECS**.
- If no ECS is available, go to **Step 2**.

Step 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 User Guide*.

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

Step 4 On the **Basic Information** page of the GaussDB(for MySQL) DB instance, view the region and VPC of the DB instance.

----End

Binding an EIP

You can bind an EIP to a DB instance for public accessibility and can unbind the EIP from the DB instance if needed.

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

For details, see **Binding an EIP**.

Step 1 Log in to the management console.

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

Step 3 Click [icon] in the upper left corner of the page, choose **Database** > **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** in the **Public IP Address (EIP)** field.

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 **Public IP Address (EIP)** field of the **Network Information** area, view the EIP that was bound.

----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 a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **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.

----End

Testing Connectivity and Installing a MySQL Client

Step 1 Log in to the ECS. For details, see "Login Using VNC" in the *Elastic Cloud Server User Guide*.

Step 2 On the ECS, check whether the EIP and database port of the DB instance (obtained in **Querying the EIP of the DB Instance to Be Connected**) 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 available.
- If no, check the security group rules.
  - If **Destination** is not **0.0.0.0/0** and **Protocol & Port** is not **All** on the **Outbound Rules** page of the ECS, add the EIP and port of the DB instance to the outbound rules.
  - To view the inbound rules of the GaussDB(for MySQL) security group, add the EIP and port of the ECS to the inbound rules. For details, see **Configuring Security Group Rules**.

Step 3 Download the MySQL client installation package for Linux locally. A MySQL client running a version later than that of the DB instance is recommended.

Find the **link** to the required version on the download page. The mysql-community-client-8.0.21-1.el6.x86_64 is used as an example.
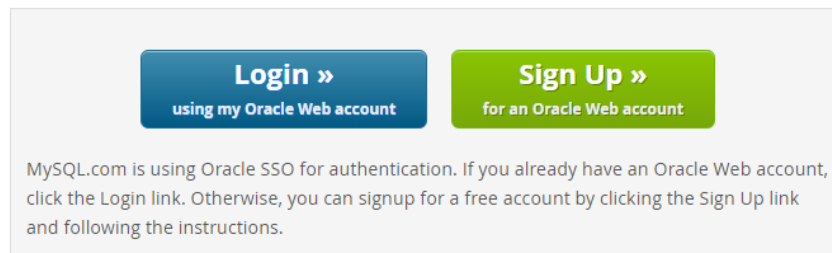
Figure 3-8 Downloading a MySQL client

Step 4 Upload the installation package to the ECS.

Step 5 You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

Step 6 Run the following command to install the MySQL client:

rpm -ivh _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. Example:

  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, you can add the **nodeps** parameter to the command and install the client again. Example:

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

----End

Using CLI to Connect to a DB Instance

In Linux, use a MySQL client to connect to your instance through **an SSL connection** or **non-SSL connection**. The SSL connection encrypts data and is more secure.

- SSL Connection (Recommended)

  a. Log in to the management console.

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

c. In the **DB Instance Information** area, check whether SSL is enabled next to the **SSL** field.

- If SSL is enabled, go to **d**.

- If SSL is disabled, click ⬤. In the displayed dialog box, click **Yes** to enable SSL. Then go to **d**.

d. Click ⬇ next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

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

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

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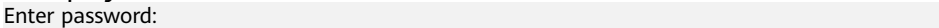
Example:

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

Table 3-6 Parameter description

Parameter	Description
<*host*>	EIP obtained in **Step 5**.
<*port*>	Database port obtained in **Step 5**. 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.

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

Enter password:

- Non-SSL Connection

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

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p**

Example:

mysql -h *192.168.0.16* -P 3306 -u root -p

Table 3-7 Parameter description

Parameter	Description
<*host*>	EIP obtained in **Step 5**.
<*port*>	Database port obtained in **Step 5**. The default value is 3306.
<*userName*>	Administrator account **root**.

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

Enter password:

Figure 3-9 Connection succeeded

3.4.4 Connecting to a DB Instance from a Windows ECS

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.

1. **Binding an EIP to a DB Instance**
2. **Querying the EIP of the DB Instance to Be Connected**
3. **Testing Connectivity and Installing MySQL-Front**
4. **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 from a Linux ECS**.
- 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 User Guide*.

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

Step 4 On the **Basic Information** page of the GaussDB(for MySQL) DB instance, view 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 accessibility and can unbind the EIP from the DB instance if needed.

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

For details, see **Binding an EIP**.

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **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** in the **Public IP Address (EIP)** field.

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 **Public IP Address (EIP)** field of the **Network Information** area, view the EIP that was bound.

----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 a project.

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **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.

----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 available.
- If no, check the security group rules.
    - If **Destination** is not **0.0.0.0/0** and **Protocol & Port** is not **All** on the **Outbound Rules** page of the ECS, add the EIP and port of the DB instance to the outbound rules.
    - To view the inbound rules of the GaussDB(for MySQL) security group, add the EIP and port of the ECS to the inbound rules. 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 3-10 Connection management
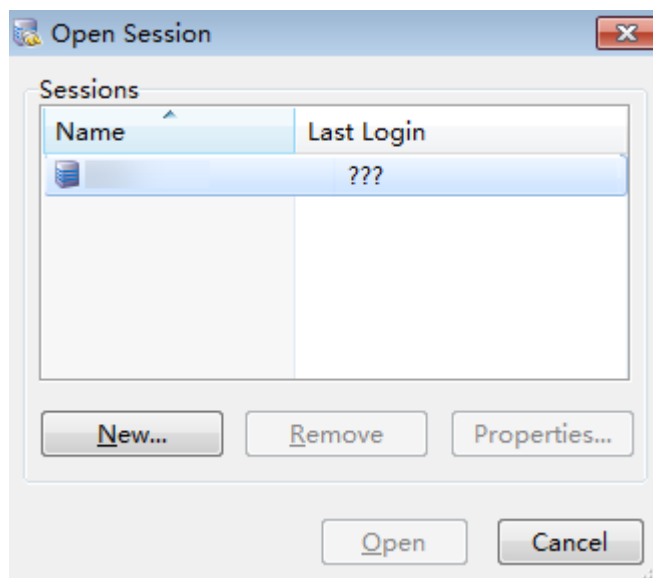
Step 3 Enter the information of the DB instance to be connected and click **Ok**.

Figure 3-11 Adding an account
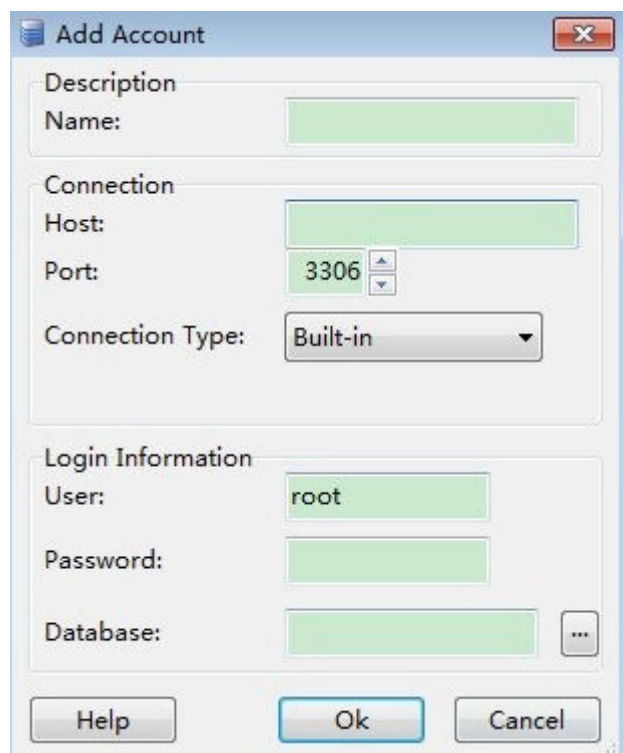
Table 3-8 Parameter description

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	EIP obtained in **Step 5**.
Port	Database port obtained in **Step 5**. 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 have created in **Step 3** and click **Open**. If the connection information is correct, the DB instance is successfully connected.

Figure 3-12 Opening a session
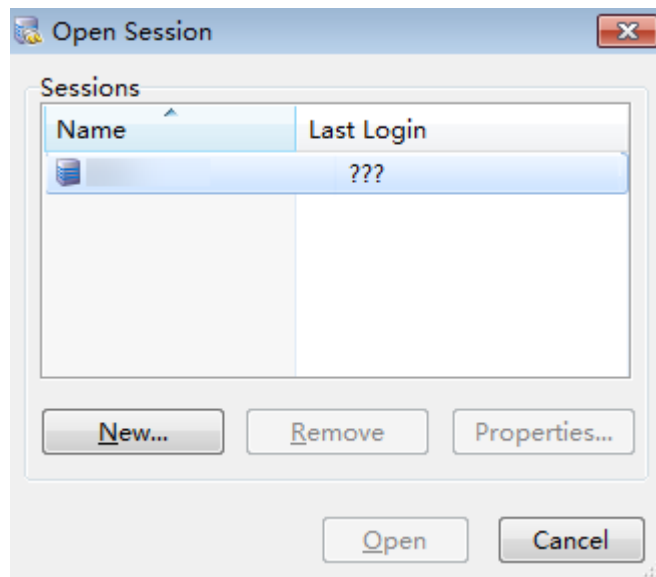
----End

3.4.5 Configuring Security Group Rules

Scenarios

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

To ensure database security and reliability, you need to configure security group rules to allow specific IP addresses and ports to access instances.

When you attempt to access an instance through an EIP, you need to configure an inbound rule for the security group associated with the instance.

Precautions

The default security group rule allows all outbound data packets. If an ECS and an instance are in the same security group, they can access each other. When a security group is created, you can configure security group rules to control access to and from 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.
- To access an 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.

  If you use **0.0.0.0/0**, all IP addresses can access instances associated with the security group.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page, choose **Database** > **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 on the **Basic Information** page, click the security group name next to the **Security Group** field.

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

You can click **+** to add more 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 DB instance through a public network, enter the port of the DB 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**.

Parameter	Description	Example Value
Address	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

3.5 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 data encryption is enabled for newly created GaussDB(for MySQL) instances. Enabling SSL will increase the network connection response time and CPU usage. Before enabling SSL, evaluate the impact on service performance.

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. In the **DB Instance Information** area, click ⬇ next to **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 3-10 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 the GaussDB(for MySQL) instance through JDBC.

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

Table 3-11 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**. ● If you are accessing the instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can view the private IP address in the **Network Information** area on the **Basic Information**.
<instance_port>	Database port of the instance. The default port is **3306**. **NOTE** You can view the private IP address in the **Network Information** area on the **Basic Information**.
<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 3-12**.
<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 3-12**.
<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 3-12**.
<truststore_file>	Path for storing the truststore file configured in **Step 2**.
<password>	Password of the truststore file configured in **Step 2**.

Table 3-12 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 {
    static final String USER = "xxx";
    static final String PASS = "xxx";

    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 3-13 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**. ● If you are accessing the instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can view the private IP address in the **Network Information** area on the **Basic Information**.
<instance_port>	Database port of the instance. The default port is **3306**. **NOTE** You can view the private IP address in the **Network Information** area on the **Basic Information**.
<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):

```
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");
            conn = DriverManager.getConnection(url, "root", "password");
            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

    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

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

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

A Change History

Released On	Description
2023-04-18	This issue is the second official release, which incorporates the following change: Added **Connecting to a DB Instance Using DAS (Recommended)**.
2023-02-28	This issue is the first official release.