TaurusDB

User Guide

Issue 01

Date 2025-07-14

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 Permissions Management

1.1 Creating a User and Granting TaurusDB Permissions

This section describes how to use IAM for fine-grained permissions control over your TaurusDB 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 TaurusDB resources.
- Grant only the permissions required for users to perform specific tasks.
- Entrust an account or cloud service to perform professional and efficient O&M on your TaurusDB resources.

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

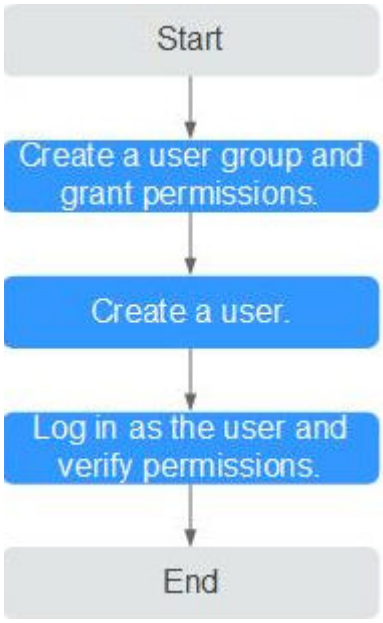
Figure 1-1 describes the procedure for granting permissions.

Prerequisites

Learn about the permissions (see **System-defined permissions**) supported by TaurusDB and choose policies or roles according to your requirements.

Process Flow

Figure 1-1 Process of granting TaurusDB permissions



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

   Create a user group on the IAM console, and attach the **TaurusDB GaussDB FullAccess** policy to the group.
2. Create an IAM user and add it to a user group.

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

   Log in to the TaurusDB console using the created user, and verify that the user only has read permissions for TaurusDB.

   Under the service list, choose **Databases** > **TaurusDB**. On the **Instances** page, click **DB Instance** in the upper right corner. If you can an instance, the required permission policy has already been applied.

1.2 Creating a TaurusDB Custom Policy

Custom policies can be created to supplement the system-defined policies of TaurusDB.

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.

The following lists examples of common TaurusDB custom policies.

Example Custom Policies

- Example 1: Allowing users to create TaurusDB instances

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

- Example 2: Denying TaurusDB 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 TaurusDB instances. Create a custom policy for denying TaurusDB instance deletion, and attach both policies to the group the user belongs to. Then, the user can perform all operations on TaurusDB instances except deleting TaurusDB 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.

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 TaurusDB console.

Procedure

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

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

Table 2-1 Basic information

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	Region where an instance is deployed.
DB Instance 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. • 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. • Each name of the instances created in batches can contain 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
DB Engine Version	Select **TaurusDB V2.0**.
DB Instance Type	Select **Cluster** or **Single**. • **Cluster**: A cluster instance can contain one primary node and 1 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, TaurusDB automatically fails over to a read replica. Cluster 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 between 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 TaurusDB.
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. • **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.

Table 2-2 Instance specifications

Parameter	Description
Instance Specifications	Different instance specifications support different numbers of database connections and maximum IOPS.
CPU Architecture	Select **x86** or **Kunpeng**.
Nodes	Total number of one primary node and read replicas you created for the instance. You can create up to 9 read replicas at a time.

Parameter	Description
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.
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 your databases and data files. After TDE is enabled, you need to select a cryptographic algorithm **AES256** or **SM4** as needed.

Table 2-3 Network

Parameter	Description
VPC	A dedicated virtual network where your instance is located. It isolates networks for different workloads to enhance security. You need to select a VPC and subnet. If no VPC is available, TaurusDB will allocate a default VPC (**default_vpc**) for your instance. You can also use an existing or new VPC and subnet. **NOTICE** After a TaurusDB instance is created, the VPC cannot be changed.
Security Group	A security group enhances security by controlling access to TaurusDB 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, TaurusDB allocates a security group to your instance by default.
IPv6	Before enabling IPv6, ensure that IPv6 has been enabled for the VPC and subnet where the DB instance is located. For details about how to configure IPv6 for the VPC and subnet, see "IPv4/IPv6 Dual-Stack Management" in *Virtual Private Cloud Operation Guide*. After IPv6 is enabled, the DB instance can run in dual-stack mode. It means that the DB instance can use both IPv4 and IPv6 addresses. The DB instance can be accessed through either an IPv4 or IPv6 address, and the communications are independent of each other.

Table 2-4 Database configuration

Parameter	Description
Administrator	The default login name for the database is **root**.
Administrator Password	The 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	Enter the administrator password again.

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 • 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.
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. Select an enterprise project from the drop-down list. The default project is **default**.

Table 2-6 Tag

Parameter	Description
Tag	Tags a DB instance. This configuration is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags.

Table 2-7 Batch 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 DB 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 DB 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 is 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.
- After the instance is created, you can click the instance name to go to the **Basic Information** page. In the **Network Information** area, obtain the private IP address and database port.
- 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 TaurusDB 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
Billing Mode	Select **Yearly/Monthly**.
Region	A region where the DB instance is located.
DB Instance 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. • 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. • Each name of the instances created in batches can contain 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine Version	Select **TaurusDB V2.0**.
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. • **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.

Table 2-9 Specifications and storage

Parameter	Description
Instance Specifications	Different instance specifications support different numbers of database connections and maximum IOPS.
CPU Architecture	x86 or Kunpeng.
Nodes	Total number of one primary node and read replicas you created for the instance. You can create up to 9 read replicas at a time.
Storage Space	Contains the system overhead required for inode, reserved block, and database operation. 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	A dedicated virtual network where your instance is located. It isolates networks for different workloads to enhance security. You need to select a VPC and subnet. If no VPC is available, TaurusDB will allocate a default VPC (**default_vpc**) for your instance. You can also use an existing or new VPC and subnet. **NOTICE** After a TaurusDB instance is created, the VPC cannot be changed.
Security Group	A security group enhances security by controlling access to TaurusDB 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, TaurusDB allocates a security group to your instance by default.

Parameter	Description
IPv6	Before enabling IPv6, ensure that IPv6 has been enabled for the VPC and subnet where the DB instance is located. For details about how to configure IPv6 for the VPC and subnet, see "IPv4/IPv6 Dual-Stack Management" in *Virtual Private Cloud Operation Guide*. After IPv6 is enabled, the DB instance can run in dual-stack mode. It means that the DB instance can use both IPv4 and IPv6 addresses. The DB instance can be accessed through either an IPv4 or IPv6 address, and the communications are independent of each other.

Table 2-11 Database configuration

Parameter	Description
Administrator	The default login name for the database is **root**.
Administrator Password	The 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 2-12 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 • 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.
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.
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 DB 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.

  NOTE

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

----End

3 Connecting to a DB Instance

3.1 Connection Methods

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 TaurusDB instance, you are advised to connect the ECS to the instance over a private IP address.	• Secure and excellent performance • Recommended

Connect Through	Connection Address	Description	Comments
Public network	EIP	If you cannot access the TaurusDB 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 TaurusDB 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 a DB instance using DAS or other database clients.
- If an ECS is in the same VPC as the TaurusDB 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 project.

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

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

Step 5 Enter the database username and password, and click **Test Connection**. After the connection is successful, click **Log In**.

----End

3.3 Connecting to a DB Instance Through the mysql Client

3.3.1 Connecting to a DB Instance over a Private Network

After buying a DB instance, you can connect to it using a Linux ECS with the mysql client installed over a private network. This section describes how to access a DB instance from an ECS using the mysql client.

Step 1: Buy 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 Through MySQL-Front**.
- 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 DB instance for mutual communications.

For details about how to create a Linux ECS, see section "Creating 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 DB instance, view the region and VPC of the DB instance.

Step 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 the mysql Client**.
- If they are in different regions, create another ECS or DB 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 section "Changing a VPC" in *Elastic Cloud Server User Guide*.

----End

Step 2: Test Connectivity and Install the mysql Client

Step 1 Log in to the ECS. For details, see section "Logging In to a Linux ECS Using VNC" in *Elastic Cloud Server User Guide*.

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

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

Step 4 On the ECS, check whether the private IP address and database port of the DB instance can be connected.

telnet *private_IP_address port*

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 private IP address and port of the DB instance.
  - If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS. For details, see **Configuring Security Group Rules**.

Step 5 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 3-1 Downloading a mysql client

Step 6 Upload the installation package to the ECS.

Step 7 Use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

Step 8 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 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

----End

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

1. On the **Instances** page of the TaurusDB console, click the instance name to go the **Basic Information** page.
2. In the **DB 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:
```

3.3.2 Connecting 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 and connect to it from an ECS or a public server 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

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 Through MySQL-Front**.
- 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 create a Linux ECS, see section "Creating 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 DB instance, view the region and VPC of the DB instance.

----End

Step 2: Test Connectivity and Install the mysql Client

Step 1 Log in to the ECS. For details, see section "Logging In to a Linux ECS Using VNC" in *Elastic Cloud Server User Guide*.

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

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

If no EIP has been bound to the DB instance, bind one by referring to **Binding an EIP**.

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

telnet *EIP port*

NOTE

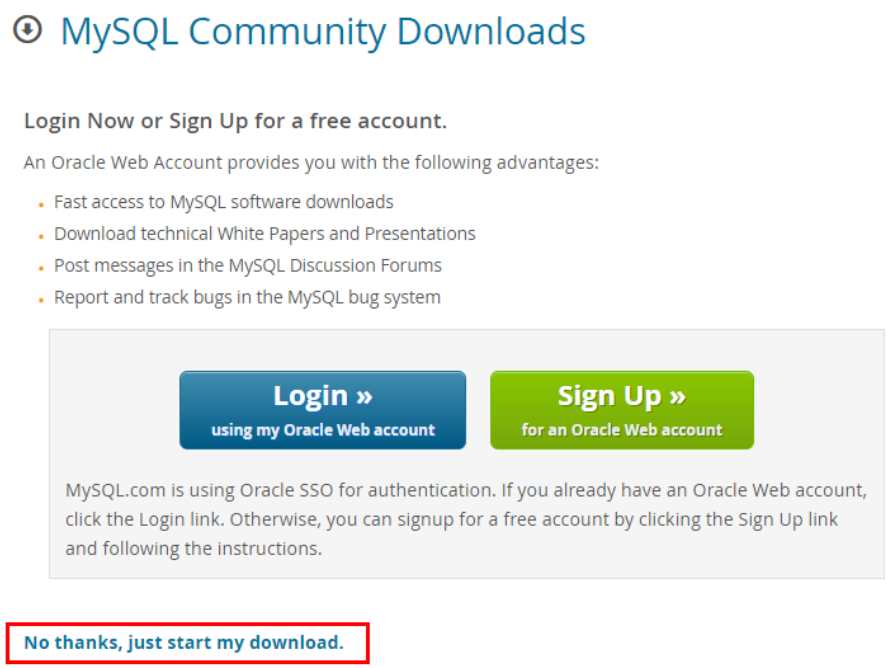
If the message "command not found" is displayed, install the Telnet tool based on the OS used by the ECS.

- 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.
  - If in the security group of the DB instance, there is no inbound rule allowing the access from the EIP and port of the ECS, add an inbound rule for the EIP and port of the ECS. For details, see **Configuring Security Group Rules**.

Step 5 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 6 Upload the installation package to the ECS.

Step 7 Use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

Step 8 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 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

----End

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

1. On the **Instances** page of the TaurusDB console, click the instance name to go to the **Basic Information** page.
2. In the **DB 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:
   ```

3.4 Connecting to a DB Instance Through 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.

1. **Purchasing an ECS**
2. **Binding an EIP to a DB Instance**
3. **Querying the EIP of the DB Instance to Be Connected**
4. **Testing Connectivity and Installing MySQL-Front**
5. **Using MySQL-Front to Connect to the 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 over a Public Network**.
- 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 create a Windows ECS, see section "Creating 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 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 instance if needed.

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

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 **Database** > **TaurusDB**.

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 project.

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

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*

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.
  - If in the security group of the DB instance, there is no inbound rule allowing the access from the EIP and port of the ECS, add an inbound rule for the EIP and port of the ECS. For details, see **Binding an EIP**.

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

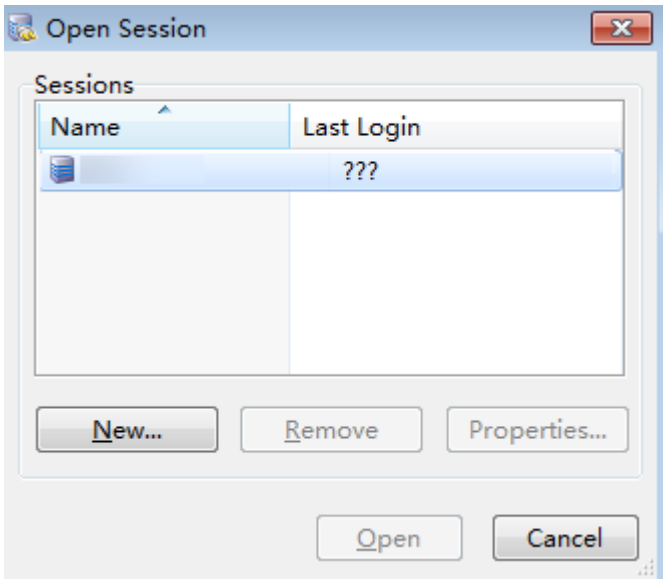
----End

Using MySQL-Front to Connect to the DB Instance

Step 1 Start MySQL-Front.

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

Figure 3-3 Connection management



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

Figure 3-4 Adding an account

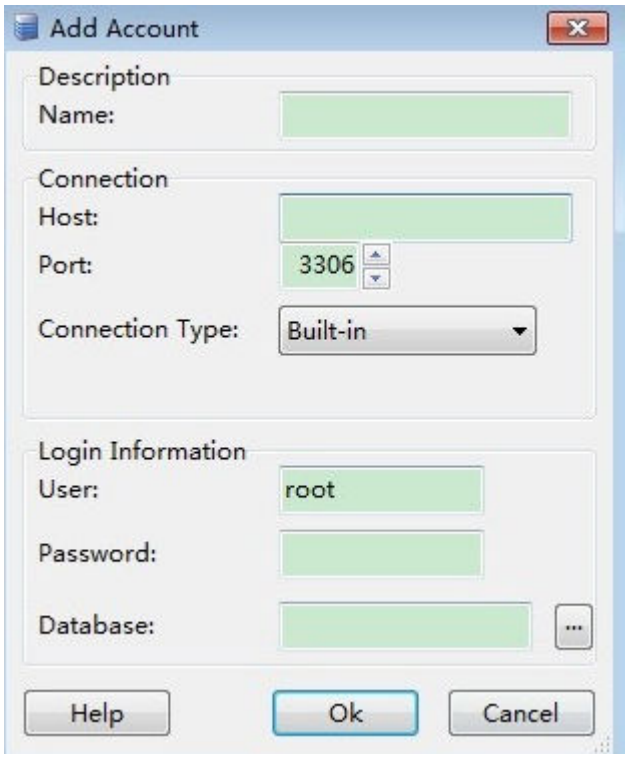
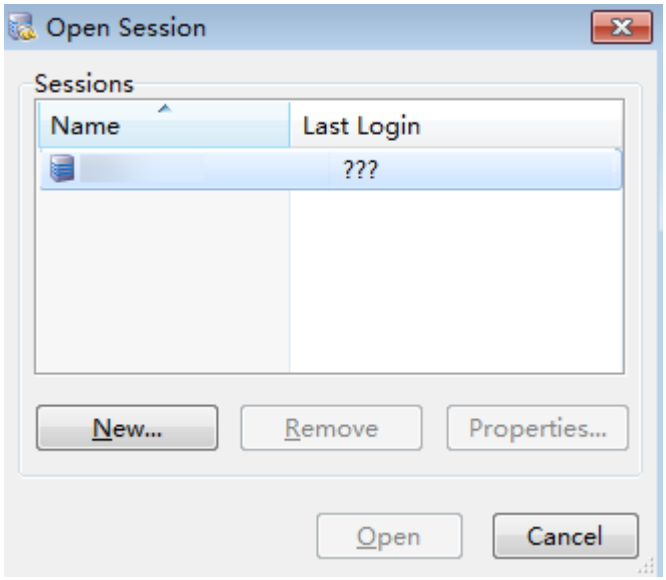


Table 3-4 Parameter description

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	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 3-5 Opening a session



----End

3.5 Connecting to a DB Instance Through 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 TaurusDB 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 3-5 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 the TaurusDB 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 an 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 EIP in the **Network Information** area on the **Basic Information** page.
<instance_port>	Database port of the 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**.
<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	verifyServerCertificate	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 TaurusDB 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 the TaurusDB 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 an 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 EIP in the **Network Information** area on the **Basic Information** page.
<instance_port>	Database port of the instance. The default port is **3306**. **NOTE** You can view the database port 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 TaurusDB 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");
                    //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 TaurusDB 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 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.

This section describes how to configure an inbound rule for a DB 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 project.

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

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 **Security Group**.

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.

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)

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

3.6.2 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 TaurusDB 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 project.

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

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

3.6.3 Changing a Database Port

Scenarios

You can change the database port of a DB instance. The change will be applied to the ports of the primary node and read replicas.

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 **Database** > **TaurusDB**.

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  in the **Database Port** field.

The database port of a TaurusDB 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.

- To submit the change, click ✓.
  - In the displayed dialog box, click **Yes**.
    i. 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.
    ii. This process takes about 1–5 minutes.
- To cancel the change, click ✗.

Step 6 View the results on the **Basic Information** page.

----End

3.6.4 Configuring and Changing a Private IP Address

Scenarios

You can plan and change private IP addresses after migrating on-premises databases or other cloud databases to TaurusDB.

Constraints

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. Therefore, 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 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 **Database** > **TaurusDB**.

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  next to the **Private IP Address** field.

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 instance.

----End

4 Database Usage

4.1 Usage Rules

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 Community Edition 8.0, see **Keywords and Reserved Words**.

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

  Table 4-1 lists the new keywords and reserved words in TaurusDB.

Table 4-1 New keywords and reserved words in TaurusDB

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 latency.
- 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**.

> **NOTE**
>
> 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, for example, **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;**

This pagination method is recommended. The boundary value from the last record of the previous page is transferred.

- 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 your TaurusDB 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.

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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Databases**. On the displayed page, click **Create Database**. In the displayed dialog box, enter a database name, select a character set, and authorize permissions for users. Then, click **OK**.

Table 4-2 Parameter description

Parameter	Description
Database Name	The database name can consist of up 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**.
Remarks	The remarks can consist of up to 512 characters. It cannot contain carriage returns or any of the following special characters: !<"='>&

Step 6 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

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 **Database** > **TaurusDB**.

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.

You can use either of the following methods to create a database:

1. On the home page, click **Create Database**. In the displayed dialog box, set the database name, character set, and collation, and click **OK**.
2. Choose **SQL Operations** > **SQL Query**. In the displayed SQL window, select the target database and run the following command:
   ```
   create database database_name;
   ```

----End

4.2.2 Deleting a Database

Scenarios

You can delete databases you have created.

Constraints

- Deleted databases cannot be recovered. Exercise caution when performing this operation.
- 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, choose **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Databases**. On the displayed page, locate a database and click **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, confirm the database information and enter **DELETE** as prompted.

Step 6 Click **OK**.

----End

4.3 Account Management (Non-Administrator)

4.3.1 Creating a Database Account

Scenarios

When you create a TaurusDB instance, account **root** is created at the same time by default. You can create other database accounts as needed.

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, choose **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Accounts**. On the displayed page, click **Create Account**. In the displayed dialog box, enter a username, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

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, enter **%** for **Host IP Address**. • To enable all IP addresses in the subnet 10.10.10.*to access your DB instance, enter **10.10.10.%** for **Host IP Address**. • To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.*,172.16.213.*** (no spaces before or after the comma).
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 delete a database somewhere other than on the TaurusDB console, 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**.
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 parameters, click the instance name, choose **Parameters** in the navigation pane, and search for **validate_password** in the upper right corner of the page. • 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 6 After the account is created, manage it on the **Accounts** page.

----End

Creating a Database 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, choose **Database** > **TaurusDB**.

Step 4 On the displayed login page, enter the username and password and click **Log In**.

Step 5 Choose **SQL Operations** > **SQL Query** and enter the following command:

```
create user Account name;
```

----End

4.3.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect your 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.

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 **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Accounts**. On the displayed page, locate the target account and click **Reset Password** in the **Operation** column.

Step 5 In the displayed dialog box, enter a new password and confirm it.

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 parameters, click the instance name, choose **Parameters** in the navigation pane, and search for **validate_password** in the upper right corner of the page.

- 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 6 Click **OK**.

----End

4.3.3 Changing Permissions for Database 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, choose **Database** > **TaurusDB**.

Step 4 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the account and choose **More** > **Change Permission** in the **Operation** column.

Step 5 In the displayed dialog box, 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.

----End

4.3.4 Deleting a Database Account

Scenarios

You can delete database accounts you have created.

NOTICE

Deleted database accounts cannot be restored. Exercise caution when deleting an account.

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, choose **Database** > **TaurusDB**.

Step 4 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the database that you want to delete and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

----End

5 Data Migration

5.1 Migrating Data to TaurusDB Using mysqldump

NOTICE

Mysqlpump is not recommended because it can result in a core dump in parallel backup scenarios. Mysqldump is recommended instead.

Preparing for Data Migration

You can access a TaurusDB instance through a private network or a public network.

1. Prepare an ECS in the same VPC and subnet as the TaurusDB instance or bind an EIP to the TaurusDB instance.
   - To connect to an instance through a private network, an ECS has to be created first.

     For details on how to create and log in to an ECS, see descriptions about creating an ECS and logging in an ECS in the *Elastic Cloud Server User Guide*.
   - To connect to an instance through an EIP, you must:
     i. Bind the EIP to the instance.
     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.

   NOTE

   The MySQL client version must be the same as or later than that installed on the TaurusDB instance. The MySQL database or client provides the mysqldump and mysql tools by default.

Exporting Data

Before migrating data to TaurusDB, data needs to be exported first.

NOTICE

- The export tool must match the DB engine version.
- Database migration is performed offline. Before the migration, you must stop any applications using the source database.

Step 1 Log in to the prepared ECS or device that can access the TaurusDB instance.

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

NOTICE

MySQL databases are required for TaurusDB 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.xx.xx.xx -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 the data into an SQL file.

NOTICE

MySQL databases are required for TaurusDB 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 TaurusDB instance through an ECS or a device 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 TaurusDB 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 TaurusDB instance.
- ***DB_PORT*** indicates the port of the TaurusDB 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 TaurusDB 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 TaurusDB instance.
- ***DB_PORT*** indicates the port of the TaurusDB 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 instance and view the result.

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

6 Instance Management

6.1 Instance Lifecycle Management

6.1.1 Changing a DB Instance Name

Scenarios

You can change the name of a TaurusDB 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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate an instance and click in the **Name/ID** column to edit the instance name.

Alternatively, click the instance name to go to the **Basic Information** page. In the **DB Instance Name** field in the **DB Instance Information** area, click to edit the instance 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.
- 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 View that the instance name has been changed. It takes less than 1 minute to change a DB instance name.

----End

6.1.2 Changing a DB Instance Description

Scenarios

After a TaurusDB 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, choose **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate an instance and click in the **Description** column to edit the instance description.

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

Alternatively, click the instance name to go to the **Basic Information** page. In the **DB Instance Information** area, click in the **Description** field to edit the instance description.

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

NOTE

The instance description can contain up to 64 characters, and cannot start with and end with a space. Only letters, digits, hyphens (-), underscores (_), periods (.), and spaces are allowed.

Step 5 View the results on the **Basic Information** page.

Alternatively, view the results on the **Instances** page.

----End

6.1.3 Deleting a DB Instance

Scenarios

You can manually delete a DB instance on the TaurusDB **Instances** page.

Constraints

- Instances cannot be deleted when operations are being performed on them.
- If you delete a DB instance, the read replicas associated with it are also deleted.

- You can rebuild deleted instances from the recycle bin. For details, see **Rebuilding a Deleted Instance from Recycle Bin**.

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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the instance you want to delete and click **More** > **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**. Refresh the **Instances** page later to check that the deletion is successful.

----End

6.1.4 Rebooting a DB Instance

Scenarios

You may need to reboot a DB instance for maintenance reasons. For example, after changing some parameters, you must reboot the instance for the modifications to take effect.

Constraints

- 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. During this period, the instance status is **Rebooting**.
- Rebooting DB instances will cause instance unavailability. To prevent traffic congestion during peak hours, reboot instances 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 **Database** > **TaurusDB**.

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

Step 6 View the task execution progress on the **Task Center** page. If its status is **Available**, it has been rebooted.

----End

6.1.5 Changing a Node Name

Scenarios

You can change a node name for easy identification.

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 **Database** > **TaurusDB**.

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

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

- Click **OK** to save the modifications.
- Click **Cancel** to cancel the modifications.

You can also click next to a node name, enter the new node name, and click **OK**.

NOTE

- 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 View that the node names have been changed.

----End

6.1.6 Exporting Instance Information

Scenarios

You can export information about all instances for review and analysis.

Exporting Information About All Instances

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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, click  in the upper right corner of the page. In the displayed dialog box, select the items to be exported and click **Export**.

Step 5 After the export task is complete, a .csv file is generated locally.

----End

6.1.7 Rebuilding a Deleted Instance from Recycle Bin

You can rebuild unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances in the recycle bin.

Modifying the Recycling Policy

NOTICE

The new recycling policy takes effect only for instances that are put into the recycle bin after the modification. For instances that already exist in the recycle bin before the modification, the original recycling policy takes effect.

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 **Database** > **TaurusDB**.

Step 4 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances (value range: 1 to 7 days).

Step 5 Click **OK**.

----End

Rebuilding a DB Instance

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

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 **Database** > **TaurusDB**.

Step 4 On the **Recycle Bin** page, locate the instance you want to rebuild and click **Rebuild** in the **Operation** column.

Step 5 On the **Rebuild DB Instance** page, configure required information and submit the task. For details, see **Restoring Data to a DB Instance**.

----End

6.2 Instance Modifications

6.2.1 Changing vCPUs and Memory of a DB Instance

Scenarios

You can change the vCPUs and memory of a DB instance if needed. If the status of a DB instance changes from **Changing instance specifications** to **Available**, the change was successful.

Constraints

- A DB instance cannot be deleted when its specifications are being changed.
- The vCPUs and memory can be changed only at the instance level. It means that the specifications of the primary node or read replicas cannot be changed separately for a given instance.
- Changing instance specifications will cause a primary/standby switchover. To prevent service interruptions, change the instance specifications 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.

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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the instance for which you want to change specifications and choose **More** > **Change Instance Specifications** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **DB Instance Information** area, click **Change** next to **Instance Specifications**.

Step 5 On the displayed page, select new specifications as required and the scheduled time, and click **Next**.

Choose either of the following scheduled time:

- **Upon submission**: The instance specifications will be changed immediately after the task is submitted.
- **In maintenance window**: The instance specifications will be changed during the maintenance window you specify.

Step 6 On the displayed page, confirm the instance 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 instance 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 instance 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.

----End

6.2.2 Configuring Auto Scaling Policies

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

- Configuring auto scaling requires the iam:agencies:listAgencies permission.
- 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 **Creating a Proxy Instance**.
- 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.

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, choose **Database** > **TaurusDB**.

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

Step 5 In the **DB Instance Information** area, click **Modify** next to **Auto Scaling**.

Step 6 In the displayed dialog box, configure the required parameters.

Table 6-1 Parameter configuration

Parameter	Parameter description
Auto Scale-up	You can enable or disable it as needed.
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	• When auto scale-up is enabled, the system periodically checks CPU usage. If the average CPU usage exceeds the preset limit within the observation period, the system 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.

Parameter	Parameter description
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, choose **Database** > **TaurusDB**.

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

Step 5 In the **DB Instance Information** area, click **View Change History** next to **Auto Scaling**.

Step 6 In the displayed dialog box, view the change time, change type, status, original specifications, and new specifications.

----End

6.2.3 Changing a Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default, but you can change it if needed. To prevent service interruption, set the maintenance window to 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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page. In the **DB Instance Information** area, click **Change** in the **Maintenance Window** field.

Step 5 In the displayed dialog box, select a maintenance window and click **OK**.

NOTE

Changing the maintenance window will not affect the timing that has already been scheduled.

----End

6.2.4 Selecting Instance Displayed Items

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, choose **Database** > **TaurusDB**.

Step 4 On the **Instances** page, click to edit items displayed in the instance list.

- The following items are displayed by default: instance name/ID, instance type, description, DB engine, status, enterprise project, billing mode, private IP address, and operation.

  These default displayed items cannot be hidden.
- You can also select other items, including the creation time, database port, and storage type.

----End

6.2.5 Upgrading a Minor Version

Scenarios

TaurusDB supports manual minor version upgrades, which can improve performance, add new functions, and fix bugs.

Precautions

- The upgrade will cause the instance to reboot and briefly interrupt services. To limit 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 the primary node and read replicas of a DB instance are deployed in the same AZ, a minor version upgrade will trigger a failover. If they are in different AZs, a minor 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 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 version upgrade cannot be rolled back after the upgrade is complete.
- DDL operations, such as creating events, dropping events, and altering events, are not allowed during a minor version upgrade.
- If the replication latency between the primary node and read replicas is longer than 300 seconds, the minor version cannot be upgraded.
- To upgrade the kernel version to 2.0.54.240600 or later, ensure that the value of **rds_global_sql_log_bin** is **ON** and the value of **binlog_expire_logs_seconds** is at least **86400**. For details about how to set these parameters, see **Modifying Parameters of a TaurusDB 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 **Database** > **TaurusDB**.

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

Step 5 In the **DB Instance Information** area, click **Upgrade** next to **DB Engine Version**.

Alternatively, go to the **Instances** page and click **Upgrade** in the **DB Engine Version** column.

Step 6 In the displayed dialog box, select a scheduled time and click **OK**.

----End

6.2.6 Enabling or Disabling Event Scheduler

You can enable or disable event scheduler on the TaurusDB console. Read **Disclaimer** carefully before using it.

Disclaimer

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, choose **Database** > **TaurusDB**.

Step 4 In the **DB Instance Information** area, click  next to **Event Scheduler**.

Step 5 In the displayed dialog box, read the disclaimer and click **Agree and Continue**.

Step 6 In the displayed dialog box, confirm the instance information and click **OK**.

----End

Disabling Event Scheduler

Step 1 In the **DB Instance Information** area, click  next to **Event Scheduler**.

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

----End

7 Billing Management

7.1 Renewing a DB Instance

Scenarios

A yearly/monthly instance can be renewed based on service requirements.

Constraints

- Only yearly/monthly instances can be renewed.
- The statuses of yearly/monthly instances to be renewed must be **Available** or **Abnormal**.

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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the instance you want to renew and click **Renew** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Renew** on the right of **Billing Mode**.

Step 5 On the displayed page, renew the instance.

----End

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

Scenarios

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

NOTICE

The pay-per-use billing mode is applied only after a yearly/monthly subscription expires and auto-renew is not in effect.

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

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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the yearly/monthly instance that you want to change to pay-per-use instance and choose **More** > **Change to Pay-per-use** in the **Operation** column.

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

Step 5 On the displayed page, change the billing mode of the instance.

Step 6 On the **Change to Pay-per-Use After Expiration** page, confirm the instance billing information and click **Change to Pay-per-Use**.

Step 7 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the instance list, click  to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Pay-per-use**.

----End

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

Scenarios

If you want to use TaurusDB for a long time, you can change the billing mode of one instances from pay-per-use to yearly/monthly for a lower tariff. After the change, you can check whether the operation has taken effect in the order status.

Constraints

- The billing of the primary node and read replicas for a pay-per-use instance cannot be changed separately to yearly/monthly.
- Pay-per-use instances in any of the following statuses cannot be changed to yearly/monthly instances: frozen, creation failed, changing instance specifications, scaling up, and creating read replicas.

Changing the Billing Mode 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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the pay-per-use instance that you want to change to the yearly/monthly instance and choose **More** > **Change to Yearly/Monthly** in the **Operation** column. On the displayed page, select your desired storage space and click **Next**.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Billing Information** area, click **Change** next to the **Billing Mode** field. On the displayed page, select your desired storage space and click **Next**.

Figure 7-1 Changing the billing mode from pay-per-use to yearly/monthly

Storage space contains the system overhead required for inodes, reserved blocks, and database operation. The storage space must be a multiple of 10.

Step 5 Select how many months you want to renew the subscription for. The minimum duration is one month.

- If you do not need to modify your settings, click **Pay** to go to the payment page.
- If you are not sure about the settings, the system will reserve your order. You can choose **Billing Center** > **Unpaid Orders** in the upper right corner and pay

or cancel the order. In addition, the instance status is **Changing to Yearly/Monthly. Payment incomplete. Pay Now.**

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

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

In the upper right corner of the instance list, click  to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Yearly/Monthly**.

----End

7.4 Unsubscribing a Yearly/Monthly Instance

Scenarios

To unsubscribe an instance billed on a yearly/monthly basis, you need to unsubscribe the order first. After you unsubscribe the instance order, all resources including read replicas of the instance are also deleted.

Procedure

Unsubscribe a yearly/monthly instance on the **Instances** page.

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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the instance you want to unsubscribe and choose **More** > **Unsubscribe** in the **Operation** column. In the displayed dialog box, click **Yes**.

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

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

NOTICE

1. Unsubscribed DB instances will be moved to the recycle bin, but will be permanently deleted after a length of time determined by the recycling policy. Automated backups are deleted, but manual backups are retained and still billed. To delete the manual backups, go to the **Backups** page on the console.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription result. After the instance order is successfully unsubscribed, the instance will be deleted.

----End

8 Data Backups

8.1 Backup Principles

TaurusDB supports automated and manual backups. You can periodically back up databases. If a database is faulty or data is damaged, you can restore the database using backups to ensure data reliability.

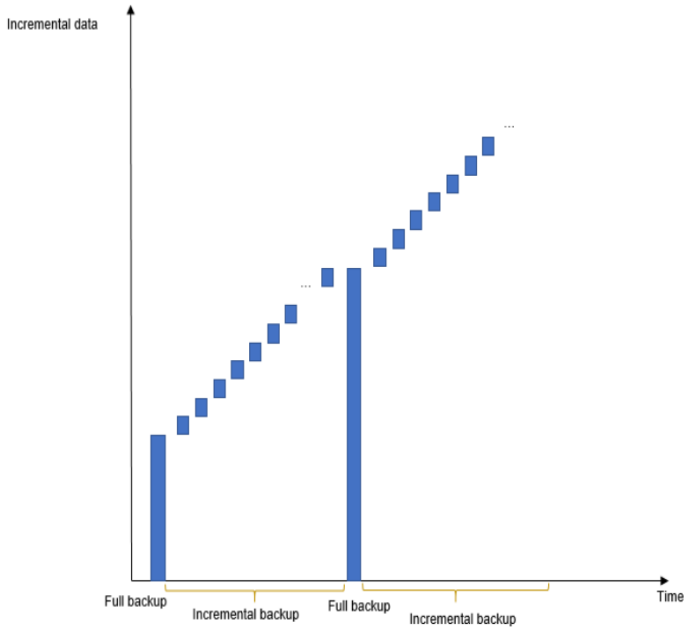
- **Automated backup**

  You can click **Configure Some-Region 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 specify in the backup policy and will store the data for as long as you have configured the retention period for.

  - Automated backups cannot be manually deleted. To delete them, you adjust the retention period specified in your **same-region backup policy**. Retained backups (including full and incremental backups) will be automatically deleted at the end of the retention period.
  - A full backup means that all data in your DB instance is backed up. In an incremental backup, only data that has changed within a certain period is backed up.

    Incremental backups are created based on the most recent full backup, as shown in **Figure 8-1**, so the most recent full backup that exceeds the retention period is still retained. For details, see the following example.

Figure 8-1 Backup restoration



- **Manual backup**

  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.

Backup Principles

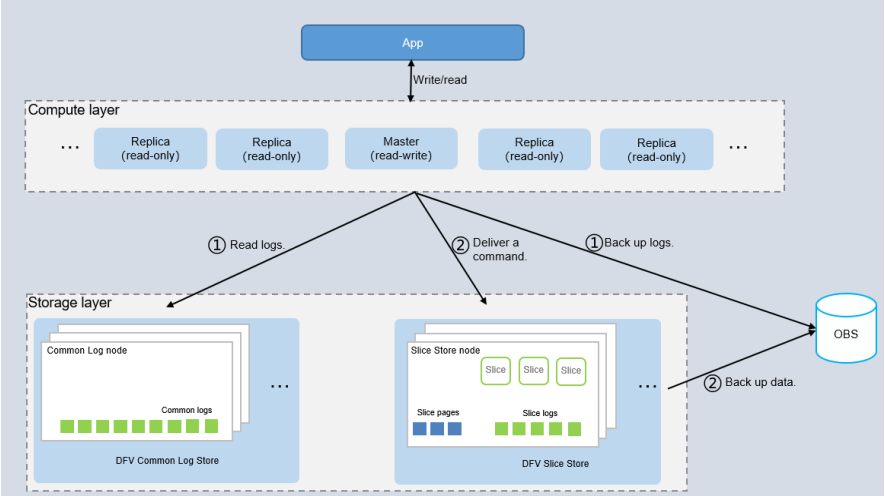
TaurusDB uses DFV storage, which decouples storage from compute. The compute layer provides services for external systems and manages logs, and the storage layer stores data. The storage layer consists of Common Log nodes and Slice Store nodes.

As shown in **Figure 8-2**, the creation of backups involves in the compute layer and storage layer.

- The primary node at the compute layer reads the log content of the Common Log node at the storage layer and backs it up to OBS.
- The primary node at the compute layer sends a command for backing up data to the Slice Store node at the storage layer. The Slice Store node backs 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.

Figure 8-2 Backup principles



8.2 Backup Types

TaurusDB supports multiple backup types. Based on different dimensions, there are the following backup types:

- Full backups and incremental backups based on data volume

Table 8-1 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.
Characteristic	• 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.	• The system automatically backs up data modifications made after the most recent automated or incremental backup every 5 minutes or when a certain amount of incremental data is generated. • Incremental backups are automated backups.

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 8-2 Comparison between automated backups and manual backups

Backup Type	Automated backups	Manual backups
Enabled by Default	Yes	Yes
Retention Period	1 to 3,660 days	Manual backups are always retained until you delete them manually.
Characteristic	TaurusDB saves automated backups based on the retention period you specified.	Manual backups are user-initiated full backups of instances. They are retained until you delete them manually.
Configuration	**Configuring a Same-Region Backup Policy**	**Creating a Manual Backup**

8.3 Configuring a Same-Region Backup Policy

Scenarios

When you create a TaurusDB instance, an automated backup policy is enabled by default and cannot be disabled. However, it can be modified after instance creation is complete. TaurusDB backs up data based on the automated backup policy you specify.

TaurusDB backs up data at the 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

- 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, TaurusDB 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.

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, choose **Database** > **TaurusDB**.

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

Step 5 Choose **Backups** in the left navigation pane, click **Modify Backup Policy**. You can view the configured backup policy. To modify the backup policy, adjust the parameter values as needed.

Table 8-3 Parameter description

Parameter	Description
Retention Period	Number of days that your automated backups can be retained. The retention period is from 1 to 3,660 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.
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 6 Click **OK**.

----End

8.4 Creating a Manual Backup

Scenarios

TaurusDB allows you to create manual backups for available DB instances. You can use these backups to restore data.

Constraints

- 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.

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 **Database** > **TaurusDB**.

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.

Alternatively, on the **Instances** page, click the instance name. Choose **Backups** in the navigation pane and click **Create Backup**.

Step 5 In the displayed dialog box, enter a backup name and description and click **OK**.

Table 8-4 Parameter description

Parameter	Description
Backup 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.
Description	The description can consist of up to 256 characters. It cannot contain carriage returns or any of the following special characters: !<''=>&

Step 6 Click **OK**.

View and manage the created backup on the **Backups** page.

----End

8.5 Exporting Backup Information

Scenarios

You can export backup information of a TaurusDB 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, backup location, status, size, and description.

Constraints

Automated and manual backup files 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, choose **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Backups**.

Step 5 Select the backups to be exported and click **Export**.

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**.

- 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

8.6 Deleting a Manual Backup

Scenarios

You can delete manual backups to release storage space.

NOTICE

Deleted manual backups 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 **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Backups**. On the displayed page, locate the manual backup to be deleted and click **Delete** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. 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 5 Click **Yes**.

----End

9 Data Restorations

9.1 Restoring a DB Instance

If data is damaged or mistakenly deleted, you can restore it from backups.

Table 9-1 Restoring data

Scenario	Description
Restoring Instance Data to a Specific Point in Time	You can restore instance data to a point in time. The data can be restored to a new DB instance, the original DB instances, and an existing DB instance.
Restoring Data to a DB Instance	You can restore data to a new DB instance, the original DB instance, or an existing DB instance using automated or manual backups.

9.2 Restoring Instance Data to a Specific Point in Time

Scenarios

You can restore data of an instance to a specified 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, choose **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Restore to Point in Time**.

Step 6 Select a time range, select or enter a time point within the acceptable range, and set **Restoration Method** to **Create New Instance** or **Restore to Original**.

1. **Create New Instance**: Click **OK**. On the **Create New Instance** page, configure parameters and click **Next**.
   - The region, DB engine and version of the new instance are the same as those of the original instance and cannot be changed.
   - The default database port is **3306**.
   - Retain the default values for other parameters. You can also set the parameters as required.
2. **Restore to Original**: Click **Next**. In the displayed dialog box, click **OK**.

   Data on the original instance will be overwritten and the original DB instance will be unavailable during the restoration.

Step 7 View the restoration results.

- **Create New Instance**: After the creation is complete, the instance status changes from **Creating** to **Available**. The new instance is independent from the original one and includes the data before the backup was created. If you want to offload read pressure from the primary node, create one or more read replicas for the new instance.

  A full backup is triggered after the new instance is created.
- **Restore to Original**: When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

9.3 Restoring Data to a DB Instance

Scenarios

You can use an automated or manual backup to restore a DB instance to the status when the backup was created. The restoration is at the instance level.

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 **Database** > **TaurusDB**.

Step 4 Select the backup to be restored in either of the ways:

In the navigation pane, choose **Backups**. On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name. Choose **Backups** in the navigation pane, select the backup to be restored, and click **Restore** in the **Operation** column.

Step 5 Select a restoration mode. Currently, the backup can be restored to a new DB Instance or the original DB Instance.

1. **Create New Instance**: Click **OK**. On the **Create New Instance** page, configure parameters and click **Next**.
   - The region, DB engine and version of the new instance are the same as those of the original 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 and can be modified.
2. **Restore to Original**: Click **Next**.

   Data on the original instance will be overwritten and the original DB instance will be unavailable during the restoration.

Step 6 View the restoration results.

- **Create New Instance**: After the creation is complete, the instance status changes from **Creating** to **Available**. The new instance is independent from the original one and includes the data before the backup was created. If you want to offload read pressure from the primary node, create one or more read replicas for the new instance.

  A full backup is triggered after the new instance is created.
- **Restore to Original**: When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

10 Read Replicas

10.1 Introducing Read Replicas

Scenarios

A TaurusDB instance contains read replicas in addition to a primary node.

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

- Specifications of read replicas are the same as those of the primary node.
- You do not need to maintain accounts and databases for 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 DB 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 Creating a Read Replica

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 create read replicas for it.

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 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

- You can create a maximum of 15 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, choose **Database** > **TaurusDB**.

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 read replicas can be added. • Yearly/Monthly DB instance: Yearly/Monthly and pay-per-use 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.
AZ	TaurusDB 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.
Instance Specifications	This parameter is only available for cluster 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	• You can create up to 15 read replicas for a **yearly/monthly** or **pay-per-use** DB instance. • You can contact customer service to create up to 7 read replicas for a **serverless** DB instance.

Step 6 For a yearly/monthly instance, click **Next** and select a payment mode.

Step 7 For a pay-per-use or serverless 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 replica information 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

10.3 Promoting a Read Replica to the Primary Node

A TaurusDB 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, choose **Database** > **TaurusDB**.

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.

Step 6 On 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

- 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 the services 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.

----End

Automatic Failover

TaurusDB uses a high availability active-active architecture that automatically fails over to a read replica automatically 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.

TaurusDB 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, TaurusDB attempts to promote another read replica by priority and repeats the process until a read replica is successfully promoted.

11 Database Proxy (Read/Write Splitting)

11.1 Introducing Read/Write Splitting

Read/write splitting enables read and write requests to be automatically routed through a proxy address. You can **create a proxy instance** after a TaurusDB instance is created. Thanks to the IP address of the proxy instance, write requests are automatically routed to the primary node and read requests are routed to read replicas and the primary node by user-defined weights.

Constraints

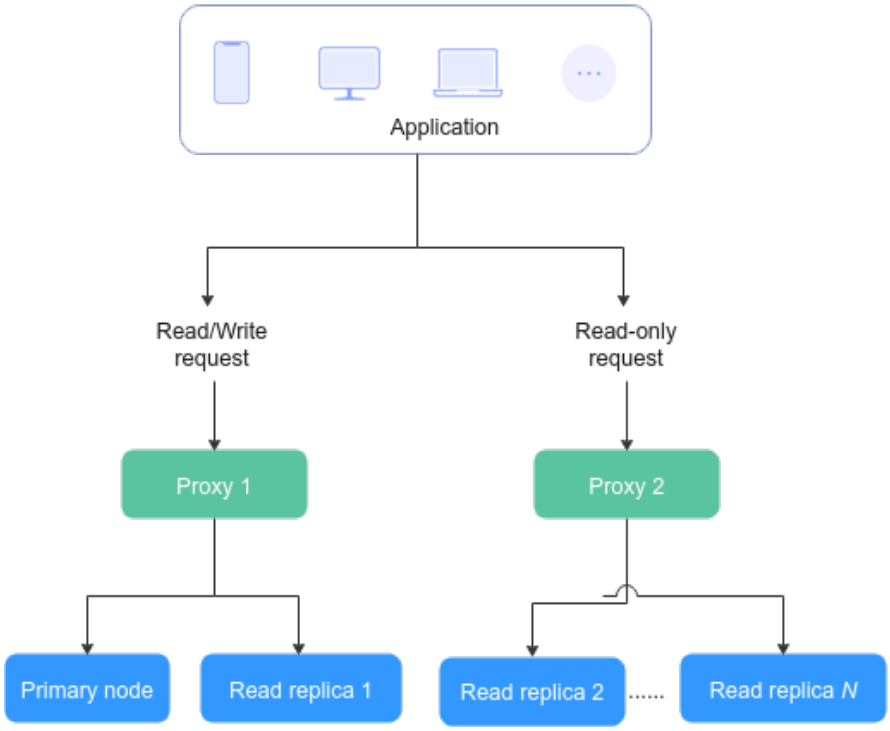
- If the kernel version of a TaurusDB instance is earlier than 2.0.42.230601, only one proxy instance can be created.
- If the kernel version of a TaurusDB instance is 2.0.42.230601 or later, a maximum of four proxy instances can be created.
- Read/write splitting can be enabled only when at least one read replica is created.
- After read/write splitting is enabled, the database port and private IP address of your TaurusDB instance cannot be changed.
- Read/write splitting does not support compression protocols.
- If multi-statements are executed, all subsequent requests will be routed to the primary node. To restore read/write splitting, you must disconnect your application from your instance and then connect it back again.
- When a proxy address is used, you can run **show processlist** command on the proxy instance or TaurusDB instance. If **show processlist** is executed on a proxy instance, only the services delivered through proxy nodes are displayed.
- When 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, workload on the deleted proxy node may be displayed when **show processlist** is executed on the proxy instance.

- When **kill** is executed, 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 result will be returned normally.
- Proxy instances do not support the transaction isolation level READ UNCOMMITTED.
- To create proxy instances, ensure that the data in a single column of a table cannot exceed 16 MB.

Scenarios

When enabling read/write splitting for a DB instance, you need to select the nodes (including the primary node and read replicas) to be associated with the proxy instances.

- Different applications can connect to the DB instance through the IP addresses of different proxy instances. Read requests are routed to the proxy instances that applications connect to. You can also associate nodes with or remove nodes from proxy instances.
- A primary node or read replica can be associated with multiple proxy instances at the same time.
- In the read/write mode, all write requests are routed to the primary node, and read requests are routed to each node based on the read weights or active connections.
- In the read-only mode, only read requests can be routed to read replicas based on the read weights and active connections.
- By default, proxy instances provide overload protection to prevent server OOM (out of memory) due to heavy pressure when users perform operations on large result sets. This function is enabled by default and does not need to be configured separately. If the pressure is caused by the database kernel, you need to configure a flow control policy.



11.2 Introducing Consistency Levels

GaussDB(for MySQL) provides two consistency levels to meet requirements in different scenarios.

- Eventual consistency (default)
- Session consistency

Constraints

- To configure consistency level, the kernel version of your GaussDB(for MySQL) instance must be 2.0.28.1 or later.
- To use session consistency, the kernel version of your proxy instance must be 2.7.4.0 or later.

Eventual Consistency

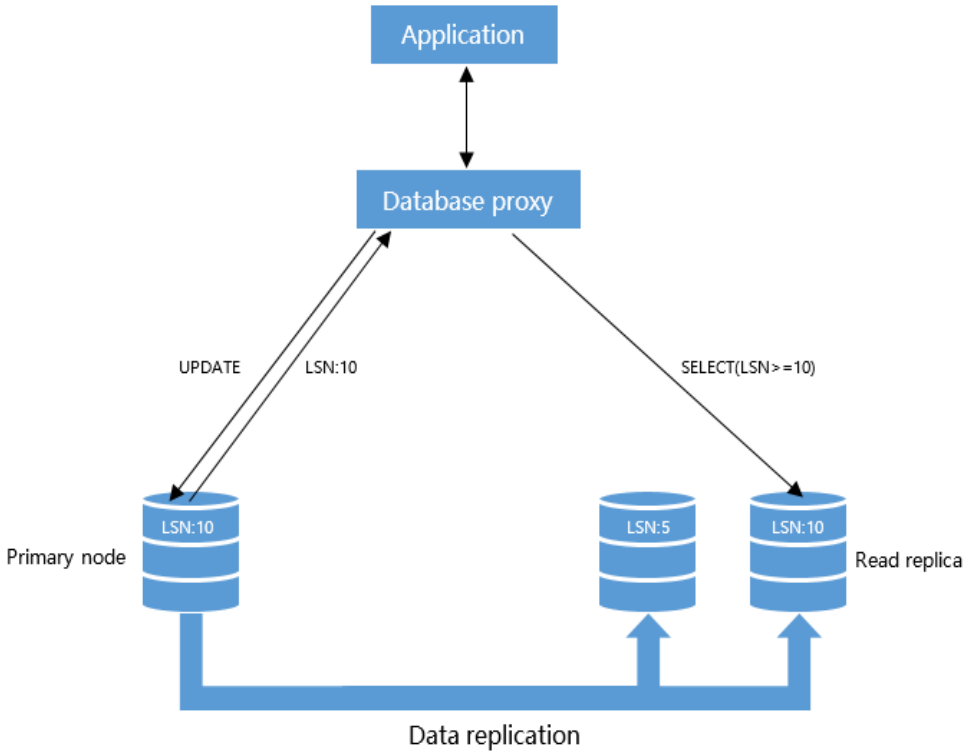
After a proxy instance is created, requests for SELECT operations are routed to different nodes based on their read weights. Because there is a replication delay between the primary node and each read replica and the replication delay varies for different read replicas, the result returned by each SELECT 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 database proxy 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 11-1 Principle of session consistency



NOTE

In session consistency, if there is significant replication delay 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.

11.3 Creating a Proxy Instance

A proxy instance enables read and write requests to be automatically routed through its IP address.

This section describes how to create a proxy instance.

Constraints

Proxy instances cannot be created if the TaurusDB kernel version is:

- From 2.0.26.2 to 2.0.28.3
- 2.0.29.1

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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click **Create Proxy Instance**.

Step 7 In the displayed dialog box, configure required parameters by referring to **Table 11-1** and click **OK**.

NOTE

After a proxy instance has been created, you can click **Create Proxy Instance** in the **Database Proxy** page to add a new proxy instance.

Table 11-1 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.
Proxy Mode	• **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. The default read weight of a node is **100**. • **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. **NOTE** – Only read requests are supported. If write requests are forwarded to the selected nodes, an error message is displayed. – This mode offloads the pressure of the primary node by routing all read requests to read replicas. – DDL, DML, and temporary table operations are not supported in the read-only mode.

Parameter	Description
Routing Policy	Values: • **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.
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
Proxy Instance Nodes	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 you enable this function, new read replicas are automatically associated with the proxy instance.
New Node Weight	If **Associate New Nodes** is enabled and **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.
Database Nodes	Select the nodes that need to be associated with the database proxy to process read requests.

----End

11.4 Configuring Connection Pools

Scenarios

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.

Constraints

- The kernel version of proxy instances must be 2.22.07.000 or later.

How a Session-Level Connection Pool Works

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.

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 **Database** > **TaurusDB**.

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

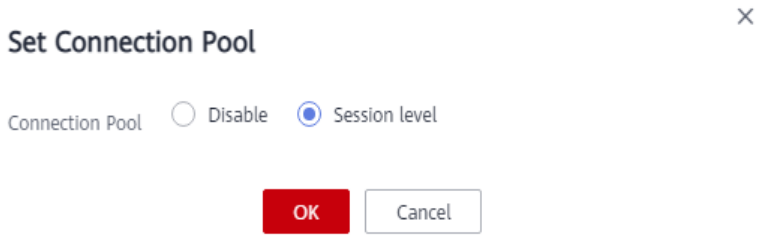
Step 5 In the navigation pane on the left, choose **Database Proxy**.

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 11-2 Configuring a connection pool



----End

11.5 Configuring Transaction Splitting

Scenarios

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 loads on the primary node.

Constraints

- The kernel version of the proxy instances must be 2.3.9.5 or later.

- Transaction splitting is only available for instances whose translation isolation level is READ UNCOMMITTED or READ COMMITTED. The default isolation level is REPEATABLE READ.
- To enable transaction splitting, the proxy mode must be set to read/write.
- After transaction splitting is enabled, read requests of the transactions submitted using **BEGIN** cannot be routed to read replicas.
- After transaction splitting is enabled, read requests of the transactions started using **SET AUTOCOMMIT = 0** cannot be routed to read replicas once the transactions are committed.

Function

Proxy instances support transaction splitting. With transaction splitting is enabled, GaussDB(for MySQL) can route the read requests prior to write operations in a transaction to read replicas, reducing the pressure on the primary node.

Transaction splitting is disabled by default.

After transaction splitting is enabled and **autocommit** is set to **0**, TaurusDB starts a transaction only for write requests. Before the transaction starts, read requests are routed to read replicas through load balancers.

Precautions

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 the function.

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 **Database** > **TaurusDB**.

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**.

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click  next to **Transaction Splitting**.

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

----End

NOTE

Transaction splitting takes effect only for new connections established after this function is enabled or disabled.

11.6 Configuring a Routing Policy

Proxy instances support weighted and load balancing routing policies.

To configure a routing policy, you need to:

- Create a proxy instance. For details, see **Creating a Proxy Instance**.
- Select a routing policy by referring to this section.

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 **Database** > **TaurusDB**.

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**.

Step 6 On the **Database Proxy** page, click the name of a proxy instance.

Step 7 On the **Basic Information** page, click **Change** next to **Routing Policy**.

Step 8 In the displayed dialog box, select a 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 load balancing policy, you do not need to configure the weights of nodes.

NOTE

The proxy mode of a proxy instance affects read requests assigned to different nodes.

- Read-only mode: All read requests are assigned to the selected read replica based on the routing policy and weights you specify, but not to the primary node.
- Read/write mode: All read requests are assigned to the selected nodes (including primary nodes and read replicas) based on the routing policy and weights you specify.

Figure 11-3 Changing the routing policy of a proxy instance in read/write mode

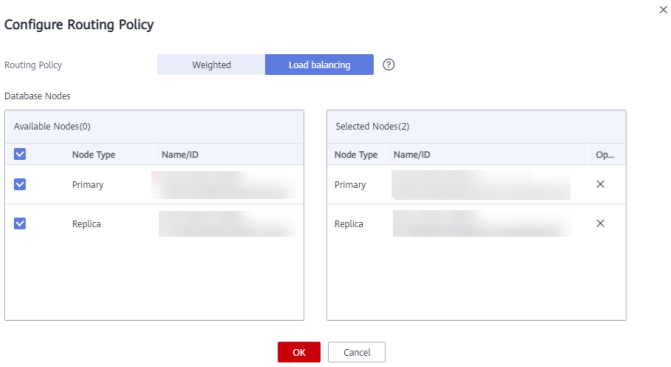
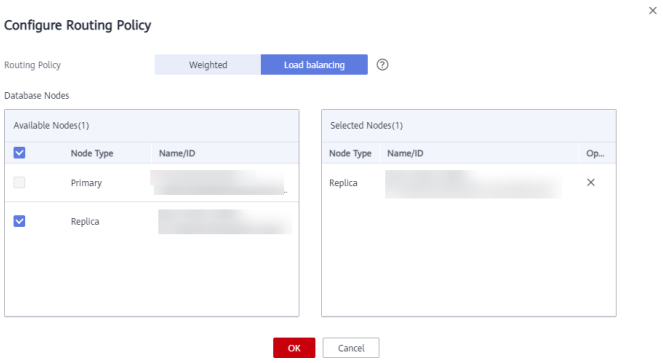


Figure 11-4 Changing the routing policy of a proxy instance in read-only mode



----End

11.7 Assigning Read Weights

After read/write splitting is enabled, you can assign read weights as required.

Description

- After read/write splitting is enabled, you can assign read weights for 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 **Database** > **TaurusDB**.

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

Step 5 On the **Database Proxy** page, click the name of a proxy instance.

Step 6 On the **Basic Information** page, click **Assign Weight**.

Step 7 In the displayed dialog box, select nodes for which you want to assign weights on the left and assign the weights on the right.

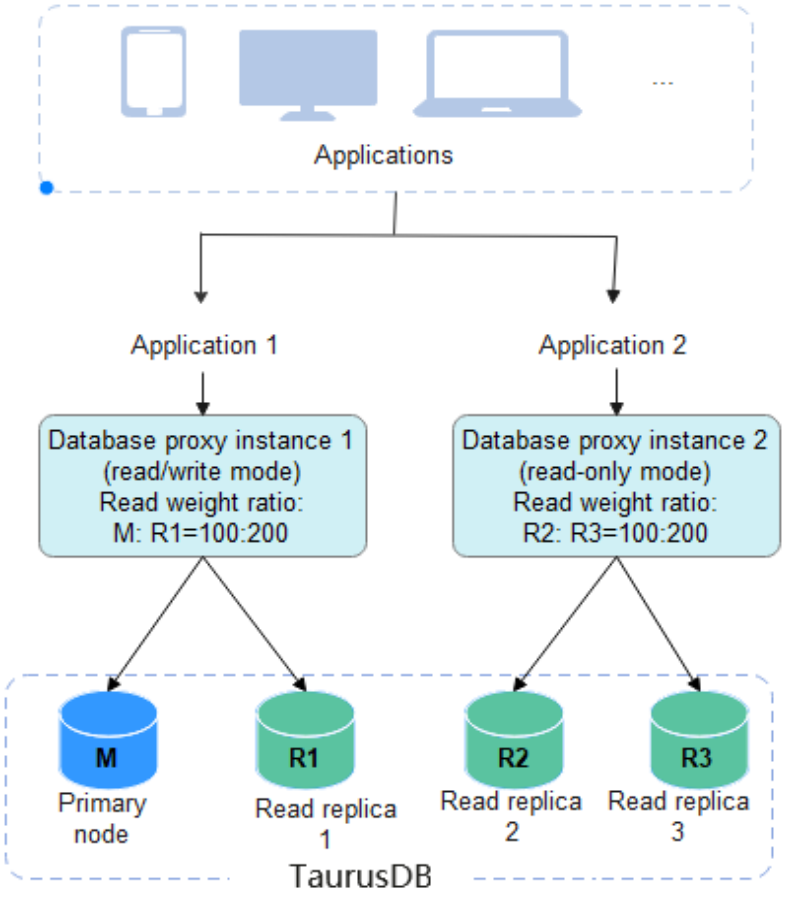
- Different applications can connect to the TaurusDB instance through different proxy addresses. Read and write requests are forwarded to associated nodes. You can also add nodes to or remove nodes from proxy instances.
- In the read/write mode, all write requests are routed to the primary node, and read requests are routed to each node based on the read weights.
- In the read-only mode, only read requests can be routed to read replicas based on the read weights.

Example:

As shown in **Figure 11-5**, one TaurusDB instance has one primary node and three read replicas, and two proxy instances have been created.

- 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 11-5 Read/Write splitting in multi-proxy scenarios (weighted routing policy)



----End

11.8 Changing the Specifications of a Proxy Instance

Constraints

- You can change the proxy instance specifications only when the statuses of the DB instance, primary node, read replicas, and proxy instance are **Available**.
- 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 in the upper left corner and select a region and project.

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

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**.

Step 6 On the **Database Proxy** page, locate the desired proxy instance and click **Change Specifications** in the **Operation** column.

You can also click the proxy instance name. In the **Proxy Instance Information** area, click **Change** next to the **Specifications** field.

Step 7 In the displayed dialog box, select new specifications and click **OK**. You can reduce or expand the specifications as required.

Step 8 View the new specifications on the **Database Proxy** page.

----End

11.9 Changing the Number of Nodes for a Proxy Instance

Scenarios

You can change the number of proxy nodes as required.

Constraints

- Your TaurusDB instance must be available.
- If a proxy instance is abnormal, you can only add nodes to it but cannot reduce nodes.
- There can be 2 to 16 proxy nodes. To request more nodes, contact customer service.

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 **Database** > **TaurusDB**.

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**. Click the proxy instance name.

Step 6 In the **Proxy Instance Information** area, click **Change** next to the **Proxy Instance Nodes** field.

NOTE

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.

Step 7 In the displayed dialog box, set the number of proxy instance nodes and click **OK**.

----End

11.10 Upgrading the Kernel Version of a Proxy Instance

Scenarios

You can manually upgrade your database proxy instance to the latest kernel version to improve performance, add new functions, and fix problems.

Precautions

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 in the upper left corner and select a region and project.

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

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**.

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** in the **DB Engine Version** field.

Step 8 In the displayed dialog box, select a scheduled time and click **OK**.

----End

11.11 Using a Private Domain Name for a Proxy Instance

You can use a private network domain name to connect to a proxy instance.

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 **Database** > **TaurusDB**.

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

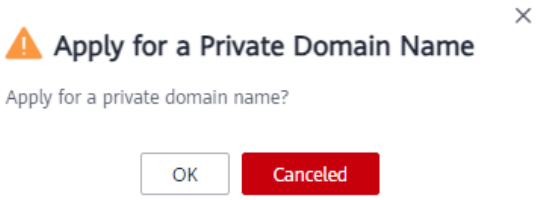
Step 5 In the navigation pane on the left, choose **Database Proxy**.

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.

Step 8 Click **OK**.

Figure 11-6 Applying for a private domain name (2)



Step 9 In the **Private Domain Name** field, view the generated private domain name.

----End

Changing 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 **Database** > **TaurusDB**.

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**.

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 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 **Database** > **TaurusDB**.

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**.

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

11.12 Changing the IP Address of a Proxy Instance

Scenarios

You can change the IP address of a proxy instance.

Precautions

Changing a proxy address will interrupt database connections and services. Perform the operation during off-peak hours or when services are stopped.

Constraints

- The new IP address is not in use and must be in the same subnet as the TaurusDB 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 **Database** > **TaurusDB**.

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, click **Database Proxy**.

Step 6 Click the desired proxy instance name. In the **Proxy Instance Information** area, click **Change** next to the **Proxy Address** field.

Step 7 In the displayed dialog box, enter a new IP address and click **OK**.

In-use IP addresses cannot be used.

----End

11.13 Changing the Port of a Proxy Instance

Scenarios

After a proxy instance is created, you can change its port as needed.

Precautions

- Changing a proxy port will interrupt the database connection. You are advised to change the port number 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 **Database** > **TaurusDB**.

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.

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, 33062, and 33071, which are reserved by the system)

- To submit the change, click .
  - In the displayed dialog box, click **Yes** to confirm the change.
  - In the displayed dialog box, click **No** to cancel the change.
- To cancel the change, click .

----End

11.14 Changing Consistency Level

Scenarios

After a proxy instance is created, you can change its consistency level.

Constraints

- To configure consistency level, the kernel version of your GaussDB(for MySQL) instance must be 2.0.28.1 or later.
- To use session consistency, the kernel version of your proxy instance 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, choose **Database** > **TaurusDB**.

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, click **Database Proxy**.

Step 6 Click the desired proxy instance name. In the **Proxy Instance Information** area, click next to the **Consistency Level** field.

Step 7 Select a consistency level and click .

NOTICE

After the consistency level is changed, you need to manually reboot the proxy instance or reconnect your application to the proxy instance on the management console.

For details about how to reboot a proxy instance, see **Rebooting a Proxy Instance**.

----End

11.15 Modifying Proxy Instance Parameters

Scenarios

You can change parameter for your database proxy instances.

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 **Database** > **TaurusDB**.

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 Modification**. 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

11.16 Enabling or Disabling Automatic Association of New Nodes with a Proxy Instance

After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the 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 **Creating a Proxy Instance**.

Enabling Automatic Association of New Nodes with 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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Database Proxy**. Then 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**.

Step 7 In the displayed dialog box, enable **Associate New Nodes**.

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. You can modify the default setting as required.

Step 8 Click **OK**.

----End

Disabling Automatic Association of New Nodes with a Proxy Instance

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **Database Proxy**. Then click the name of a proxy instance to go to the **Basic Information** page.

Step 3 In the **Proxy Instance Information** area, click  next to **Associate New Nodes**.

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

----End

11.17 Enabling or Disabling Access Control

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.

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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Database Proxy**. Then 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.

- **Access Control**: The blocklist and allowlist cannot be configured at the same time. If you switch between lists, your previously entered settings will be lost. IP addresses or CIDR blocks in the blocklist are not allowed to access the proxy instance.
- **IP Address or CIDR Block**: Enter valid 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|TaurusDB01**. 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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Database Proxy**. Then click the name of a proxy instance to go to the **Basic Information** page.

Step 6 Click  next to **Access Control**.

Step 7 In the displayed dialog box, click **Yes** to disable access control.

----End

11.18 Binding an EIP to or Unbinding an EIP from 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.

Binding an EIP to 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 **Database** > **TaurusDB**.

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 name to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click **Bind** next to **Public IP Address (EIP)**.

Step 8 In the displayed dialog box, select an EIP and click **OK**.

Step 9 On the **Basic Information** page, check that the EIP has been bound to the proxy instance.

----End

Unbinding an EIP from a Proxy Instance

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **Database Proxy**.

Step 3 Click a proxy instance name to go to the **Basic Information** page.

Step 4 In the **Proxy Instance Information** area, click **Unbind** next to **Public IP Address (EIP)**.

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

Step 6 On the **Basic Information** page, check that the EIP has been unbound from the proxy instance.

----End

11.19 Rebooting a Proxy Instance

Scenarios

You can reboot a proxy instance as needed.

Constraints

- You have obtained the required permissions from customer service.
- If the proxy 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.

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 **Database** > **TaurusDB**.

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**, locate the target proxy instance, and choose **More** > **Reboot** in the **Operation** column.

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

NOTE

Reboot a proxy instance interrupts the database connection. You are advised to reboot it during off-peak hours.

----End

11.20 Deleting a Proxy Instance

You can delete a proxy instance as required.

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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Database Proxy**.

----End

11.21 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.

Specifying Whether a SQL Statement Is Sent to a Primary Node or Read Replica By Adding a Hint

Hints supported by read/write splitting are as follows:

/*FORCE_MASTER*/: A SQL statement is executed on a primary node.

/*FORCE_SLAVE*/: A SQL statement is executed on read replicas.

NOTE

- 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 an instance using the MySQL CLI and Hints, add the **-c** option.

11.22 Testing Read/Write Splitting Performance

After a proxy instance is created, you can connect your TaurusDB instance through a proxy address. You can use internal SQL commands to verify the read/write splitting performance.

Procedure

Step 1 Log in to an ECS. For details, see **Elastic Cloud Server User Guide**.

Step 2 Connect to a DB instance through a proxy address.

mysql -h <*hostIP*> **-P** <*port*> **-u** <*userName*> **-p** <*password*>

Table 11-2 Parameter description

Parameter	Description
<*hostIP*>	Proxy address.
<*port*>	Database port. By default, the value is **3306**.
<*userName*>	Username of the TaurusDB database administrator account. The default username is **root**.
<*password*>	Password

Step 3 Run the following command to view the instance that executes the SQL command:

Run **show last route;** to view the routing result of the previous SQL statement.

Figure 11-7 Query result

```
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.08 sec)

mysql> show last route;
+-----------------+
| LAST ROUTE      |
+-----------------+
| [illegible]     |
+-----------------+
1 row in set (0.05 sec)

mysql>
```

Do not use **show last route** for service code or multi-statement execution.

----End

12 DBA Assistant

12.1 Function Overview

DBA Assistant provides you with a range of database O&M functions, making it easy to diagnose database problems, locate faults, analyze and optimize database performance.

DBA Assistant consists of the following modules:

Dashboard

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. For details, see **Dashboard**.

Sessions

On the **Sessions** page, you can view current session statistics of your instance, identify abnormal sessions, and kill the sessions. For details, see **Sessions**.

Performance

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. For details, see **Performance**.

Slow Query Log

The **Slow Query Log** page displays slow queries within a specified time period. Slow query logs are collected by user, IP address, SQL template, and other keywords, sort statistics, and identify sources of slow SQL statements. For details, see **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. For details, see **SQL Insights** and **Concurrency Control**.

12.2 Dashboard

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, choose **Database** > **TaurusDB**.

Step 4 On the **Instances** page, click the DB 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.

----End

Health

In the **Health** area, you can view real-time health diagnosis results of your instance. By default, the diagnosis results of high vCPU utilization, memory bottleneck, high-frequency slow SQL, and lock wait are displayed.

For abnormal metrics, click **Diagnose** to view diagnosis details and suggestions. For details, see **Table 12-1**.

Table 12-1 Health diagnosis and suggestions

Health 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 within 5 minutes.

Health Item	Exception Trigger Condition
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 5 minutes.
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 for 5 consecutive minutes.
Lock wait	After you configure alarm rules on Cloud Eye, any of the following alarms is reported: • Too long row lock time • Too many InnoDB row locks • Too many row lock waits

NOTE

- For details about how to configure alarm rules, see **Creating Alarm Rules for a DB Instance**.
- For details about monitoring metrics, see **Supported 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 within 5 minutes.

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 within 5 minutes.

Key Performance Metrics

In the **Key Performance Metrics** area, the CPU usage & slow query logs, connections, memory utilization, and disk reads/writes in the last hour are displayed by default. The displayed values are real-time values.

12.3 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, choose **Database** > **TaurusDB**.

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.

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.

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.

----End

12.4 Performance

Creating Alarm Rules

You can create alarm rules for TaurusDB to customize the monitored objects and notification policies and stay aware of the TaurusDB instance statuses.

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 **Database** > **TaurusDB**.

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 your instance metric trends within the same time range on different days.

----End

12.5 Slow Query Logs

Scenarios

Slow Query Logs 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.

Constraints

- 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 **Subscribing to Intelligent O&M**.
- After **Collect Slow Query Logs** is enabled, SQL text content will be stored in OBS.

Subscribing to Intelligent O&M

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** > **TaurusDB**.

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 Logs** tab.

Step 7 Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.

Get 5 GB of storage for free after your instance has subscribed to Intelligent O&M.

Step 8 Select "I have read and understand the billing rules." and click **Subscribe**.

----End

Slow Query Log Storage

- Intelligent O&M subscribed:

  Click **Log Settings** in the upper right corner to set slow query log retention days.

  - **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.

- Intelligent O&M not subscribed:
  - **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

Viewing Slow Queries over Time

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** > **TaurusDB**.

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 Logs** tab.

Step 7 Select a time range, and view slow queries over time by instance or node.

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).

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.

----End

Viewing Top 5 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** > **TaurusDB**.

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 Logs** tab.

Step 7 Select an instance, node, and time range. In the **Top 5 Slow Query Logs** area, view the top 5 slow SQL statements sorted by user or client IP address.

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).

----End

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** > **TaurusDB**.

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 Logs** tab.

Step 7 Select an instance, node, and time range, and view the slow query log details. The details include the SQL statement, execution start time, database, client, user, execution duration, lock wait duration, and scanned and returned rows.

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).

- To export slow query log details to an OBS bucket, click **Export**. In the displayed dialog box, select an OBS bucket and click **OK**. Up to 100,000 records can be exported.

  If no OBS bucket is available, click **Create**. In the displayed dialog box, enter an OBS bucket name, and click **OK**.

  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, click **View Export List** to view export records. You can also download the details to your local PC for analysis.
- To add a concurrency control rule, click **Concurrency Control** in the **Operation** column. In the displayed dialog box, specify **SQL Type**, **Keyword**, and **Max. Concurrency**. For details, see **Concurrency Control**.

- To view the SQL diagnosis results, click **Diagnose** in the **Operation** column.

----End

Viewing Template Statistics

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** > **TaurusDB**.

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 Logs** tab.

Step 7 Select an instance, node, and time range, and view the template statistics.

- Click **View Sample** in the **Operation** to view the sample of the SQL template.
- Export slow query log details.
  a. Click **Export**. In the displayed dialog box, select an OBS bucket and click **OK** to export the log details to the OBS bucket. Up to 100,000 records can be exported.
  b. If no OBS bucket is available, click **Create**. In the displayed dialog box, enter an OBS bucket name, and click **OK**.

     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 **View Export List** to view export records. You can also download the details to your local PC for analysis.

----End

12.6 Top SQL

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, choose **Database** > **TaurusDB**.

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**.

Step 7 View execution durations of the top SQL statements in the last 1 hour, last 3 hours, last 6 hours, or a custom time period (spanning no more than one day).

- Click a point in time or drag to select a time period to view the SQL statistics of a 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 a SQL template and click **Details** to view the total execution times, average rows scanned, average execution duration, and the like.
- Locate a SQL template and click **Concurrency Control** in the **Operation** column. For details, see **Concurrency Control**.

----End

12.7 SQL Insights

Scenarios

The SQL Insights function 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 a 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 TaurusDB 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 **Database** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click **SQL Explorer** and then **SQL Insights**.

Step 7 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 8 Click **Create Task** and specify **Time Range**, **Dimension**, **Username**, **Keyword**, **Database**, **Thread ID**, **SQL Type**, and **Execution Status**.

You can set **Dimension** to **Instance** or **Node**. When **Node** is selected, you can view the SQL logs of deleted nodes.

Step 9 Click **OK**.

Step 10 In the task list, click **Details** in the **Operation** column to view task details.

Step 11 Specify conditions such as **Time Range**, **User**, **Keyword**, **Database** and click **Query**. The selected time range must be after the time when the new task is added.

----End

12.8 Concurrency Control

Scenarios

Concurrency Control keeps TaurusDB instances stable regardless of how many SQL statements are concurrently submitted.

Constraints

- This function is only available to TaurusDB 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 a 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, choose **Database** > **TaurusDB**.

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 **Concurrency Control**.

Step 7 On the displayed page, enable **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

12.9 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 **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, choose **Database** > **TaurusDB**.

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**.

Step 8 Click **Add Kill Task**.

Step 9 In the displayed dialog box, set the criteria for killing sessions.

NOTICE

- The parameters listed in **Table 12-2** 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 12-2 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.
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**.

Step 5 Click **Kill 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**.

Step 5 Click **View History**.

Step 6 In the displayed dialog box, select a time range to view killed sessions within that period.

A maximum of 500 session records can be displayed.

----End

12.10 Storage Analysis

Storage occupied by data and logs and changes of storage usage are important for database performance. On the **Storage Analysis** page, you can 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, choose **Database** > **TaurusDB**.

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.

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**.

Step 2 In the **Subscribe to Intelligent O&M** dialog box, confirm the information, select the agreement, and click **Subscribe**.

Step 3 In the **Tablespaces** area, view 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.

  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.

- 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.

  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.

- **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  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.

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.

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

12.11 Anomaly Diagnosis

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, choose **Database** > **TaurusDB**.

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**.

Step 7 Click  on the right of **Anomaly Collection** to enable 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

13 Parameter Template Management

13.1 Creating a Parameter Template

You can use database parameter templates to manage DB engine configurations. A database parameter template acts as a container for engine configuration values that can be applied to one or more instances.

NOTICE

Not all DB engine parameters can be changed in a custom parameter template.

If you want to use a custom parameter template, you just create a parameter template and select it when you create an instance or apply it to an existing instance. For details, see **Applying 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 template, you can replicate that parameter template. For details, see **Replicating a Parameter Template**.

The following are the key points you should know when using parameter templates:

- To change the parameters in a parameter template of the current instance, go to the **Parameters** page, change parameter values and save the changes. Dynamic parameter changes take effect immediately, but static parameter changes take effect only after you manually reboot the instance. This changes will apply only to the current instance. They will not affect other instances.
- To change the parameters in a parameter template, go to the **Parameter Templates** page and under **Custom Templates** tab, click the template name, change its parameter values and save the changes. Then, apply the changed parameter template to instances. Dynamic parameter changes take effect immediately, but static parameter changes take effect only after you manually reboot the instances.
- Inappropriate parameter settings may have unintended consequences, including degraded performance and system instability. Exercise caution when modifying database parameters and you need to back up data before

modifying parameters in a parameter template. Before applying parameter template changes to a production instance, you should try out these changes on a test instance.

NOTE

Each user can create up to 100 parameter templates.

All TaurusDB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

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

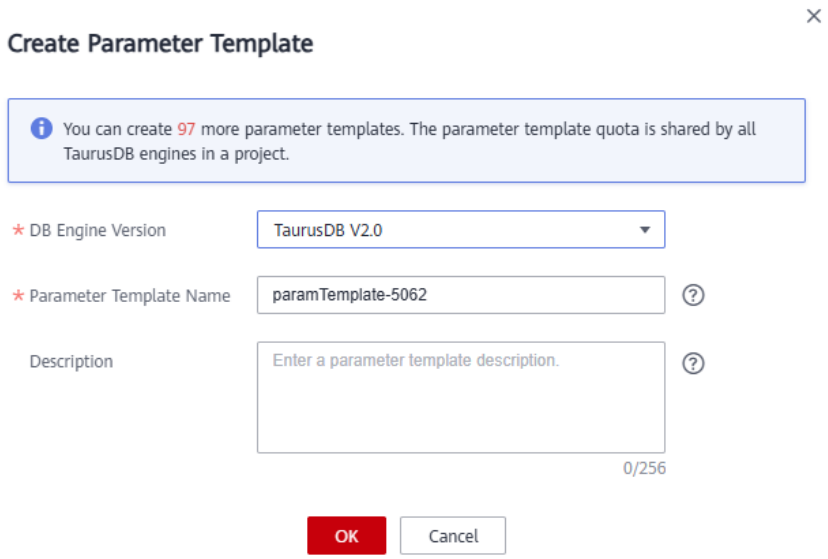
Step 3 Click in the upper left corner of the page, choose **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Parameter Templates**. On the displayed page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, configure required parameters and click **OK**.

- The DB engine is TaurusDB.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description can consist of up to 256 characters. It cannot contain carriage return characters or special characters (>!<"&'=).

Figure 13-1 Creating a parameter template



----End

13.2 Modifying Parameters of a TaurusDB Instance

To ensure optimal performance of TaurusDB, you can modify parameters of DB instances.

Modifying Parameters of a Single 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 **Database** > **TaurusDB**.

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.

After modifying the parameters, you can perform the following operations:

- To save the modifications, click **Save** and then click **Yes**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

NOTICE

- Dynamic parameter modifications take effect immediately, but static parameter modifications take effect only after you manually reboot the instance. After you modify a parameter, check the value in the **Effective upon Reboot** column. You are advised to reboot the instance during off-peak hours and ensure that your applications support automatic reconnection.

Step 6 After the parameters are modified, click **Change History** to view the modification records.

----End

Modifying Parameters in a 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 **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Parameter Templates**.

Step 5 On the **Custom Templates** tab, click the parameter template name.

Step 6 On the displayed **Parameters** page, modify parameters as required.

After modifying the parameters, you can perform the following operations:

- To save the modifications, click **Save** and then click **Yes**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

NOTICE

- Dynamic parameter modifications take effect immediately, but static parameter modifications take effect only after you manually reboot the instance. After you modify a parameter, check the value in the **Effective upon Reboot** column. You are advised to reboot the instance during off-peak hours and ensure that your applications support automatic reconnection.

Step 7 Choose **Change History** in the navigation pane to view the changes.

Step 8 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

Step 9 View the status of the DB instance to which the parameter template was applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

----End

13.3 Exporting Parameters

Scenarios

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.

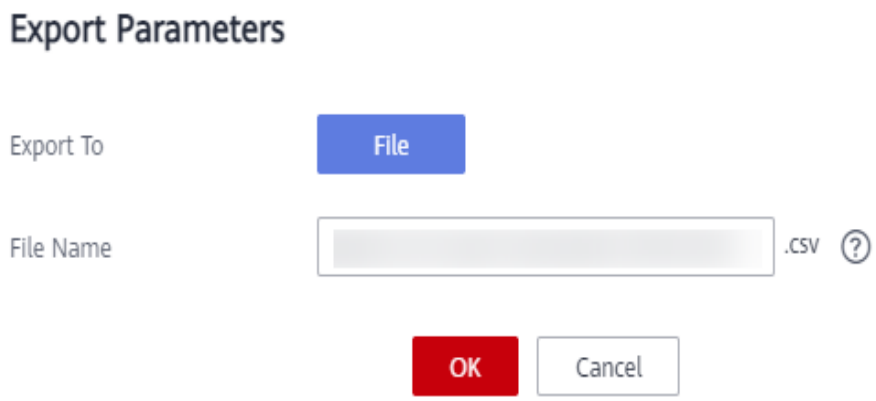
Step 2 Click in the upper left corner and select a region and project.

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

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 **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 13-2 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

13.4 Comparing Parameter Templates

Scenarios

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 a 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 **Database** > **TaurusDB**.

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 **Parameters**. On the displayed page, click **Compare** above the parameter list.

Step 6 In the displayed dialog box, select a parameter template that you want to compare with the current template and click **OK**.

- 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 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 **Database** > **TaurusDB**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be compared and click **Compare** in the **Operation** column.

Step 5 In the displayed dialog box, select a parameter template and click **OK**.

- 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

13.5 Viewing Parameter Change History

Scenarios

You can view the change history of instance parameters or custom parameter templates.

NOTE

If you did not make any change to a parameter template, the change history for the template is blank.

Viewing Change History of Instance Parameters

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 **Database** > **TaurusDB**.

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 **Parameters**. On the displayed page, click **Change History**.

You can view the parameter names, original parameter values, new parameter values, modification statuses, modification time, application statuses, and application time.

You can apply the parameter template to instances if needed. For details, see **Applying a Parameter Template**.

----End

Viewing Change History of a 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 **Database** > **TaurusDB**.

Step 4 In the navigation pane on the left, choose **Parameter Templates**. On the **Custom Templates** page, click the parameter template name.

Step 5 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter names, original parameter values, new parameter values, modification statuses, and modification time.

----End

13.6 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. 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.

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 **Database** > **TaurusDB**.

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, configure required parameters and click **OK**.

- The template name consists of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- 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 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 **Database** > **TaurusDB**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be replicated and click **Replicate** in the **Operation** column.

Step 5 In the displayed dialog box, configure required parameters and click **OK**.

- The template name consists of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- 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

13.7 Resetting a Parameter Template

Scenarios

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 **Database** > **TaurusDB**.

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**.

NOTE

After you reset a parameter template, click the instance to which the parameter template is applied to view the status of the parameter template. On the displayed **Basic Information** page, if the status of the parameter template is **Parameter change. Pending reboot**, you must reboot the instance.

----End

13.8 Applying a Parameter Template

Scenarios

Changes to parameters in a custom parameter template do not take effect until the template is applied to instances.

- 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 **Database** > **TaurusDB**.

Step 4 On the **Parameter Templates** page, apply a default template or a custom template to an instance:

- To apply a default template, click **Default Templates**, locate a parameter template and click **Apply** in the **Operation** column.
- To apply a custom template, click **Custom Templates**, locate a parameter template and choose **More** > **Apply** in the **Operation** column.

A parameter template can be applied to one or more instances.

Step 5 In the displayed dialog box, select one or more instances to which the parameter template will be applied and click **OK**.

After the parameter template is applied, you can view its **application records**.

----End

13.9 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a 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 **Database** > **TaurusDB**.

Step 4 Choose **Parameter Templates** in the navigation pane on the left:

- On the **Default Templates** page, locate a parameter template and click **View Application Record** in the **Operation** column.
- On the **Custom Templates** page, locate a parameter template and choose **More** > **View Application Record** in the **Operation** column.

You can view the name or ID of the instance the parameter template is applied to, as well as the application status, application time, and failure cause.

----End

13.10 Editing a Parameter Template Description

Scenarios

You can edit the description of a parameter template you have created.

NOTE

You cannot edit the description of a 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 **Database** > **TaurusDB**.

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 contains up to 256 characters, and cannot contain carriage return characters and any of the following special characters:

  >!<"&'=

----End

13.11 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer needed.

NOTICE

- 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 **Database** > **TaurusDB**.

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

14 Data Security

14.1 Resetting the Administrator Password

Scenarios

If you forget the password of the administrator account, you can reset the password.

You cannot reset the administrator password under the following circumstances:

- 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, choose **Database** > **TaurusDB**.

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, click the instance name on the **Instances** page to go to the **Basic Information** page. In the **DB Instance Information** area, click **Reset Password** in the **Administrator** field.

Step 5 In the displayed dialog box, enter and confirm the new password.

The new password must:

- Contain 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.

Step 6 Click **OK**.

NOTICE

Keep your password secure. The system cannot retrieve it if it is lost.

----End

14.2 Changing a Security Group

Scenarios

You can change the security group associated with your TaurusDB 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 **Database** > **TaurusDB**.

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

Step 5 In the **Network Information** area on the **Basic Information** page, click  in the **Security Group** field and select a new security group.

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

Step 6 Click  in the upper right corner on the **Basic Information** page to view the result of the change. This process takes about 1 to 3 minutes.

----End

14.3 Configuring SSL

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.

SSL is enabled by default. 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.

Enabling SSL

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 **Database** > **TaurusDB**.

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

Step 5 In the **DB Instance Information** area, click  in the **SSL** field.

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

Step 7 On the **Basic Information** page, view the results.

----End

Disabling SSL

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 **Database** > **TaurusDB**.

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

Step 5 In the **DB Instance Information** area, click in the **SSL** field.

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

Step 7 On the **Basic Information** page, view the results.

----End

14.4 Enabling TDE

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.

Constraints on Usage

- To configure TDE, you must have the iam:agencies:createServiceLinkedAgencyV5 permission.
- You need to enable Key Management Service (KMS) for your TaurusDB instance first. The data keys used for encryption are generated and managed by KMS. TaurusDB does not provide any keys or certificates required for encryption.
- To enable TDE, the kernel version of your TaurusDB instance must be 2.0.47.231100 or later.
- 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:
  - 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.

Step 3 After the DB instance is created, click the DB instance name to go to the **Basic Information** page and view the **TDE** field.

----End

15 Metrics and Alarms

15.1 Supported Monitoring Metrics

Function

You can monitor the status of your instances. The namespaces, descriptions, and dimensions of monitoring metrics of instances can be reported to Cloud Eye.

Namespace

SYS.GAUSSDB

Monitoring Metrics Supported by Instances

Table 15-1 Monitoring metrics supported by TaurusDB instances

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%	TaurusDB instances	1 minute 5 seconds 1 second
gaussdb_mysql002_mem_util	Memory Usage	Memory usage of the monitored object	0–100%	TaurusDB instances	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	TaurusDB instances	1 minute 5 seconds 1 second
gaussdb_mysql005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	TaurusDB instances	1 minute
gaussdb_mysql006_conn_count	Total Connections	Total number of connections that attempt to connect to the TaurusDB server	≥ 0 counts	TaurusDB instances	1 minute 5 seconds 1 second
gaussdb_mysql007_conn_active_count	Current Active Connections	Number of active connections	≥ 0 counts	TaurusDB instances	1 minute 5 seconds 1 second
gaussdb_mysql008_qps	QPS	Query times of SQL statements (including DDL, DML, SHOW, SET statements and storage procedures) per second	≥ 0 times/s	TaurusDB instances	1 minute 5 seconds 1 second
gaussdb_mysql009_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0 times/s	TaurusDB instances	1 minute 5 seconds 1 second
gaussdb_mysql010_innodb_buf_usage	Buffer Pool Usage	Ratio of used pages to total pages in the InnoDB buffer	0-1	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0-1	TaurusDB instances	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%	TaurusDB instances	1 minute
gaussdb_mysql013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥ 0 bytes/s	TaurusDB instances	1 minute
gaussdb_mysql014_innodb_writes	InnoDB Write Throughput	Bytes written to pages by InnoDB per second. TaurusDB writes data only to temporary tables	≥ 0 bytes/s	TaurusDB instances	1 minute
gaussdb_mysql017_innodb_log_write_req_count	InnoDB Log Write Request Frequency	Number of InnoDB log write requests per second	≥ 0 counts	TaurusDB instances	1 minute
gaussdb_mysql019_innodb_log_writes	InnoDB Log Writes	Number of physical writes to the InnoDB redo log file	≥ 0 counts	TaurusDB instances	1 minute
gaussdb_mysql020_temp_tbl_count	Temporary Tables	Number of temporary tables automatically created on disks when TaurusDB statements are executed	≥ 0 counts	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0 counts/s	TaurusDB instances	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	TaurusDB instances	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	TaurusDB instances	1 minute
gaussdb_mysql031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥ 0 counts/s	TaurusDB instances	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0 counts/s	TaurusDB instances	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	TaurusDB instances	1 minute
gaussdb_mysql036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0 GB-128 TB	TaurusDB instances	1 minute
gaussdb_mysql072_conn_usage	Connection Usage	Percent of used TaurusDB connections to the total number of connections	0–100%	TaurusDB instances	1 minute
gaussdb_mysql074_slow_queries	Slow Query Logs	Number of TaurusDB slow query logs generated per minute	≥ 0 counts/min	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql077_replication_delay	Replication Delay	Delay between the primary node and read replicas	≥ 0s	TaurusDB instances	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	TaurusDB instances	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	TaurusDB instances	1 minute
gaussdb_mysql106_innodb_row_lock_current_waits	InnoDB Row Locks	Number of row locks being waited by operations on the InnoDB table	≥ 0 locks/s	TaurusDB instances	1 minute
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	TaurusDB instances	1 minute
gaussdb_mysql108_com_commit_count	COMMIT Statements per Second	Number of COMMIT statements executed per second	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql109_com_rollback_count	ROLLBACK Statements per Second	Number of ROLLBACK statements executed per second	≥ 0 counts/s	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
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	TaurusDB instances	1 minute
gaussdb_mysql111_innodb_bufpool_read_requests	InnoDB Read Requests per Second	Number of InnoDB read requests per second	≥ 0 counts/s	TaurusDB instances	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	TaurusDB instances	1 minute
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	TaurusDB instances	1 minute
gaussdb_mysql116_innodb_bufpool_read_ahead_rnd	InnoDB Bufpool Read Ahead Rnd	Number of random read-aheads initiated by InnoDB	≥ 0 counts	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql117_innodb_pages_read	InnoDB Pages Read	Number of pages read from the InnoDB buffer pool by operations on InnoDB tables	≥ 0 counts	TaurusDB instances	1 minute
gaussdb_mysql118_innodb_pages_written	InnoDB Pages Written	Number of pages written by operations on InnoDB tables	≥ 0 counts	TaurusDB instances	1 minute
gaussdb_mysql119_disk_used_ratio	Disk Usage	Disk usage of the monitored object	0–100%	TaurusDB instances	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	TaurusDB instances	1 minute
gaussdb_mysql121_innodb_row_lock_time	Row Lock Time	Total time spent in acquiring row locks for InnoDB tables	≥ 0 ms	TaurusDB instances	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	TaurusDB instances	1 minute
gaussdb_mysql123_sort_range	Sorts Using Ranges	Number of sorts that were done using ranges	≥ 0 counts/min	TaurusDB instances	1 minute
gaussdb_mysql124_sort_rows	Sorted Rows	Number of sorted rows	≥ 0 counts/min	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql125_sort_scan	Sorts by Scanning Tables	Number of sorts that were done by scanning tables.	≥ 0 counts/min	TaurusDB instances	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	TaurusDB instances	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	TaurusDB instances	1 minute
gaussdb_mysql128_long_trx_count	Long-Running Transactions	Number of long transactions that are not closed	≥ 0 counts	TaurusDB instances	150s
gaussdb_mysql342_iostat_iops_write	I/O Write IOPS	I/O write IOPS	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql344_iostat_iops_read	I/O Read IOPS	I/O read IOPS	≥ 0 counts/s	TaurusDB instances	1 minute
gaussdb_mysql346_iostat_throughput_write	I/O Write Bandwidth	Disk write bandwidth per second	≥ 0 bytes/s	TaurusDB instances	1 minute

Metric ID	Metric	Metric Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gaussdb_mysql348_iostat_throughput_read	I/O Read Bandwidth	Disk read bandwidth per second	≥ 0 bytes/s	TaurusDB instances	1 minute
gaussdb_mysql371_taurus_binlog_total_file_counts	Binlog Files	Number of TaurusDB binlog files	≥ 0	TaurusDB instances	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 TaurusDB statements are executed	≥ 0 counts/min	TaurusDB instances	1 minute

Dimension

Table 15-2 Monitoring metric dimension

Key	Value
gaussdb_mysql_instance_id	TaurusDB instance ID.
gaussdb_mysql_node_id	TaurusDB node ID.

15.2 Viewing Monitoring Metrics

15.2.1 Viewing Instance Monitoring Metrics

Scenarios

Cloud Eye monitors status of your instances. You can view the monitoring metrics of 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

- Instances are running properly.

  Monitoring metrics of the instances that are faulty or have been deleted cannot be displayed on the Cloud Eye console, but you can view them after the instances are rebooted or restored to be available.

  NOTE

  If an instance has been faulty for 24 hours, Cloud Eye considers that it does not exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the instance.

- Instances have kept running properly for about 10 minutes.

  For a newly created instance, you need to wait for a while before viewing the monitoring metrics.

Viewing Monitoring Metrics of Nodes

Step 1 Log in to the management console.

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

Step 3 In the navigation pane, choose **Cloud Service Monitoring** > **TaurusDB**.

Step 4 Click ⌄ in the front of the instance. Locate a node and click **View Metric** in the **Operation** column.

You can also perform the following operations to switch to the Cloud Eye console:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click **View Metric** to go to the Cloud Eye console and view monitoring metrics.
- In the **Node List** area of the **Basic Information** page, locate the primary node or a read replica and click **View Metric** in the **Operation** column to go to the Cloud Eye console and view the monitoring metrics.

Step 5 Click **Select Metric** in the upper right corner. In the displayed dialog box, select the metrics to be displayed and sort them by dragging them at desired locations.

Cloud Eye can monitor performance metrics from the last 1 hour, last 3 hours, last 12 hours, last 24 hours or last 7 days.

----End

Viewing Real-Time Instance Monitoring 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 **Database** > **TaurusDB**.

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

15.3 Configuring Alarm Rules

15.3.1 Creating Alarm Rules for a DB Instance

Scenarios

You can create alarm rules for TaurusDB to customize the monitored objects and notification policies and stay aware of the TaurusDB instance statuses.

The TaurusDB alarm rules include alarm rule names, services, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Creating Alarm Rules for Instances

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Deployment**, click **Cloud Eye**.

Step 3 In the navigation pane, choose **Cloud Service Monitoring** > **TaurusDB**.

Step 4 In the instance list, click ⌄ in the front of the instance. Locate a node and click **Create Alarm Rule** in the **Operation** column.

Step 5 On the **Create Alarm Rule** page, configure parameters as needed..

1. Configure the alarm rule name and description.

   Table 15-3 Parameter description

Parameter	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 content parameters.

Table 15-4 Parameter description

Parameter	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.
Template	Select the template to be used. You can select a default alarm template or create 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 is triggered.

3. Configure alarm notification parameters.

Table 15-5 Parameter description

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 Recipient	You can select a notification group or topic subscription as required.
Notification Group	Specifies the notification group that needs to send alarm notifications.
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. 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.

Table 15-6 Parameter description

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.

Step 6 Click **Create**. The alarm rule is created.

----End

Creating Alarm Rules for 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 **Database** > **TaurusDB**.

Step 4 On the **Instances** page, locate the instance and click **View Metric** 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 displayed **Basic Information** page, click **View Metric** in the upper right corner.
- In the **Node List** area of the **Basic Information** page, locate a node and click **View Metric** in the **Operation** column.

Step 5 Locate the monitoring metric that you want to create an alarm for and click in the upper right corner of the metric.

Step 6 On the **Create Alarm Rule** page, configure parameters as needed..

1. Configure the alarm rule name and description.

Table 15-7 Parameter description

Parameter	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 content parameters.

Table 15-8 Parameter description

Parameter	Description
Method	Specifies the method for triggering an alarm. If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.
Alarm Policy	Specifies the policy for triggering an alarm.

3. Configure alarm notification parameters.

Table 15-9 Parameter description

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 Recipient	You can select a notification group or topic subscription as required.
Notification Group	Specifies the notification group that needs to send alarm notifications.
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. 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.

Table 15-10 Parameter description

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.

Step 7 Click **Create**.

----End

15.4 Configuring Monitoring by Seconds

TaurusDB supports monitoring by seconds. You can set the monitoring interval to 1 second or 5 seconds to view the metric values.

Constraints

By default, monitoring by seconds is unavailable for instances with fewer than eight vCPUs. Instances with monitoring by seconds enabled are not affected.

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, choose **Database** > **TaurusDB**.

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

Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab and click  next to **Monitoring by Seconds**.

Step 7 In the displayed dialog box, select a collection period and click **OK**.

After you enable this function, monitoring data will be reported and displayed by the second after about five minutes.

Step 8 View monitoring metrics. 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.

To change the collection period, click **Modify Collection Period** next to **Monitoring by Seconds**.

----End

Disabling Monitoring by Seconds

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane on the left, choose **Advanced O&M**.

Step 3 On the displayed page, click the **Real-Time Monitoring** tab and click  next to **Monitoring by Seconds**.

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

----End

Modifying Collection Interval

Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane on the left, choose **Advanced O&M**.

Step 3 On the displayed page, click the **Real-Time Monitoring** tab and click **Modify Collection Period** next to **Monitoring by Seconds**.

Step 4 In the displayed dialog box, select a collection period and click **Yes**.

----End

16 Logs and Auditing

16.1 Enabling or Disabling Log Reporting

TaurusDB log management allows you to view database-level logs, including error logs and slow SQL query logs.

If you enable log reporting for your DB instance, new logs generated for the TaurusDB instance will be uploaded to Log Tank Service (LTS) for management.

Constraints

- 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, choose **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Log Reporting**.

Step 5 Select one or more instances and click **Enable Log Reporting**.

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.

----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, choose **Database** > **TaurusDB**.

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 in batches
  a. Select one or more instances and click **Disable Log Reporting**.
  b. In the displayed dialog box, click **OK**.
- 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**.

----End

16.2 Managing Error Logs

TaurusDB error logs contain logs generated during the database 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, choose **Database** > **TaurusDB**.

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.

----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.

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

Step 2 In the navigation pane, choose **Logs**.

Step 3 On the **Error Log** page, click  next to **Report Error Log to LTS**.

Step 4 Select an LTS log group and log stream and click **OK**.

----End

16.3 Managing Slow Query Logs

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.

TaurusDB supports the following statements:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- ALTER
- DROP

Parameter Description

Table 16-1 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_not_using_indexes	Specifies the SQL statement that can be written to the slow query log every minute. The default value is **0**.

Viewing Log Details

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, view the slow query log details.

Step 4 View slow query logs of different nodes in a given database and SQL statement types. In the upper right corner of the page:

Enter a database name, click the drop-down list, and select your desired node.

Click the drop-down list and select a SQL statement type (SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, or DROP).

Click and specify a time period.

----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.

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.

- 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 Log to LTS**.

Step 4 Select an LTS log group and log stream and click **OK**.

----End

16.4 Enabling or Disabling SQL Explorer

Enabling SQL Explorer will allow TaurusDB to store all SQL statement logs for analysis.

You can enable SQL Explorer on the console.

16.5 Interconnection with CTS

16.5.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to TaurusDB for further querying, auditing, and backtracking. **Table 16-2** lists the supported operations.

Table 16-2 TaurusDB 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

Operation	Resource Type	Trace Name
Changing a security group	instance	modifySecurityGroup
Promoting a read replica to the primary node	instance	instanceSwitchOver
Binding or unbinding an EIP	instance	setOrResetPublicIP
Deleting a DB instance	instance	deleteInstance
Renaming a DB instance	instance	renameInstance
Changing a failover priority	instance	modifyPriority
Resetting a password	instance	resetPassword
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
Adding a tag	instance	addInstanceTags
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

Operation	Resource Type	Trace Name
Resetting a parameter template	parameterGroup	resetParameterGroup
Comparing parameter templates	parameterGroup	compareParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup

16.5.2 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 CTS console.

Procedure

Step 1 Click  in the upper left corner and select a region and a project.

Step 2 Choose **Management & Deployment** > **Cloud Trace Service**.

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

Step 4 Filter conditions to query traces.

Table 16-3 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.

Filtering Criteria	Description
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.
Trace Status	Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.

Step 5 View the events that meet the search criteria.

Step 6 Click an event name. Details about the event are displayed in the dialog box on the right.

Step 7 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 CTS trace structure, see sections "**Trace Structure**" and "**Example Traces**" in the *Cloud Trace Service User Guide*.

----End

17 Task Center

17.1 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 **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Task Center**. On the displayed **Instant Tasks** tab page, locate the target task and view its details.

- 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 TaurusDB instance
    - Creating a TaurusDB read replica
    - Rebooting a TaurusDB instance
    - Changing a TaurusDB port
    - Promoting a TaurusDB read replica to the new primary node
    - Binding an EIP to a TaurusDB instance

- Unbinding an EIP from a TaurusDB instance
- Changing a TaurusDB instance name
- Changing the security group of a TaurusDB instance
- Deleting a TaurusDB instance
- Deleting a TaurusDB read replica
- Changing the specifications of a TaurusDB instance
- Restoring data to a new TaurusDB instance
- Changing the private IP address of a TaurusDB instance
- Changing the collection period of TaurusDB monitoring by seconds
- Enabling or disabling SSL of a TaurusDB instance
- Restoring data to an existing TaurusDB instance
- Rebooting a node of a TaurusDB instance
- Changing the node name of a TaurusDB instance
- Upgrading the TaurusDB instance version
- Pre-checking the TaurusDB kernel version for an upgrade
- Upgrading the version of the proxy instance
- Upgrading the kernel of a TaurusDB instance
- Modifying TaurusDB database remarks
- Modifying remarks of a TaurusDB 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 **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Task Center**. On the **Scheduled Tasks** page, view the task progresses and results.

- 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

17.2 Deleting a Task Record

You can delete the task records that no longer need to be displayed. The deletion only deletes the task records. It does not delete the instances or terminate tasks in progress.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

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 **Database** > **TaurusDB**.

Step 4 In the navigation pane, choose **Task Center**. Locate the task record to be deleted on the **Instant Tasks** tab and click **Delete** in the **Operation** column.

Step 5 In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

You can delete 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 **Database** > **TaurusDB**.

Step 4 Choose **Task Center** in the navigation pane. 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

18 Managing Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage 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, choose **Database** > **TaurusDB**.

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

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

- 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 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 **Database** > **TaurusDB**.

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

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

- Only the tag value can be edited.
- 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

Deleting 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, choose **Database** > **TaurusDB**.

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

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

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

----End

19 Managing Quotas

Scenarios

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of TaurusDB quotas include the maximum number of DB instances that you can create. Quotas are put in place to prevent excessive resource usage.

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.

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**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

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

