TaurusDB

User Guide

Issue 01

Date 2025-06-12

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 Selection Guide

Overview

Before purchasing a TaurusDB instance, consider factors such as the price, performance, workload capacity, and service scenario to choose the right one for your needs. This section describes the differences between types, billing modes, and storage types of TaurusDB instances, helping you select the most suitable instance.

Storage Engine

By default, TaurusDB uses the InnoDB storage engine to provide high-performance and reliable transaction processing. The TaurusDB kernel provides functions such as parallel query, table recycle bin, hot row update, and multi-tenancy. It is widely used in high-performance and high-concurrency scenarios.

Storage Types

TaurusDB provides two storage types: DL6 and DL5. The table below explains their key differences to help you choose the best option for your needs.

Table 1-1 Storage type description

Storage Type	Characteristic	Applicable Scenario
DL6	The shared storage is the default storage type for TaurusDB instances created before July 2024. DL6-based instances achieve zero RPO with a 3-AZ deployment and deliver better performance and higher peak throughput.	Core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming

Storage Type	Characteristic	Applicable Scenario
DL5	The new type of storage uses Huawei Cloud's hardware and network infrastructure technologies, ensuring that DL5-based instances maintain the same high availability (zero RPO in the 3-AZ deployment) as DL6-based instances. Although the peak performance may decrease, the cost per unit of capacity is significantly reduced.	CPU-intensive sub-core business systems, or application modules that need to minimize costs

Billing Modes

There are yearly/monthly, pay-per-use, and serverless billing modes to meet requirements in different scenarios.

Table 1-2 Comparison of billing modes

Billing Mode	Yearly/Monthly	Pay-per-Use	Serverless
Payment	Prepaid Billed by the required duration specified in your order	Postpaid Billed for what you use	Postpaid Billed for what you use
Supported Instance Type	• Single-node • Cluster	• Single-node • Cluster • Multi-primary	Serverless
Billing Cycle	Billed by the required duration specified in your order	Billed by second and settled by hour	Billed by second and settled by hour
Billing Item	Instance specifications (vCPUs and memory), storage space, backup space, EIP, and monitoring by seconds	Instance specifications (vCPUs and memory), storage space, backup space, EIP, and monitoring by seconds	Compute, storage space, backup space, EIP, and monitoring by seconds

Billing Mode Change	Can be changed to pay-per-use. The change is only applied after the yearly/monthly subscription expires. For details, see **Yearly/Monthly to Pay-per-Use**.	Can be changed to yearly/monthly. For details, see **Pay-per-Use to Yearly/ Monthly**.	N/A
Applicable Scenario	Recommended for resources expected to be in use long-term. A cost-effective option for scenarios where the resource usage duration is predictable.	Good for short-term, bursty, or unpredictable workloads that cannot tolerate any interruptions, such as applications for e-commerce flash sales, temporary testing, and scientific computing.	The instance capacities automatically change based on application requirements.

Yearly/Monthly

The table below lists the billing items.

Table 1-3 Billing items

Billing Item	Description	Billing Factor
(Mandatory) DB instance	You are billed for the selected instance specifications, including vCPUs, memory, and nodes.	vCPUs, memory, and number of nodes
(Mandatory) Storage	Pre-purchased storage is billed on a yearly/monthly basis. However, if the actual usage exceeds your purchased storage, you will be billed for additional storage on a pay-per-use basis.	Storage space, which is billed based on the unified standard
(Mandatory) Backup space	TaurusDB provides free backup space equal to the amount of your purchased storage space. After the free backup space is used up, charges are applied based on the backup space pricing details. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.	Backup space, which is billed based on the unified standard

Billing Item	Description	Billing Factor
(Optional) EIP bandwidth	• TaurusDB instances can be accessed through a public network, and traffic fees are generated accordingly. • You are not billed for traffic generated through a private network.	Bandwidth, traffic, and EIP reservation (An EIP is required if a DB instance needs to access the Internet.) EIP for a yearly/monthly DB instance: billed by bandwidth
(Optional) Monitoring by seconds	TaurusDB provides monitoring every 60 seconds for free, but you are billed for monitoring by seconds. Its pricing is listed on a per-hour basis, but bills are calculated based on actual usage.	Monitoring frequency, which is billed based on the unified standard
(Optional) HTAP instance	If you create a standard HTAP instance for a TaurusDB instance, you will be billed for the HTAP instance.	Instance specifications and storage type
(Optional) Cross-region backup	If cross-region backup is enabled, you will be billed for backup space and network traffic required for dumping cross-region backups.	Backup space and network traffic required for dumping cross-region backups
(Optional) Proxy instance	Proxy instances are free.	N/A
(Optional) DRS migration	If you use Data Replication Service (DRS) for data migration, you will be billed based on the DRS pricing standard.	For details, see **DRS Billing**.

Pay-per-Use

The table below lists the billing items.

Table 1-4 Billing items

Billing Item	Description	Billing Factor
(Mandatory) DB instance	You are billed for the selected instance specifications, including vCPUs, memory, and nodes.	vCPUs, memory, and number of nodes

Billing Item	Description	Billing Factor
(Mandatory) Storage	Pre-purchased storage is billed on a yearly/monthly basis.	Storage space, which is billed based on the unified standard
(Mandatory) Backup space	After the free backup space is used up, charges are applied based on the backup space pricing details. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.	Backup space, which is billed based on the unified standard
(Optional) EIP bandwidth	• DB instances can be accessed through a public network, and traffic fees are generated accordingly. • You are not billed for traffic generated through a private network.	Bandwidth, traffic, and EIP reservation (An EIP is required if a DB instance needs to access the Internet.) • Billing by bandwidth: bandwidth fee + EIP reservation fee • Billing by traffic: traffic fee + EIP reservation fee • Shared bandwidth added: bandwidth fee + EIP reservation fee
(Optional) Monitoring by seconds	Enabling monitoring by seconds will incur additional fees. Its pricing is listed on a per-hour basis, but bills are calculated based on actual usage.	Monitoring frequency, which is billed based on the unified standard
(Optional) HTAP instance	If you create a standard HTAP instance for a TaurusDB instance, you will be billed for the HTAP instance.	Instance specifications and storage type
(Optional) Cross-region backup	If cross-region backup is enabled, you will be billed for backup space and network traffic required for dumping cross-region backups.	Backup space and network traffic required for dumping cross-region backups
(Optional) Proxy instance	Proxy instances are free.	N/A
(Optional) DRS migration	If you use Data Replication Service (DRS) for data migration, you will be billed based on the DRS pricing standard.	For details, see **DRS Billing**.

Serverless

The table below lists the billing items.

Table 1-5 Billing items

Billing Item	Description	Billing Factor
DB instance	You are billed for the selected instance specifications. The billing starts immediately after the instance is purchased.	vCPUs, memory, and number of nodes
Storage space	You do not need to select storage when purchasing a DB instance. Storage will be scaled up dynamically based on how much data needs to be stored. It is billed hourly on a pay-per-use basis.	Storage space, which is billed based on the unified standard
Backup space	TaurusDB provides free backup space equal to the amount of your used storage space. If the backup space usage exceeds 100% of your provisioned database storage, the additional part will be billed based on the backup pricing.	Backup space, which is billed based on the unified standard
(Optional) Public network traffic	TaurusDB instances are accessible from both private and public networks, but only the traffic from public networks is billed.	Bandwidth, traffic, and EIP reservation (An EIP is required if a DB instance needs to access the Internet.)
(Optional) HTAP instance	If you create a standard HTAP instance for a TaurusDB instance, you will be billed for the HTAP instance.	Instance specifications and storage type

Instance Type

TaurusDB supports cluster, single-node, and multi-primary instances. You can select an instance type based on your service scenario and scale. For details, see **Table 1-6**.

Table 1-6 Instance type description

Instance Type	Description	Highlight	Applicable Scenario
Cluster	DB engine version: 8.0 Node: 1 primary node + 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.	Medium- and large-sized enterprises in the Internet, taxation, banking, and insurance sectors
Single-node	DB engine version: 8.0 Node: 1 primary node + 0 read replicas	Single-node instances do not involve data synchronization between nodes and can easily ensure atomicity, consistency, isolation, and durability of transactions.	Development and testing of microsites, and small and medium enterprises, or learning about TaurusDB
Serverless	DB engine version: 8.0 Node: 1 primary node + 0 to 7 read replicas	The instance capacities automatically change based on application requirements.	Scenarios where capacity needs to be automatically expanded based on services
Multi-primary	DB engine version: 8.0 Node: 2 to 63 primary nodes + 0 read replicas	Such an instance can process multiple reads and writes, delivering excellent read/write performance at high concurrency.	Scenarios where high concurrent write performance is required

Instance Specifications

There are general-purpose and dedicated specifications for pay-per-use and yearly/monthly TaurusDB instances, as listed in **Table 1-7**. For a serverless instance, you do not need to select the instance specifications. You only need to specify a compute range. A serverless instance is billed by TCU. 1 TCU is approximately equal to 1 vCPU and 2 GB of memory.

Table 1-7 Instance specifications

Instance Specification	Supported CPU Architecture	Description	Applicable Scenario
General-purpose	x86	The vCPUs and memory are shared with other general-purpose instances on the same physical machine. vCPU usage is maximized through resource overcommitment. General-purpose instances are cost-effective.	Scenarios where stable performance is not critical
Dedicated	x86 and Kunpeng	Your instance gets dedicated vCPUs and memory, so the performance is stable. It is not affected by other instances on the same physical machine.	Scenarios that require stable performance

Functions Supported by Different Instance Types

Table 1-8 Reference for instance type selection

Reference		Single-node	Cluster	Serverless	Multi-primary (OBT)
Specifications change	**Changing the vCPUs and Memory of a DB Instance or Node**	√	√	x	x
	Changing the Compute Range	x	x	√	x

Reference		Single-node	Cluster	Serverless	Multi-primary (OBT)
	Changing the Storage Space of a DB Instance	√ Supported only by yearly/ monthly instances	√ Supported only by yearly/ monthly instances	x	x
	Configuring Autoscaling for a DB Instance (OBT)	√ Supported only by yearly/ monthly instances	√ Supported only by yearly/ monthly instances	x	x
Version upgrade	Upgrading the Minor Version of a DB Instance	√	√	√	x
Backup and restoration	Data Backups	√	√	√	x
	Data Restorations	√	√	√	x
Data migration	Migrating Data to TaurusDB Enterprise Edition (OBT)	x	√	x	x
Proxy instance	Changing Configurations of a Proxy Instance	√	√	√	√
	Upgrading the Kernel Version of a Proxy Instance	√	√	√	√
DBA Assistant	Performance Monitoring	√	√	√	x

Reference		Single-node	Cluster	Serverless	Multi-primary (OBT)
	Problem Diagnosis	√	√	√	x
	SQL Analysis and Tunning	√	√	√	x
Security and encryption	Configuring SSL for a DB Instance	√	√	√	x
	Enabling TDE for a DB Instance	√	√	√	x
Cold and hot data separation (OBT)	Configuring a Cold Table	x	√	x	x
HTAP analysis	What Is HTAP of Standard Edition?	x	√	√	x
Tag	Tag Management	√	√	√	√

2 Using IAM to Grant Access to TaurusDB

2.1 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, skip this section.

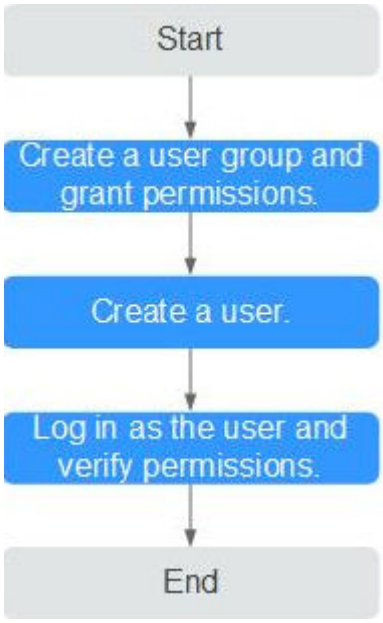
Figure 2-1 describes the procedure for granting permissions.

Prerequisites

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

Process Flow

Figure 2-1 Process for granting TaurusDB permissions



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

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

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

2. **Create an IAM user**.

   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.

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

2.2 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.

For details, see **Creating a Custom Policy**. This section provides 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"
            }
      ]
}
```


3 Buying a DB Instance

3.1 Buying a Pay-per-Use DB Instance

Scenarios

This section describes how to create a pay-per-use DB instance on the TaurusDB console.

Billing

After you buy a pay-per-use DB instance, you will be billed for resources you actually use. For billing details, see **Pay-per-Use Billing**.

Procedure

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

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

- Basic configuration

Figure 3-1 Basic configuration

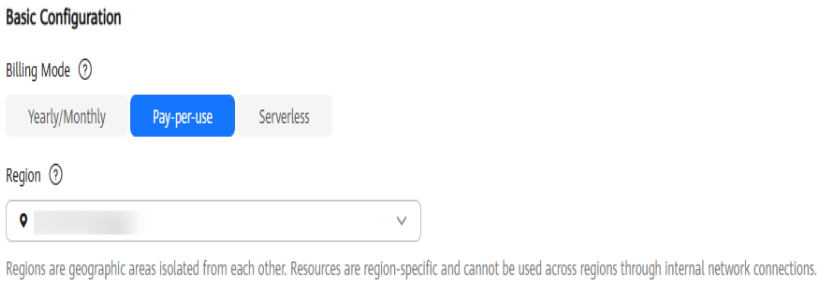


Table 3-1 Basic configuration

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	Region where an instance is deployed. **NOTE** You cannot change the region of an instance once it is purchased.

- Resource selection

Figure 3-2 Resource selection

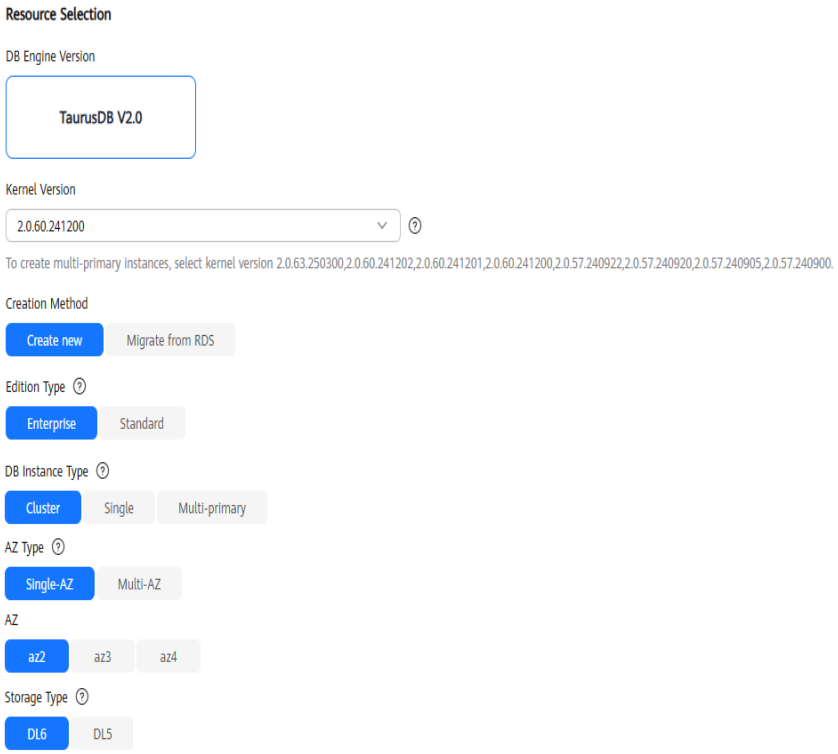


Table 3-2 Resource selection

Parameter	Description
DB Engine Version	Only **TaurusDB V2.0** is supported.
Kernel Version	DB kernel version. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** To specify the kernel version when buying an instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Parameter	Description
Creation Method	How an instance is created. The value can be **Create new** or **Migrate from RDS**. – **Create new**: A new TaurusDB instance will be created. – **Migrate from RDS**: If you want to **migrate data from an RDS instance to a TaurusDB Enterprise Edition instance**, select **Migrate from RDS**. To use this function, **submit a service ticket**.
DB Instance Type	Select **Cluster**, **Single**, or **Multi-primary**. – **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. – **Multi-primary**: A multi-primary instance can contain 2 to 63 primary nodes, with no read replicas. Such an instance can process multiple reads and writes, delivering excellent read/write performance at high concurrency. For more information about multi-primary instances, see **Multi-primary Instances (OBT)**. The kernel version of a multi-primary instance must be: 2.0.63.250300, 2.0.60.241201, 2.0.60.241200, 2.0.57.240922, 2.0.57.240920, 2.0.57.240905, or 2.0.57.240900 **NOTE** To buy a multi-primary instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Parameter	Description
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. – **Single-AZ**: The primary node and read replicas are deployed in the same AZ. – **Multi-AZ**: The primary node and read replicas are deployed in different AZs to achieve higher availability and reliability. It is suitable for workloads that require cross-AZ DR or are insensitive to cross-AZ latency.
Storage Type	– **DL6** The original shared storage. The default storage type of TaurusDB instances created before July 2024 is shared storage, while that of TaurusDB instances created in July 2024 and beyond is **DL6**. DL6-based instances achieve zero RPO with a 3-AZ deployment and deliver better performance and higher peak throughput. They are suitable for core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming. – **DL5** A new type of storage. With Huawei Cloud's hardware and network infrastructure technologies, DL5-based instances maintain the same high availability (zero RPO in the 3-AZ deployment) as DL6-based instances. Although the peak performance of DL5-based instances may be a bit less than what you get with DL6-based instances, the cost per unit of capacity is a lot less. DL5-based instances are suitable for CPU-intensive sub-core business systems, or application modules that need to minimize costs. For more information about storage types, see **Storage Types**.

- Instance options

Figure 3-3 Specifications and storage

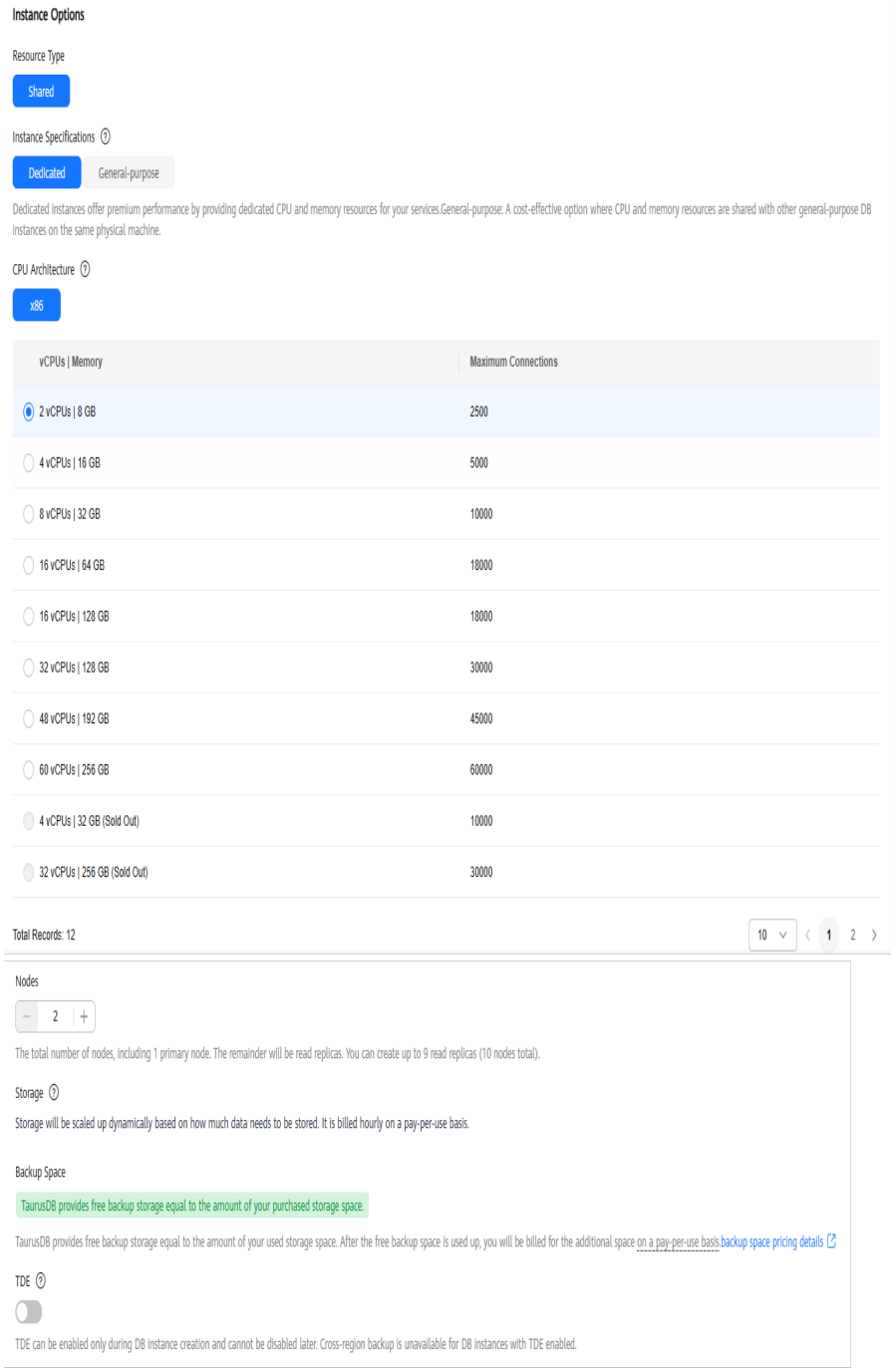


Table 3-3 Specifications and storage

Parameter	Description
Resource Type	Select **Shared**.

Parameter	Description
Instance Specifications	TaurusDB is a cloud-native database that uses the shared storage. To ensure that instances run stably under high read/write pressure, TaurusDB controls the read/write peaks of instances based on instance specifications. For details about how to select instance specifications, see **Performance White Paper**. For more information about instance specifications, see **Instance Specifications**. After a DB instance is created, you can **change its vCPUs and memory**.
CPU Architecture	Select **x86** or **Kunpeng**. – **x86**: x86 instances use Intel® Xeon® Scalable processors and feature robust and stable computing performance. When working on high-performance networks, the instances provide the additional performance and stability that enterprise-class applications demand. – **Kunpeng**: Kunpeng instances use Kunpeng 920 processors and 25GE high-speed intelligent NICs for powerful compute and high-performance networks, making them an excellent choice for enterprises needing cost-effective, secure, and reliable cloud services.
Nodes	This parameter is mandatory for cluster instances. – By default, each instance can contain one primary node and multiple read replicas. – You can create up to 9 read replicas for a pay-per-use instance at a time. – After an instance is created, you can add read replicas as required. Up to 15 read replicas can be added to an instance. For details, see **Adding Read Replicas to a DB Instance**.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage of a pay-per-use instance will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
Backup Space	TaurusDB provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis.

Parameter	Description
TDE	Transparent Data Encryption (TDE) encrypts data files and backup files using certificates to implement real-time I/O encryption and decryption. This function effectively protects the security of databases and data files. After TDE is enabled, you need to select the cryptographic algorithm **AES256** or **SM4** as needed. **NOTE** – To use TDE, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – For details about TDE constraints, see **Enabling TDE for a DB Instance**.

Figure 3-4 Network

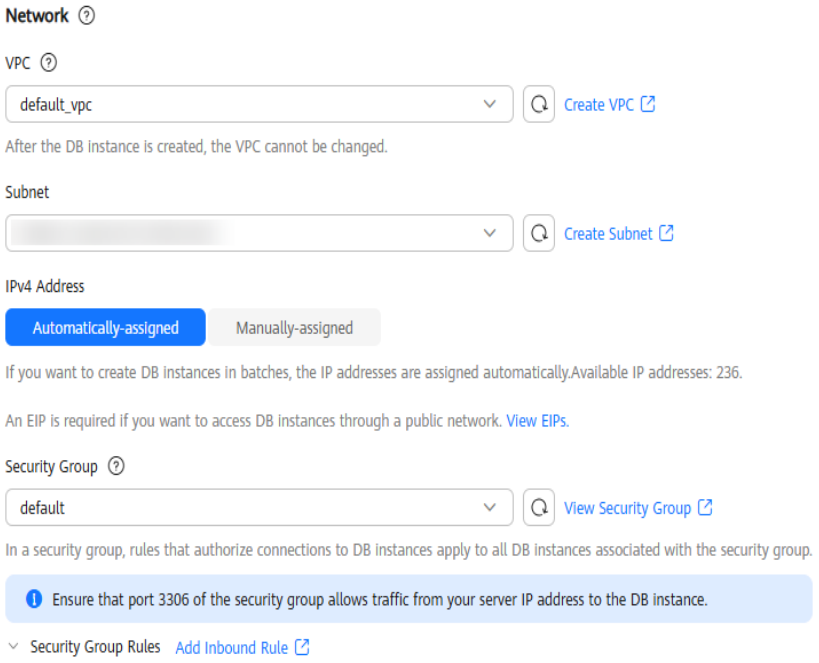
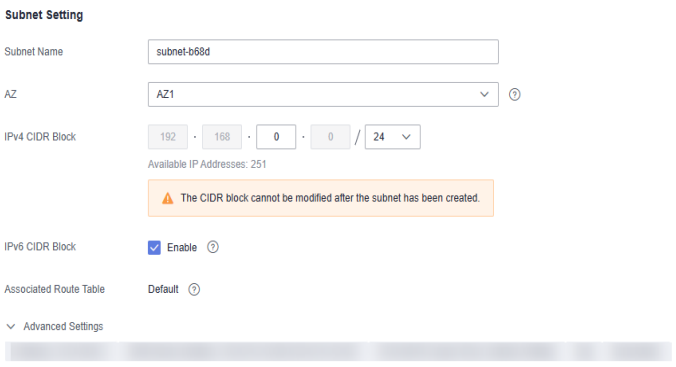


Table 3-4 Network

Parameter	Description
VPC	A dedicated virtual network where your instance is located. It isolates networks for different workloads to enhance security. TaurusDB allocates a default VPC (**default_vpc**) for your instance. You can also use an existing, new, or shared VPC. **After a TaurusDB instance is created, the VPC cannot be changed.** – To use an existing VPC, select an existing VPC under the current account from the drop-down list. – To use a new VPC, create a VPC first and then select the VPC from the drop-down list. For details about how to create a VPC, see **Creating a VPC and Subnet** in *Virtual Private Cloud User Guide*. – To use a shared VPC, select a VPC that another account shares with the current account from the drop-down list. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts, so you can easily configure and manage multiple accounts' resources at low costs. For more information about VPC subnet sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within an AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets where DB instances are located and cannot be disabled. TaurusDB supports both IPv4 and IPv6 networks. Instances using IPv4 and IPv6 cannot be in the same subnet. – IPv4 A private IPv4 address is automatically assigned when you create a DB instance. You can also enter an idle private IPv4 address within the subnet CIDR block. After the DB instance is created, the private IPv4 address can be changed. – IPv6 IPv6 addresses are used to deal with IPv4 address exhaustion. To enable IPv6, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. After IPv6 is enabled, if you want to create an instance using IPv4, you need to select an IPv4 subnet. You can create a DB instance that uses a private IPv6 address only when its specifications support IPv6. Instance specifications supporting IPv6 vary depending on regions and AZs. Whether instance specifications support IPv6 is displayed on the console after a region and an AZ are selected. IPv6 needs to be enabled for subnets where TaurusDB instances are located. If IPv6 is not enabled, enable it by following the instructions provided in **Creating a VPC and Subnet**. **Figure 3-5** Enabling IPv6 for a subnet 

Parameter	Description
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 DB instances. If no security group is available or has been created, TaurusDB allocates a security group to your DB instance by default. To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. If the port and protocol are not enabled for the selected security group, click **Add Inbound Rule** as prompted and complete the configuration in the displayed dialog box. For details, see **Configuring Security Group Rules**.

Figure 3-6 Setting an administrator password

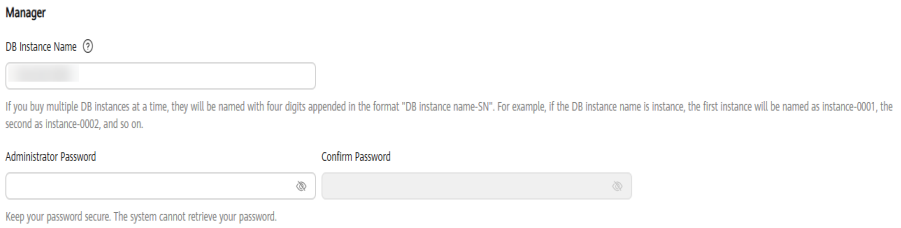


Table 3-5 Database configuration

Parameter	Description
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. – The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

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

- Advanced settings and required quantity

Figure 3-7 Advanced settings and required quantity

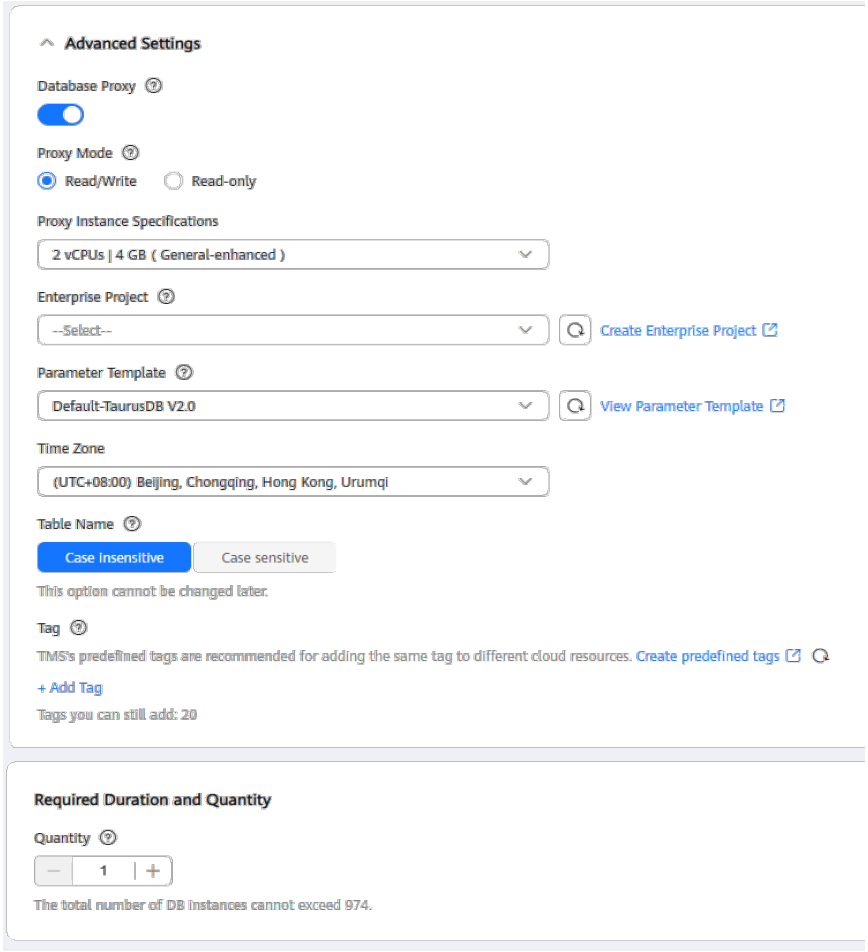


Table 3-6 Advanced settings

Parameter	Description
Database Proxy	Enabled by default. After a proxy instance is created, you can use the proxy address to connect to your DB instance. **NOTE** – To create a proxy instance when buying a DB instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – You can also create proxy instances after buying a DB instance. For details, see **Creating a Proxy Instance for Read/Write Splitting**.
Proxy Mode	You can select **Read/Write** or **Read-only** as needed. – **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights. – **Read-only**: Write requests are not forwarded to any node. The primary node does not process write and read requests, and all read requests are forwarded to the selected read replicas based on read weights.

Parameter	Description
Proxy Instance Specifications	You can select the proxy instance specifications as needed.
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**.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can select the default parameter template, the high-performance parameter template, or a custom parameter template in the current region as required. For details about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters of a DB Instance**.
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 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.
Tag	Tags a DB instance. This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.
Quantity	You can buy DB instances in batches. The default value is **1**. The value ranges from 1 to 10.

Step 3 Confirm the settings.

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

Step 4 Refresh the DB instance list and view the status of the DB instance. If the status changes from **Creating** to **Available**, it has been created successfully. You can manage the DB instance on the **Instances** page.

The automated backup policy is enabled by default and cannot be disabled, and a full backup will be automatically created.

----End

Follow-up Operations

After an instance is created, you can enter a description for it. For details, see **Modifying a DB Instance Description**.

The default database port is **3306**. You can change it after an instance is created. To ensure data and instance security, you are advised to change the database port in a timely manner. For details, see **Changing a Database Port**.

APIs

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

3.2 Buying a Yearly/Monthly DB Instance

Scenarios

This section describes how to create a yearly/monthly DB instance on the TaurusDB console.

Billing

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

Prerequisites

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

Procedure

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

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

- Basic configuration

  Figure 3-8 Basic configuration

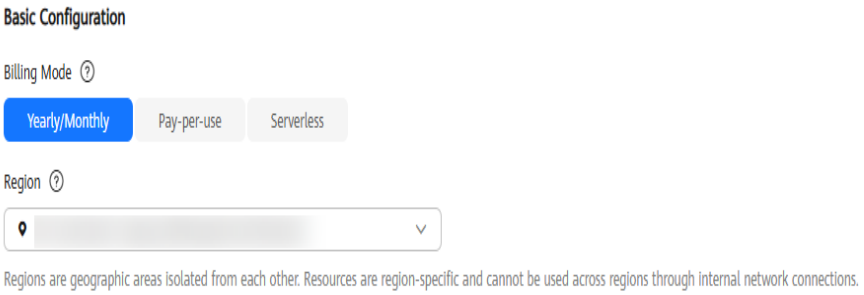
  

  Table 3-7 Basic configuration

Parameter	Description
Billing Mode	Select **Yearly/Monthly**.
Region	Region where an instance is deployed. **NOTE** You cannot change the region of an instance once it is purchased.

- Resource selection

Figure 3-9 Resource selection

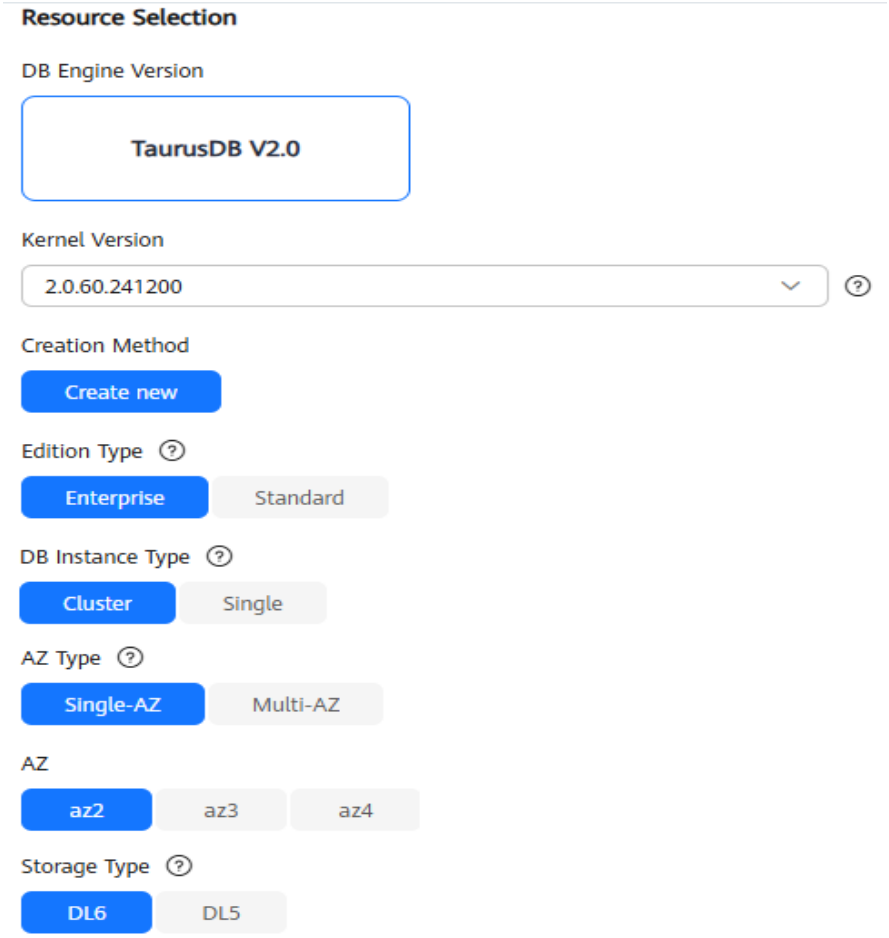


Table 3-8 Resource selection

Parameter	Description
DB Engine Version	Select **TaurusDB V2.0**.
Kernel Version	DB kernel version. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** To specify the kernel version when buying an instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
Creation Method	Select **Create new**.

Parameter	Description
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. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. – **Single-AZ**: The primary node and read replicas are deployed in the same AZ. – **Multi-AZ**: The primary node and read replicas are deployed in different AZs to achieve higher availability and reliability. It is suitable for workloads that require cross-AZ DR or are insensitive to cross-AZ latency.

Parameter	Description
Storage Type	– **DL6** The original shared storage. The default storage type of TaurusDB instances created before July 2024 is shared storage, while that of TaurusDB instances created in July 2024 and beyond is **DL6**. DL6-based instances achieve zero RPO with a 3-AZ deployment and deliver better performance and higher peak throughput. They are suitable for core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming. – **DL5** A new type of storage. With Huawei Cloud's hardware and network infrastructure technologies, DL5-based instances maintain the same high availability (zero RPO in the 3-AZ deployment) as DL6-based instances. Although the peak performance of DL5-based instances may be a bit less than what you get with DL6-based instances, the cost per unit of capacity is a lot less. DL5-based instances are suitable for CPU-intensive sub-core business systems, or application modules that need to minimize costs. For more information about storage types, see **Storage Types**.

- Instance options

Figure 3-10 Specifications and storage

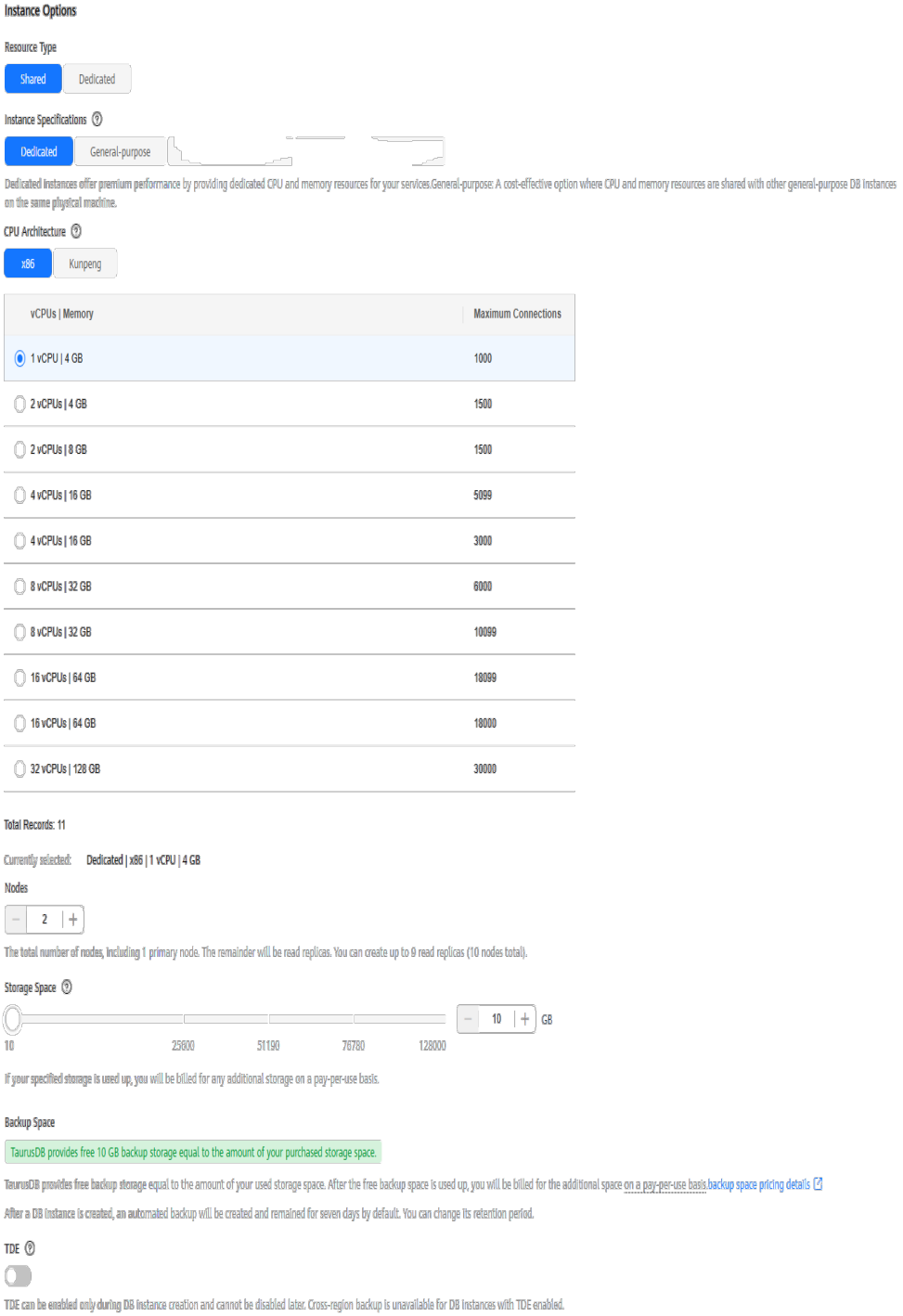


Table 3-9 Specifications and storage

Parameter	Description
Instance Specifications	TaurusDB is a cloud-native database that uses the shared storage. To ensure that instances run stably under high read/write pressure, TaurusDB controls the read/write peaks of instances based on instance specifications. For details about how to select instance specifications, see **Performance White Paper**. For more information about instance specifications, see **Instance Specifications**. After a DB instance is created, you can **change its vCPUs and memory**.
CPU Architecture	Select **x86** or **Kunpeng**. – **x86**: x86 instances use Intel® Xeon® Scalable processors and feature robust and stable computing performance. When working on high-performance networks, the instances provide the additional performance and stability that enterprise-class applications demand. – **Kunpeng**: Kunpeng instances use Kunpeng 920 processors and 25GE high-speed intelligent NICs for powerful compute and high-performance networks, making them an excellent choice for enterprises needing cost-effective, secure, and reliable cloud services.
Nodes	This parameter is mandatory for cluster instances. – By default, each instance can contain one primary node and multiple read replicas. – You can create up to 9 read replicas for a yearly/monthly instance at a time. – After an instance is created, you can add read replicas as required. Up to 15 read replicas can be added to an instance. For details, see **Adding Read Replicas to a DB Instance**.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage space ranges from 40 GB to 128,000 GB and must be a multiple of 10. After a DB instance is created, you can change its storage space. **NOTE** If you want to create a DB instance with storage of at least 10 GB, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Parameter	Description
Backup Space	TaurusDB provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis. If you purchase *X* GB storage billed on a yearly/monthly basis and *Y* GB storage billed on a pay-per-use basis, you will get (*X* + *Y*) GB backup space for free.
TDE	Transparent Data Encryption (TDE) encrypts data files and backup files using certificates to implement real-time I/O encryption and decryption. This function effectively protects the security of databases and data files. After TDE is enabled, you need to select the cryptographic algorithm **AES256** or **SM4** as needed. **NOTE** – To use TDE, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – For details about TDE constraints, see **Enabling TDE for a DB Instance**.

Figure 3-11 Network

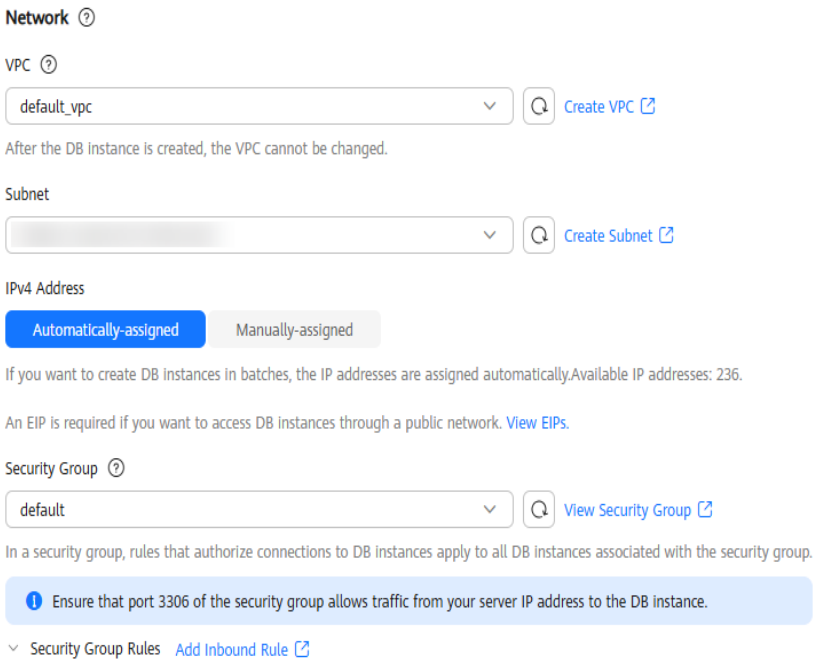
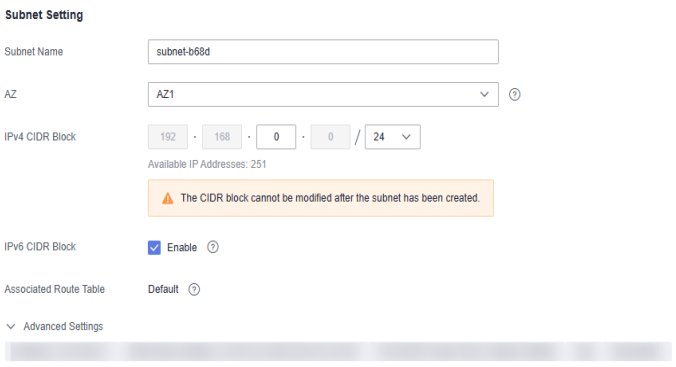


Table 3-10 Network

Parameter	Description
VPC	A dedicated virtual network where your instance is located. It isolates networks for different workloads to enhance security. TaurusDB allocates a default VPC (**default_vpc**) for your instance. You can also use an existing, new, or shared VPC. **After a TaurusDB instance is created, the VPC cannot be changed.** – To use an existing VPC, select an existing VPC under the current account from the drop-down list. – To use a new VPC, create a VPC first and then select the VPC from the drop-down list. For details about how to create a VPC, see **Creating a VPC and Subnet** in *Virtual Private Cloud User Guide*. – To use a shared VPC, select a VPC that another account shares with the current account from the drop-down list. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts, so you can easily configure and manage multiple accounts' resources at low costs. For more information about VPC subnet sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within an AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets where DB instances are located and cannot be disabled. TaurusDB supports both IPv4 and IPv6 networks. Instances using IPv4 and IPv6 cannot be in the same subnet. – IPv4 A private IPv4 address is automatically assigned when you create a DB instance. You can also enter an idle private IPv4 address within the subnet CIDR block. After the DB instance is created, the private IPv4 address can be changed. – IPv6 IPv6 addresses are used to deal with IPv4 address exhaustion. To enable IPv6, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. After IPv6 is enabled, if you want to create an instance using IPv4, you need to select an IPv4 subnet. You can create a DB instance that uses a private IPv6 address only when its specifications support IPv6. Instance specifications supporting IPv6 vary depending on regions and AZs. Whether instance specifications support IPv6 is displayed on the console after a region and an AZ are selected. IPv6 needs to be enabled for subnets where TaurusDB instances are located. If IPv6 is not enabled, enable it by following the instructions provided in **Creating a VPC and Subnet**. **Figure 3-12** Enabling IPv6 for a subnet 

Parameter	Description
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 DB instances. If no security group is available or has been created, TaurusDB allocates a security group to your DB instance by default. To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. If the port and protocol are not enabled for the selected security group, click **Add Inbound Rule** as prompted and complete the configuration in the displayed dialog box. For details, see **Configuring Security Group Rules**.

Figure 3-13 Setting an administrator password

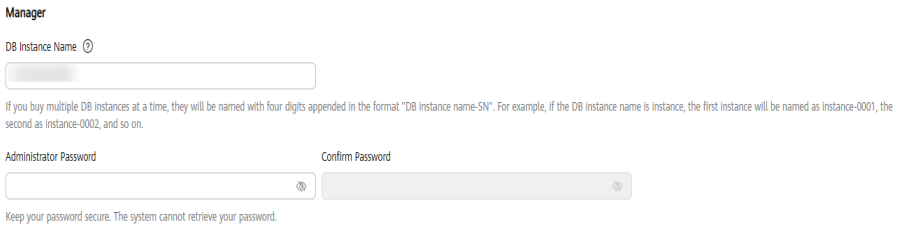


Table 3-11 Database configuration

Parameter	Description
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. – The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

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

- Advanced settings

Figure 3-14 Advanced settings

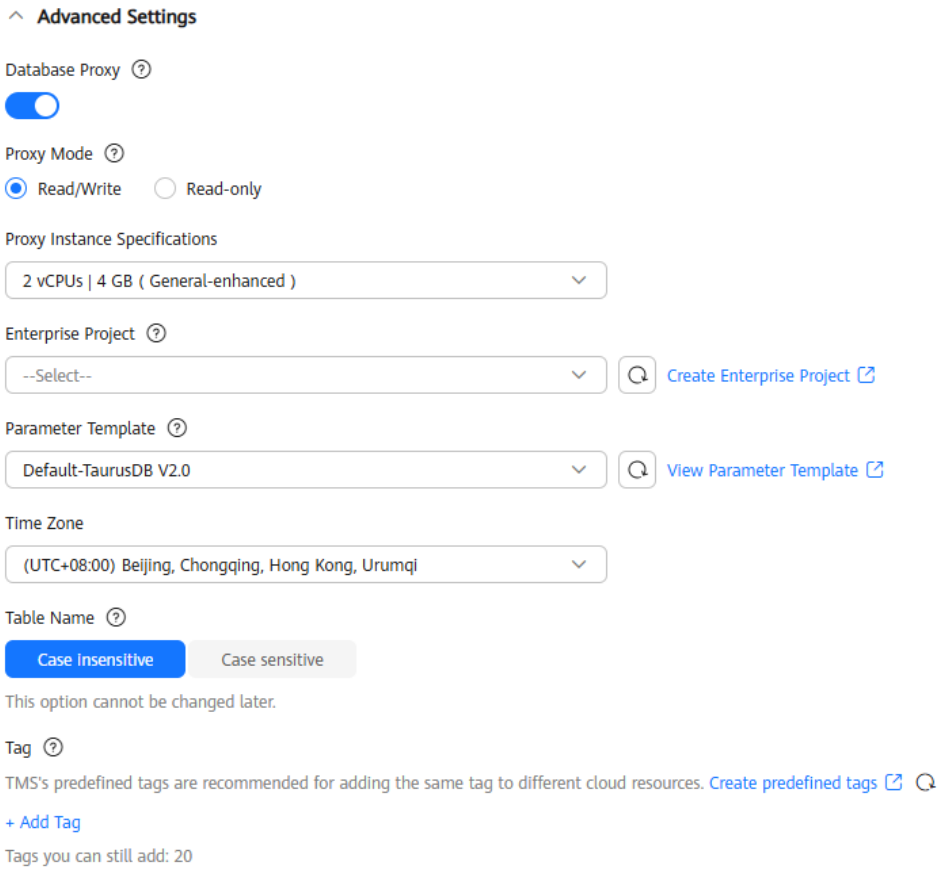


Table 3-12 Database proxy

Parameter	Description
Database Proxy	Enabled by default. After a proxy instance is created, you can use the proxy address to connect to your DB instance. **NOTE** – To create a proxy instance when buying a DB instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – You can also create proxy instances after buying a DB instance. For details, see **Creating a Proxy Instance for Read/Write Splitting**.
Proxy Mode	You can select **Read/Write** or **Read-only** as needed. – **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights. – **Read-only**: Write requests are not forwarded to any node. The primary node does not process write and read requests, and all read requests are forwarded to the selected read replicas based on read weights.

Parameter	Description
Proxy Instance Specifications	You can select the proxy instance specifications as needed.
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**.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can select the default parameter template, the high-performance parameter template, or a custom parameter template in the current region as required. For details about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters of a DB Instance**.
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 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.
Tag	Tags a DB instance. This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

- Required duration and quantity

Figure 3-15 Required duration and quantity

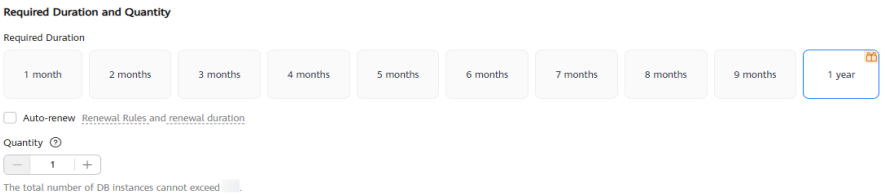


Table 3-13 Required duration and quantity

Parameter	Description
Required Duration	This parameter is available only for yearly/monthly instances. The system will automatically calculate the fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	– This parameter is available only for yearly/monthly instances and is not selected by default. – If you select this parameter, the auto-renew cycle is determined by the selected required duration.
Quantity	You can create DB 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 Refresh the DB instance list and view the status of the DB instance. If the status changes from **Creating** to **Available**, it has been created successfully. You can manage the DB instance on the **Instances** page.

The automated backup policy is enabled by default and cannot be disabled, and a full backup will be automatically created.

----End

Follow-up Operations

After an instance is created, you can enter a description for it. For details, see **Modifying a DB Instance Description**.

The default database port is **3306**. You can change it after an instance is created. To ensure data and instance security, you are advised to change the database port in a timely manner. For details, see **Changing a Database Port**.

APIs

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

3.3 Buying a Serverless DB Instance

Scenarios

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

This section describes how to create a serverless DB instance on the TaurusDB console.

Constraints

Serverless DB instances are only available in the following regions:

- CN North-Beijing4
- CN East-Shanghai1
- CN South-Guangzhou
- AP-Singapore
- AP-Bangkok
- ME-Riyadh
- CN-Hong Kong

Billing

For details, see **Serverless Billing**.

Prerequisites

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

Procedure

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

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

- Basic configuration

Figure 3-16 Basic configuration

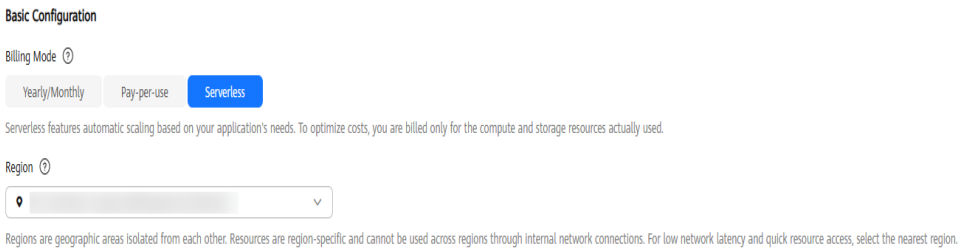


Table 3-14 Basic configuration

Parameter	Description
Billing Mode	Select **Serverless**.
Region	Region where an instance is deployed. **NOTE** You cannot change the region of an instance once it is purchased.

- Resource selection

Figure 3-17 Resource selection

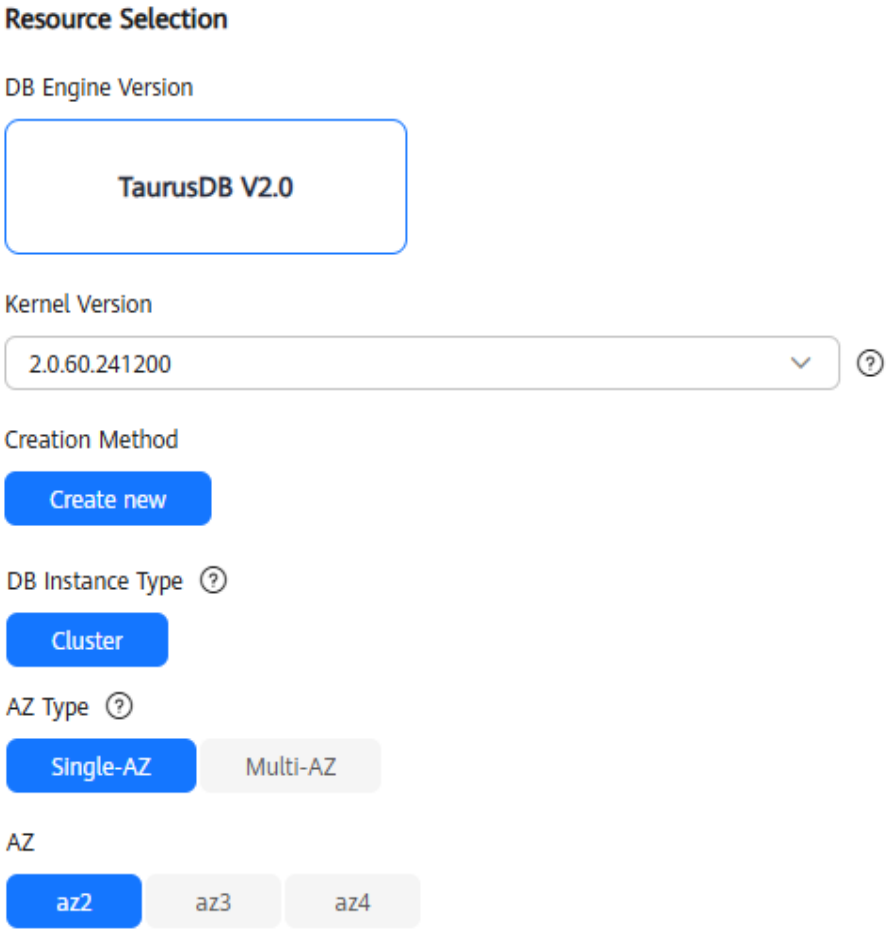


Table 3-15 Resource selection

Parameter	Description
DB Engine Version	Select **TaurusDB V2.0**.
Kernel Version	DB kernel version. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** – To specify the kernel version when buying an instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – To buy a multi-primary instance, select kernel version 2.0.45.230950 or 2.0.57.240900.
Creation Method	Select **Create new**.

Parameter	Description
DB Instance Type	Only cluster instances are supported. A cluster instance billed on a serverless basis can contain one primary node and up to seven read replicas. The primary node processes read and write requests, and the read replicas process only read requests. If the primary node becomes unavailable, 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.
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. – **Single-AZ**: The primary node and read replicas are deployed in the same AZ. – **Multi-AZ**: The primary node and read replicas are deployed in different AZs to achieve higher availability and reliability. It is suitable for workloads that require cross-AZ DR or are insensitive to cross-AZ latency.

- Instance options

Figure 3-18 Specifications and storage

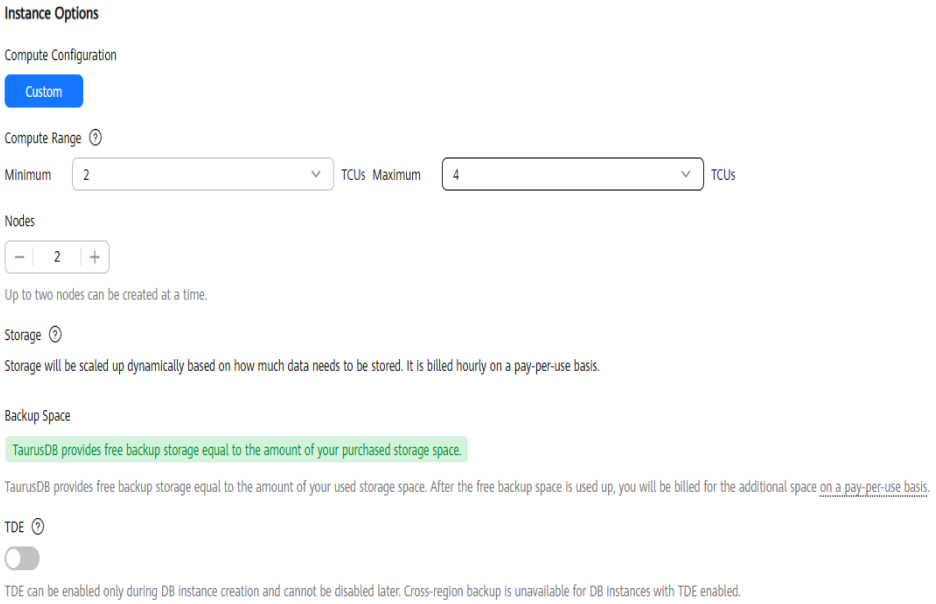


Table 3-16 Specifications and storage

Parameter	Description
Compute Configuration	Currently, only **Custom** is supported.
Compute Range	1 TCU is approximately equal to 1 vCPU and 2 GB of memory. Value range: 1 TCU to 32 TCUs
Nodes	Total number of one primary node and read replicas you created for the instance. You can create up to 8 nodes at a time.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
Backup Space	TaurusDB provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis.
TDE	Transparent Data Encryption (TDE) encrypts data files and backup files using certificates to implement real-time I/O encryption and decryption. This function effectively protects the security of databases and data files. After TDE is enabled, you need to select the cryptographic algorithm **AES256** or **SM4** as needed. **NOTE** – To use TDE, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – For details about TDE constraints, see **Enabling TDE for a DB Instance**.

Figure 3-19 Network

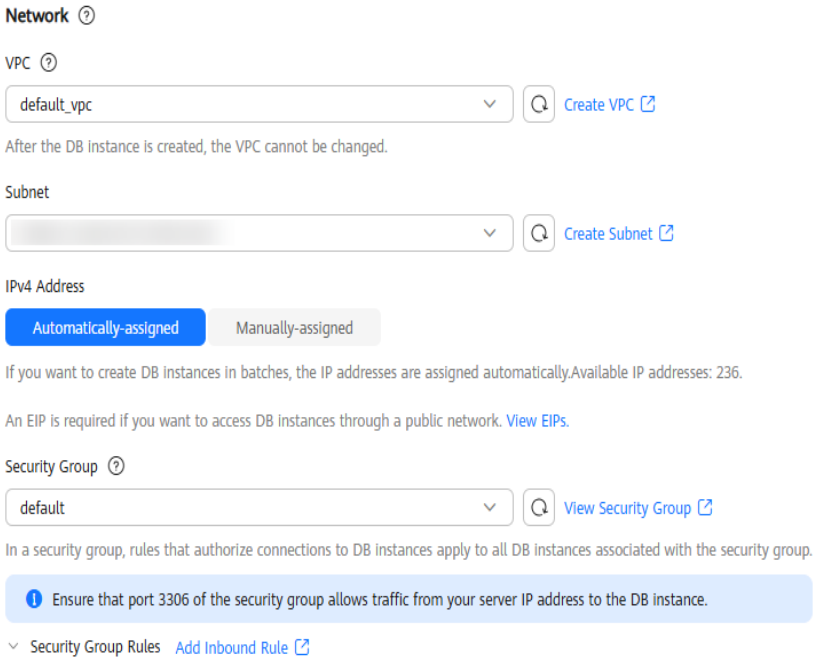
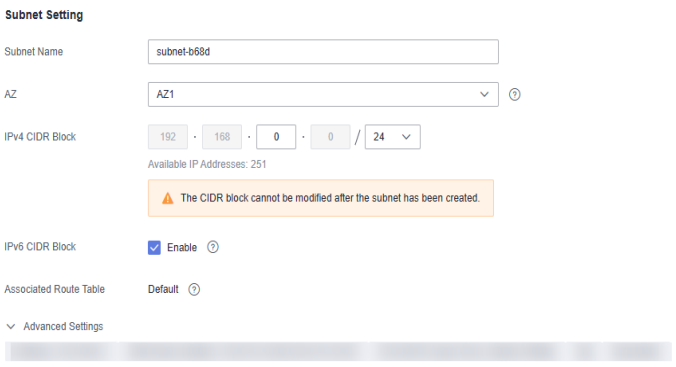


Table 3-17 Network

Parameter	Description
VPC	A dedicated virtual network where your instance is located. It isolates networks for different workloads to enhance security. TaurusDB allocates a default VPC (**default_vpc**) for your instance. You can also use an existing, new, or shared VPC. **After a TaurusDB instance is created, the VPC cannot be changed.** – To use an existing VPC, select an existing VPC under the current account from the drop-down list. – To use a new VPC, create a VPC first and then select the VPC from the drop-down list. For details about how to create a VPC, see **Creating a VPC and Subnet** in *Virtual Private Cloud User Guide*. – To use a shared VPC, select a VPC that another account shares with the current account from the drop-down list. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts, so you can easily configure and manage multiple accounts' resources at low costs. For more information about VPC subnet sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within an AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets where DB instances are located and cannot be disabled. TaurusDB supports both IPv4 and IPv6 networks. Instances using IPv4 and IPv6 cannot be in the same subnet. – IPv4 A private IPv4 address is automatically assigned when you create a DB instance. You can also enter an idle private IPv4 address within the subnet CIDR block. After the DB instance is created, the private IPv4 address can be changed. – IPv6 IPv6 addresses are used to deal with IPv4 address exhaustion. To enable IPv6, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. After IPv6 is enabled, if you want to create an instance using IPv4, you need to select an IPv4 subnet. You can create a DB instance that uses a private IPv6 address only when its specifications support IPv6. Instance specifications supporting IPv6 vary depending on regions and AZs. Whether instance specifications support IPv6 is displayed on the console after a region and an AZ are selected. IPv6 needs to be enabled for subnets where TaurusDB instances are located. If IPv6 is not enabled, enable it by following the instructions provided in **Creating a VPC and Subnet**. **Figure 3-20** Enabling IPv6 for a subnet 

Parameter	Description
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 DB instances. If no security group is available or has been created, TaurusDB allocates a security group to your DB instance by default. To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. If the port and protocol are not enabled for the selected security group, click **Add Inbound Rule** as prompted and complete the configuration in the displayed dialog box. For details, see **Configuring Security Group Rules**.

Figure 3-21 Setting an administrator password

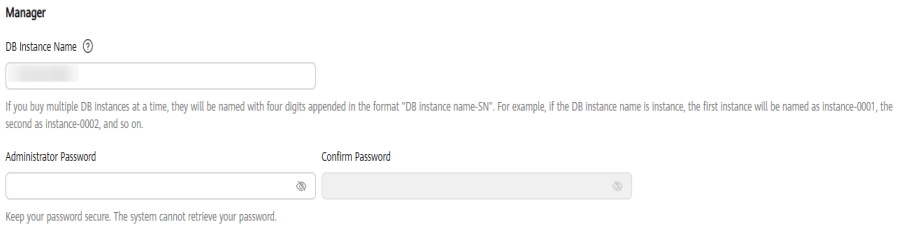


Table 3-18 Database configuration

Parameter	Description
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. – The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

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

- Advanced settings and required quantity

Figure 3-22 Advanced settings and required quantity

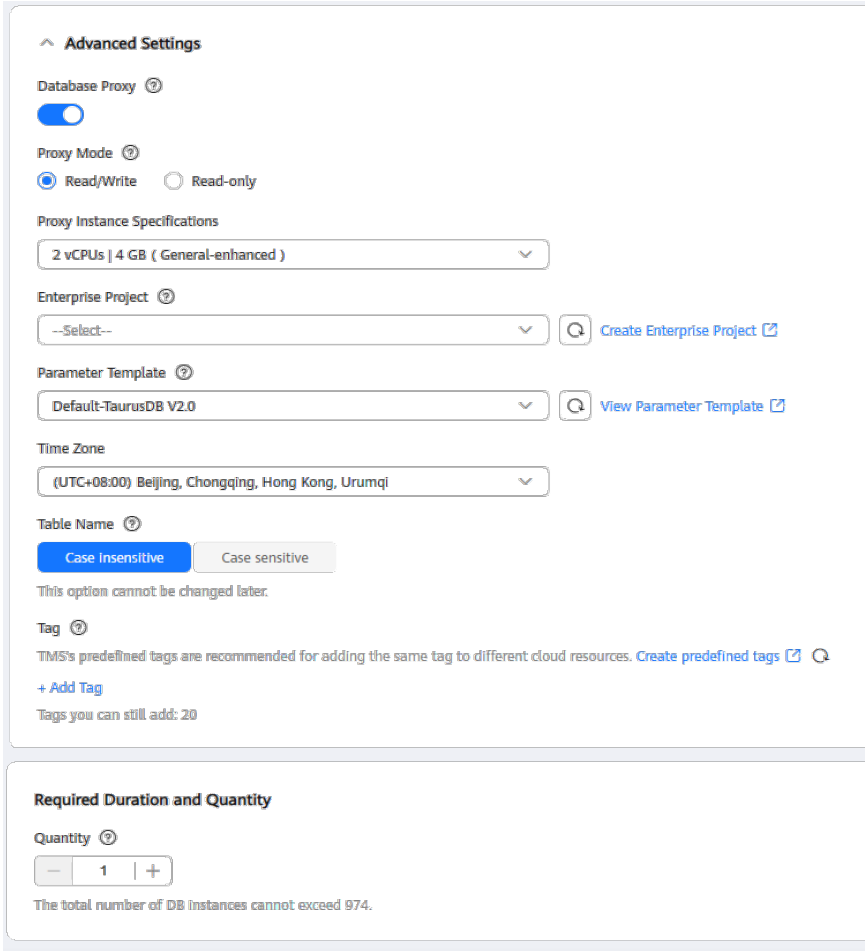


Table 3-19 Advanced settings

Parameter	Description
Database Proxy	Enabled by default. After a proxy instance is created, you can use the proxy address to connect to your DB instance. **NOTE** – To create a proxy instance when buying a DB instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. – You can also create proxy instances after buying a DB instance. For details, see **Creating a Proxy Instance for Read/Write Splitting**.
Proxy Mode	You can select **Read/Write** or **Read-only** as needed. – **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights. – **Read-only**: Write requests are not forwarded to any node. The primary node does not process write and read requests, and all read requests are forwarded to the selected read replicas based on read weights.

Parameter	Description
Proxy Instance Specifications	You can select the proxy instance specifications as needed.
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**.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can select the default parameter template, the high-performance parameter template, or a custom parameter template in the current region as required. For details about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters of a DB Instance**.
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 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.
Tag	Tags a DB instance. This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.
Quantity	You can buy DB instances in batches. The default value is **1**. The value ranges from 1 to 10.

Step 3 Confirm your specifications.

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

Step 4 Refresh the DB instance list and view the status of the DB instance. If the status changes from **Creating** to **Available**, it has been created successfully. You can manage the DB instance on the **Instances** page.

The automated backup policy is enabled by default and cannot be disabled, and a full backup will be automatically created.

----End

Follow-up Operations

After an instance is created, you can enter a description for it. For details, see **Modifying a DB Instance Description**.

The default database port is **3306**. You can change it after an instance is created. To ensure data and instance security, you are advised to change the database port in a timely manner. For details, see **Changing a Database Port**.

APIs

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

4 Connecting to a DB Instance

4.1 Connection Methods

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

Table 4-1 Connection methods

Connect Through	Connection Address	Description	Comments
DAS	Not required	DAS enables you to manage TaurusDB 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 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 your 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.
JDBC	Private IP address or EIP	JDBC is used to access TaurusDB instances.	-

4.2 Connecting to a DB Instance Through DAS (Recommended)

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

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

Prerequisites

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

Procedure

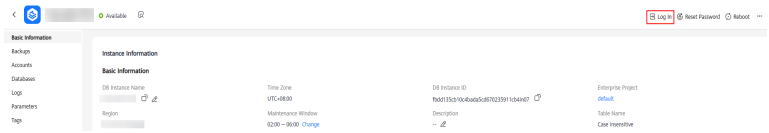
Step 1 **Log in to the management console.**

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Alternatively, on the **Instances** page, click the instance name. On the displayed **Basic Information** page, click **Log In** in the upper right corner.



Step 5 Enter the login username and password and click **Test Connection**.

Figure 4-1 Login page

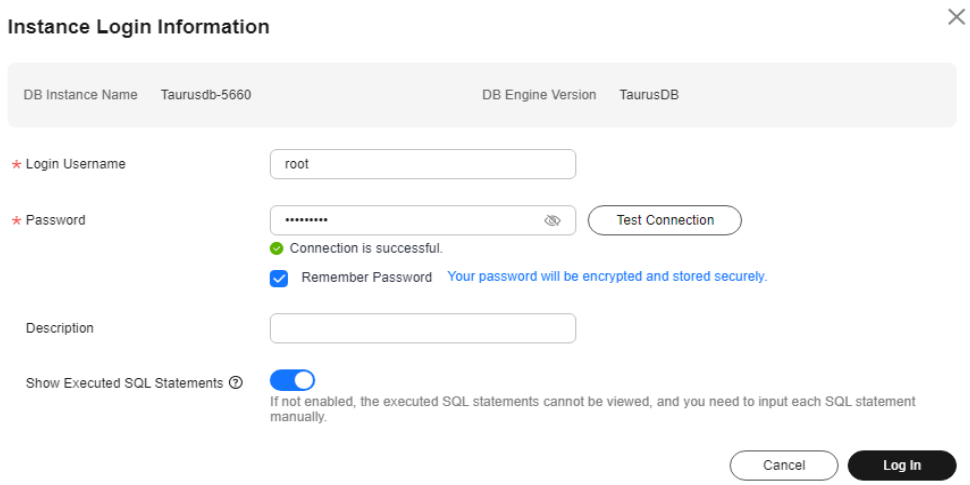


Table 4-2 Parameter description

Parameter	Description
Login Username	The default administrator is **root**.
Password	Enter the password you specified for the **root** user during instance creation. If you forget the password, reset it. For details, see **Resetting the Administrator Password**.
Show Executed SQL Statements	You are advised to enable **Show Executed SQL Statements**. If enabled, you can view the executed SQL statements under **SQL Operations** > **SQL History** and execute them again without entering the SQL statements.

Step 6 After the connection test is successful, click **Log In**. Then you can access and manage your databases.

----End

Other Methods for Connecting to a DB Instance

- **Connecting to a DB Instance Through mysql (the MySQL Command-Line Client)**
- **Connecting to a DB Instance Through MySQL-Front**
- **Connecting to a DB Instance Through JDBC**

What Can I Do If the DAS Console Is Not Displayed After I Click Log In in the Operation Column of an Instance on the Instances Page?

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

What Should I Do If I Cannot Connect to My DB Instance Due to Insufficient Permissions?

1. Error message: You do not have the required permission. The policy does not allow action das:connections:xxx.

   Error cause: Your account does not have the DAS FullAccess permission.

   Solution: Add the DAS FullAccess permission by referring to **Creating a User and Granting Permissions**.
2. Error message: You do not have the permission to perform this operation. Contact your administrator to request the required permission.

   Error cause: Your account does not have the DAS FullAccess permission.

   Solution: Add the DAS FullAccess permission by referring to **Creating a User and Granting Permissions**.
3. Error message: Your current account only has the read-only permission and cannot perform this operation. To ensure that you can use DAS smoothly, add the DAS Administrator permission.

   Error cause: Your account does not have the DAS FullAccess permission.

   Solution: Add the DAS FullAccess permission by referring to **Creating a User and Granting Permissions**.

What Should I Do If I Can't Connect to My TaurusDB Instance?

1. Error message: **Access denied for user 'user_name'@'100.xxx.xx.xx' (using password: YES).**

   a. Error cause: The username or password of a TaurusDB database is incorrect.

      Solution: Check whether the username and password are correct. If you are not sure, log in to the GaussDB console to view the username and reset the password.

      NOTICE

      Changing the password may affect services.

      If the username and password are correct, log in to the database using a client or CLI tool and run **select * from mysql.user where user =**

'user_name' to view the account. Make sure that the DAS CIDR block is within the CIDR block of the user. **user_name @ %** and **user_name @100.%** are two different users whose passwords and permissions are independent. Make sure to enter the password of **user user_name @100.%**.

b. Error cause: The IP address of the DAS server is not in the whitelist of the login user.

Solution: Log in to the database using the client or CLI tool, and create a user that can be used to access the database through DAS.

```
create user 'user_name'@'100.%' identified by 'password';
grant all privileges on *.* to 'user_name'@'100.%';
```

NOTE

1. Ensure that the IP address of the DAS server is in a CIDR block starting with 100. Add the IP address to the whitelist of the login user.
2. Grant permissions to user **user_name@100.%** based on service requirements.

2. Error message: **Trying to connect with ssl, but ssl not enabled in the server**

Error cause: The SSL function is not enabled on the server.

Solution: Run the following statement to check an SSL user is used. If yes, enable SSL on the TaurusDB instance details page. The user is an SSL user if the **ssl_type** field has a value.

```
select user, host, ssl_type from mysql.user where user = 'user_name';
```

3. Error message: **Client does not support authentication protocol requested by server.** plugin type was = 'sha256_password'

Error cause: DAS does not allow you to connect to the database whose password is encrypted with SHA-256.

Solution: Execute the following SQL statements to change the password encryption method to mysql_native_password.

```
alter user 'user_name'@'%' identified with mysql_native_password by 'password';
```

4. Error message: **Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.**

Error cause: The network between the DAS server and the instance is disconnected.

Solution: Contact technical support.

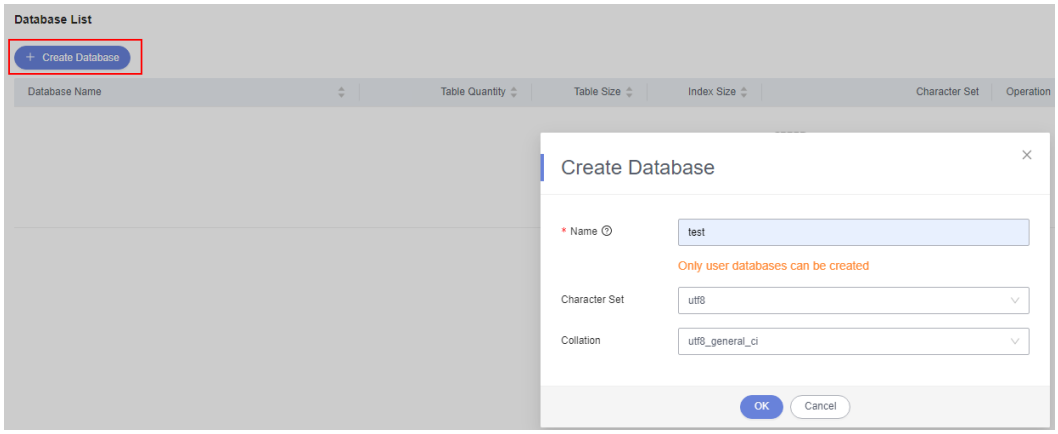
Follow-up Operations

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

Step 1 Create a database.

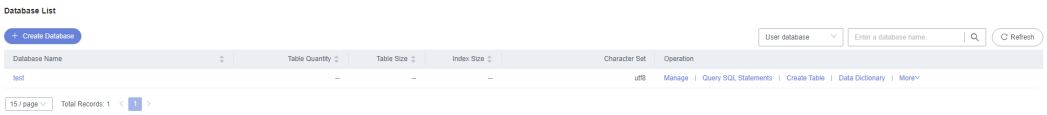
After logging in to a TaurusDB instance, click **Create Database** on the home page, enter database information, and click **OK**.

Figure 4-2 Creating a database



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

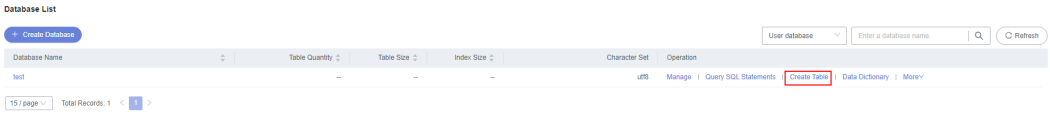
Figure 4-3 Viewing the created database



Step 2 Create a table.

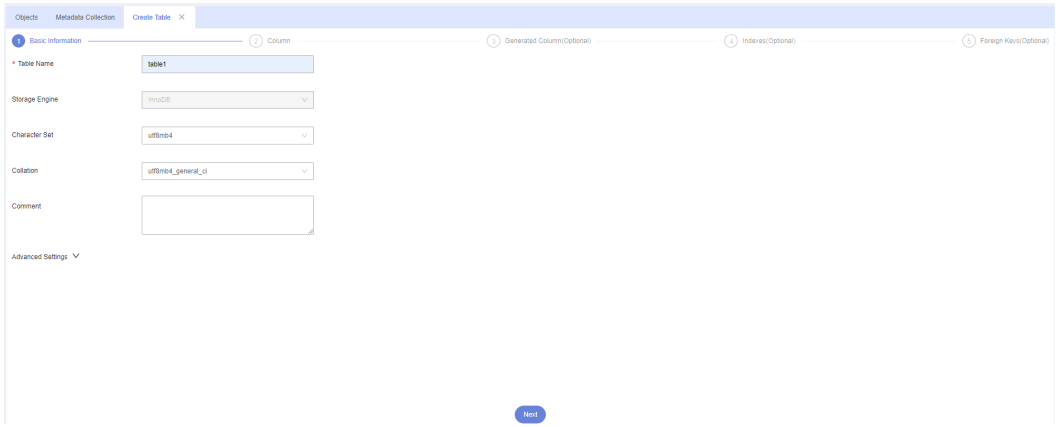
Locate the database and click **Create Table** in the **Operation** column.

Figure 4-4 Creating a table



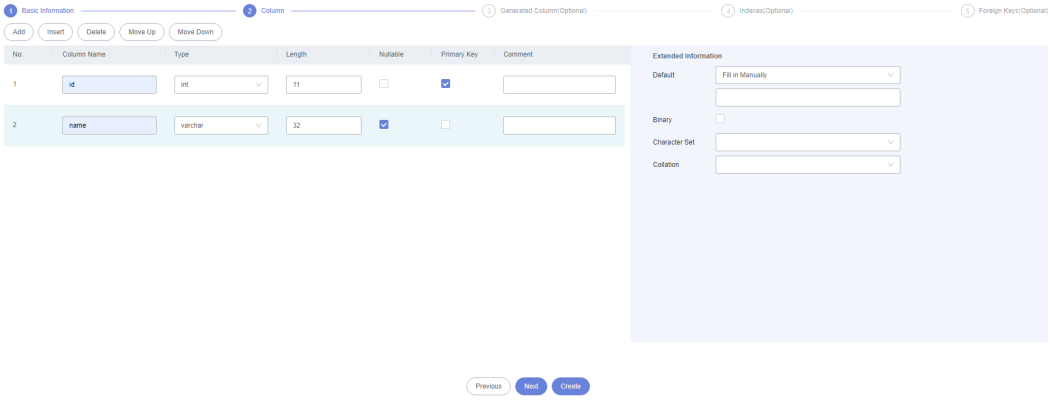
On the **Basic Information** tab, set the required parameters.

Figure 4-5 Entering basic table information



Click **Next** and enter column information.

Figure 4-6 Entering column information

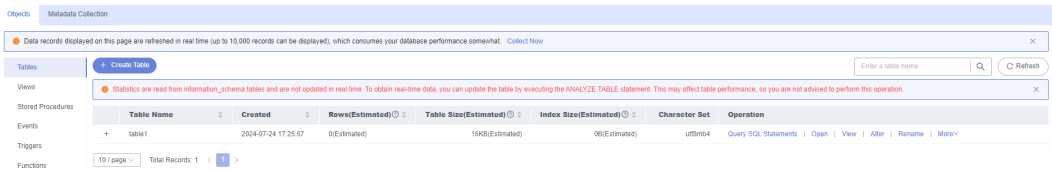


Click **Create**. In the SQL preview window, view the SQL statements for creating a table and click **Execute**.

Figure 4-7 Previewing the SQL statements for creating a table

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

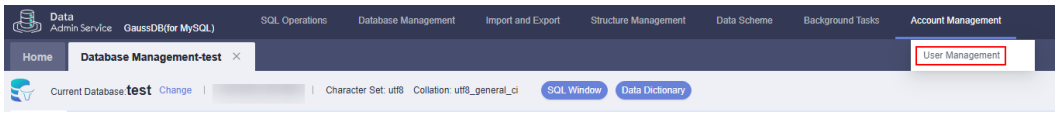
Figure 4-8 Viewing the created table



Step 3 Create a user and grant all permissions on the database created in **Step 1** to the user.

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

Figure 4-9 User management



Click **Create User** and enter user information and authorization information.

Figure 4-10 Creating a user

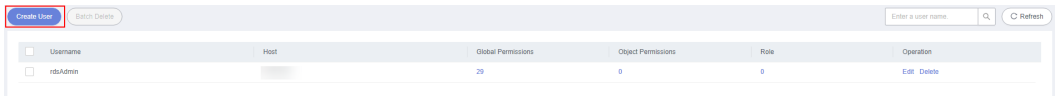
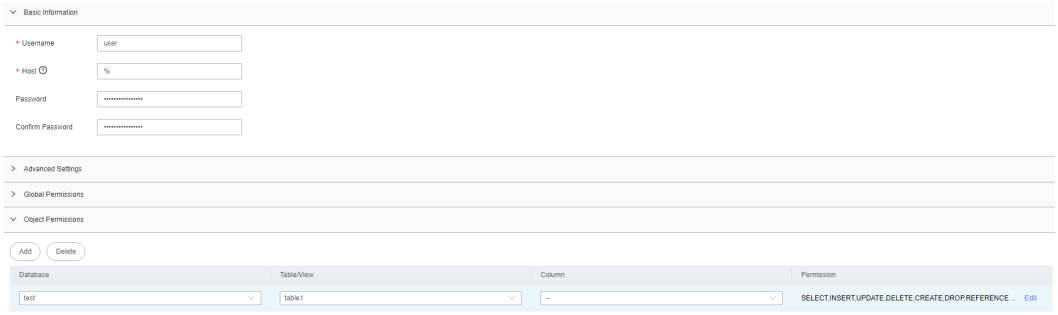


Figure 4-11 Entering user information and authorization information



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

Figure 4-12 Previewing the SQL statements for creating a user

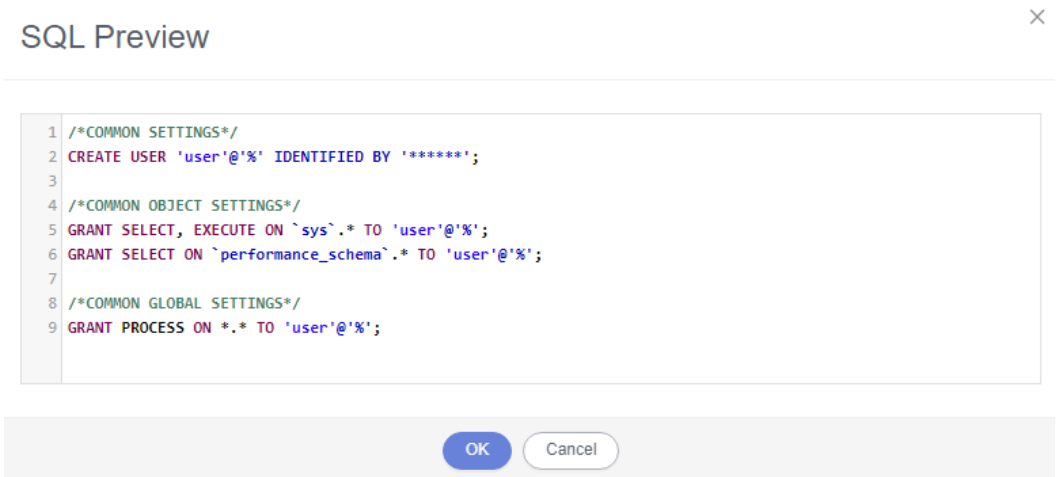
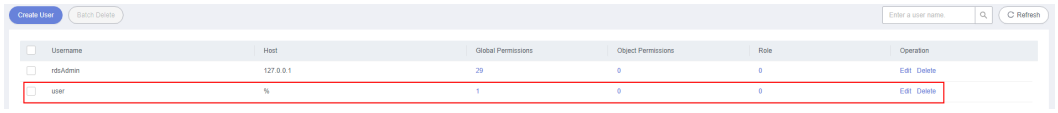


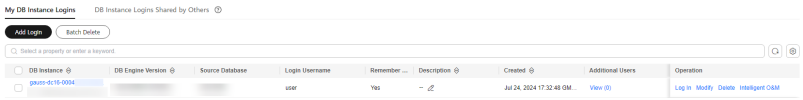
Figure 4-13 Viewing the created user



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

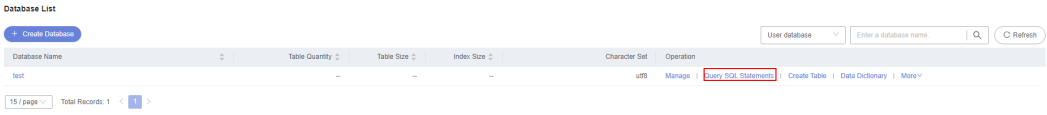
On the DAS development tool page, add a database login as user **user**. Click **Log In** in the **Operation** column to log in to the TaurusDB instance.

Figure 4-14 Adding a login as **user**



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

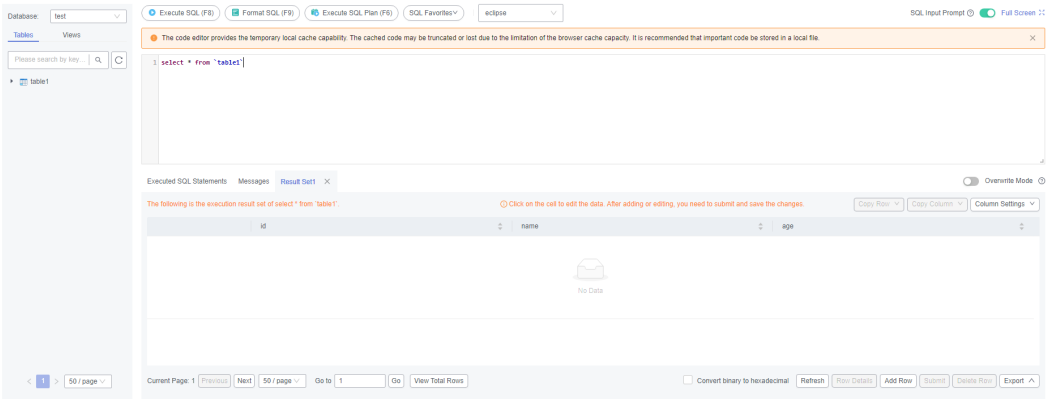
Figure 4-15 Accessing the SQL execution window



Run the following SQL statement in the SQL input box to query data in **table1**:

SELECT * FROM table1;

Figure 4-16 Viewing table data



There is no data in **table1**.

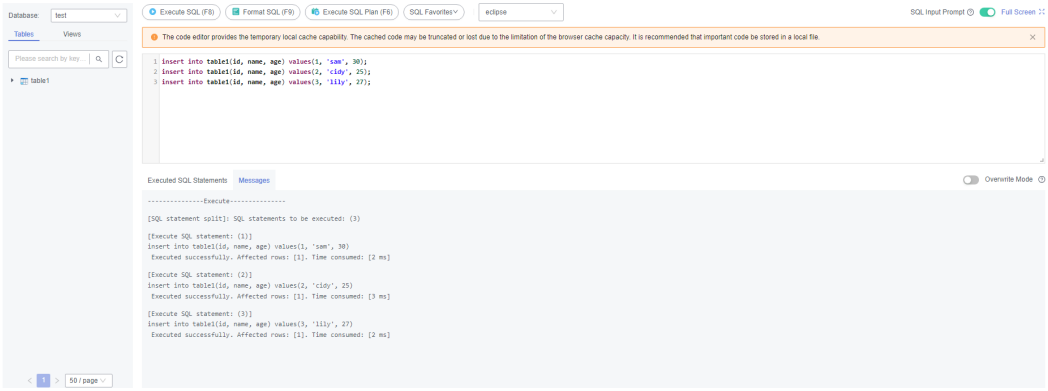
Run the following SQL statements to write several data records to **table1**:

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

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

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

Figure 4-17 Writing data to the table

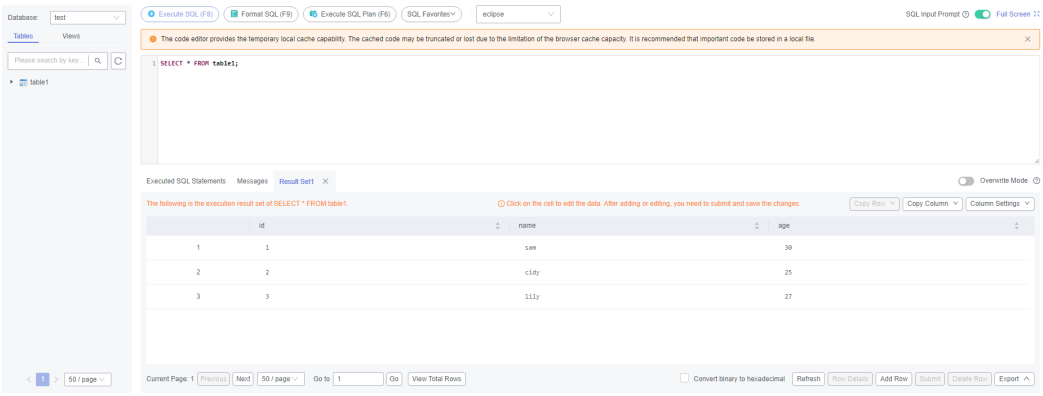


Data has been written into the table.

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

SELECT * FROM table1;

Figure 4-18 Verifying the written data



----End

4.3 Connecting to a DB Instance Through mysql (the MySQL Command-Line Client)

4.3.1 Connecting to a DB Instance over a Private Network

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

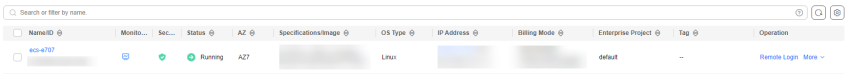
This section describes how to connect a Linux ECS to a DB instance with SSL enabled through a private IP address. SSL encrypts connections to the DB instance, making data more secure.

Step 1: Buy an ECS

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

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

Figure 4-19 ECS



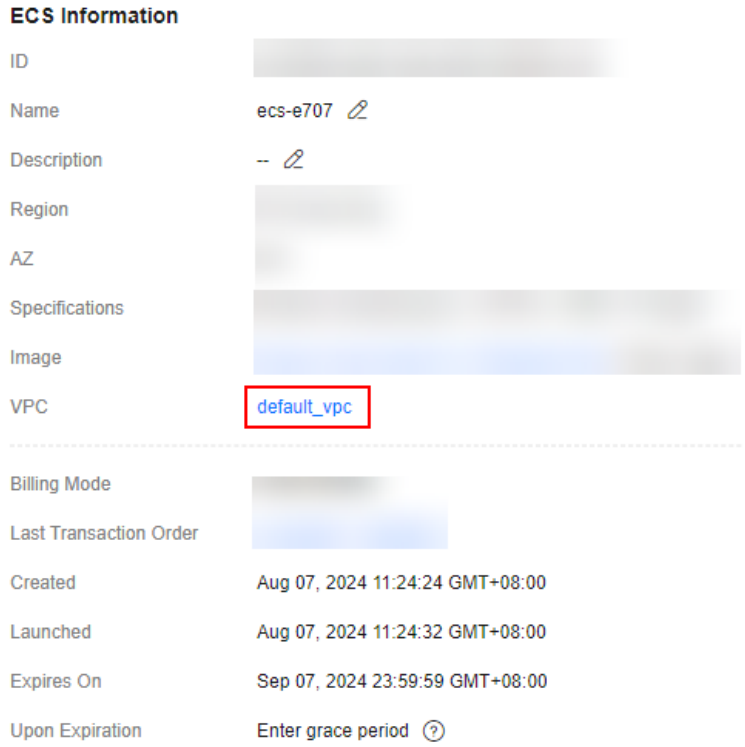
2. Buy an ECS and select Linux (for example, CentOS) as its OS.

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

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

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

   Figure 4-20 ECS information



4. On the **Basic Information** page of the DB instance, view the region and VPC of the DB instance.
5. Check whether the ECS and DB instance are in the same region and VPC.
   - If they are in the same region and VPC, go to **Step 2: Test Connectivity and Install the mysql Client**.
   - If they are in different regions, buy another instance. The ECS and DB instance in different regions cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.

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

Step 2: Test Connectivity and Install the mysql Client

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using an SSH Password** in *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the TaurusDB console, click the instance name to go to the **Basic Information** page.
3. In the **Network Information** area, obtain the private IP address and database port.

   Figure 4-21 Viewing the private IP address and database port

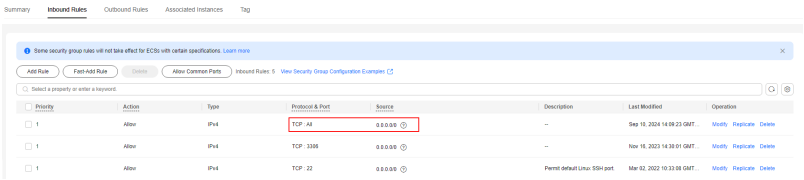
   

4. On the ECS, check whether the private IP address and database port of the DB instance can be connected.

   ```
   telnet 192.168.6.144 3306
   ```

   - If yes, network connectivity is normal.
   - If no, check the security group rules.
     - If in the security group associated with the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add the private IP address and port of the DB instance to the outbound rules.
     - If in the security group of the DB instance, there is no inbound rule with **Source** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add the private IP address and port of the ECS to the inbound rules. For details, see **Configuring Security Group Rules**.

       Figure 4-22 Security group of a DB instance

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

   Find the **corresponding version**, for example, **mysql-community-client-8.0.21-1.el6.x86_64.rpm**, and download the installation package. You are advised to use a mysql client running a version later than that of the DB instance.

6. Upload the installation package to the ECS.

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

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

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


CAUTION

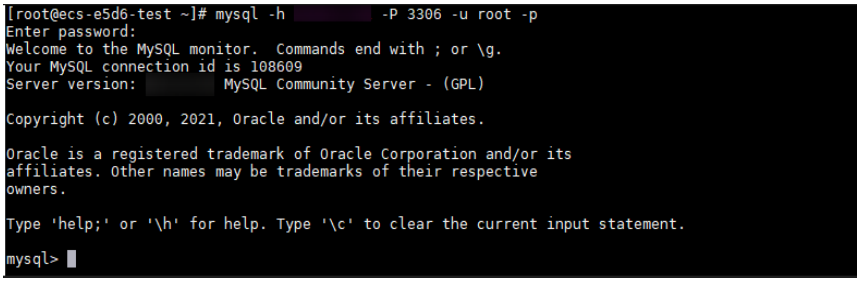
- If any conflicts occur during the installation, add the **replacefiles** parameter to the command and install the client again.
  ```
  rpm -ivh --replacefiles mysql-community-client-8.0.21-1.el6.x86_64.rpm
  ```
- If a message is displayed prompting you to install a dependent package during the installation, add the **nodeps** parameter to the command and install the client again.
  ```
  rpm -ivh --nodeps mysql-community-client-8.0.21-1.el6.x86_64.rpm
  ```

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

1. On the **Instances** page of the TaurusDB console, click the instance name to go to the **Basic Information** page.
2. In the **Instance Information** area, check whether SSL is enabled.
   - If yes, go to **3**.
   - If no, click  . In the displayed dialog box, click **Yes** to enable SSL. Then, go to **3**.
3. Click  under **SSL** to download **Certificate Download.zip**, and obtain the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.
4. Upload **ca.pem** to the ECS.
5. Run the following command on the ECS to connect to the DB instance:
   ```
   mysql -h host -P port -u userName -p --ssl-ca=caName
   ```
   Example:
   ```
   mysql -h 192.168.0.79 -P 3306 -u root -p --ssl-ca=ca.pem
   ```

   Table 4-3 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:
   ```
   
   ```
   [root@ecs-e5d6-test ~]# mysql -h [redacted] -P 3306 -u root -p
   Enter password:
   Welcome to the MySQL monitor.  Commands end with ; or \g.
   Your MySQL connection id is 108609
   Server version: [redacted] MySQL Community Server - (GPL)

   Copyright (c) 2000, 2021, Oracle and/or its affiliates.

   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>
   ```


FAQs

What Should I Do If I Can't Connect to My TaurusDB Instance?

4.3.2 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 first and connect the ECS to the DB instance through the EIP.

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

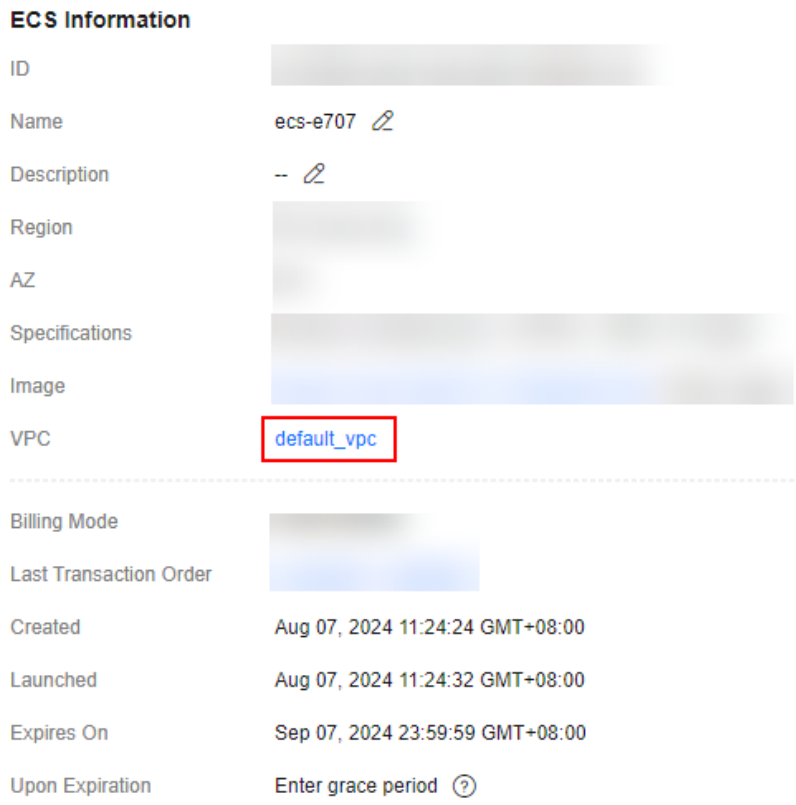
Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Linux ECS, go to **3**.
   - If there is a Windows ECS, see **Connecting to a DB Instance Through MySQL-Front**.
   - If no ECS is available, go to **2**.
2. Buy an ECS and select Linux (for example, CentOS) as its OS.

   To download the mysql client to the ECS, bind an EIP to the ECS.

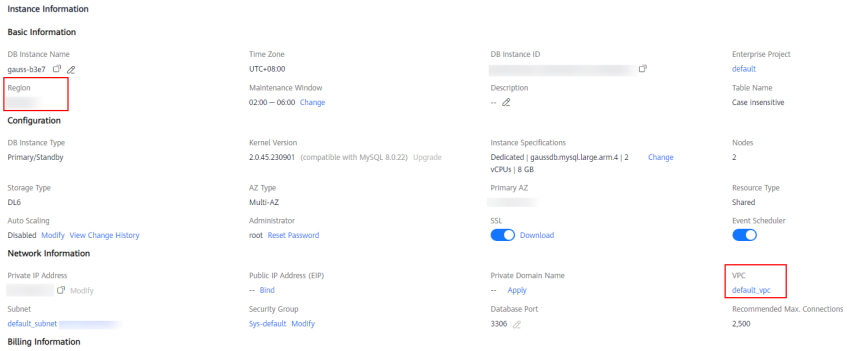
   For details about how to purchase a Linux ECS, see **Purchasing an ECS** in *Elastic Cloud Server Getting Started*.
3. On the **ECS Information** page, view the region and VPC of the ECS.

   Figure 4-23 ECS information



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

Figure 4-24 DB instance information



Step 2: Test Connectivity and Install the mysql Client

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using an SSH Password** in *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the TaurusDB console, click the instance name to go to the **Basic Information** page.
3. In the **Network Information** area, obtain the EIP and database port.

Figure 4-25 EIP and database port

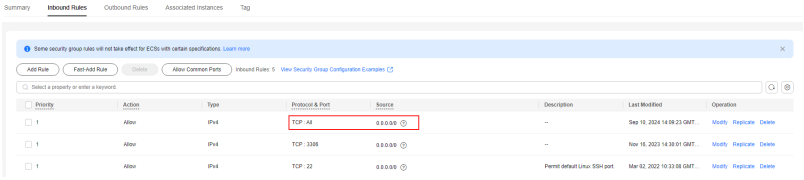
If no EIP has been bound to the DB instance, see **Binding or Unbinding an EIP**.

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

   telnet *EIP 3306*

   - If yes, network connectivity is normal.
   - If no, check the security group rules.
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add the EIP and port of the DB instance to the outbound rules.
     - If in the security group of the DB instance, there is no inbound rule with **Source** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add the private IP address and port of the ECS to the inbound rules. For details, see **Configuring Security Group Rules**.

Figure 4-26 Security group of a DB instance



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

   Find the **corresponding version**, for example, mysql-community-client-8.0.21-1.el6.x86_64.rpm, and download the installation package. You are advised to use a mysql client running a version later than that of the DB instance.

6. Upload the installation package to the ECS.

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

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

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

CAUTION

- 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 dependent package during the installation, add the **nodeps** parameter to the command and install the client again.

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

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

1. On the **Instances** page of the TaurusDB console, click the instance name to go to the **Basic Information** page.
2. In the **Instance Information** area, check whether SSL is enabled.
   - If yes, go to **3**.
   - If no, click . In the displayed dialog box, click **Yes** to enable SSL. Then, go to **3**.
3. Click under **SSL** to download **Certificate Download.zip**, and obtain the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.
4. Upload **ca.pem** to the ECS.
5. Run the following command on the ECS to connect to the DB instance:

   ```
   mysql -h host -P port -u userName -p --ssl-ca=caName
   ```

   Example:

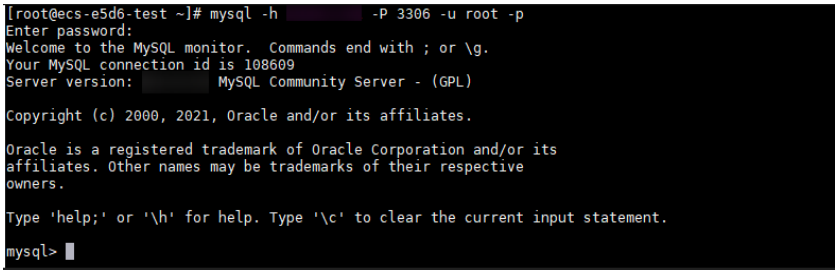
   ```
   mysql -h 172.16.0.31 -P 3306 -u root -p --ssl-ca=ca.pem
   ```


Table 4-4 Parameter description

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

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

```
Enter password:
```



FAQs

What Should I Do If I Can't Connect to My TaurusDB Instance?

4.4 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.

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, see **Connecting to a DB Instance Through mysql (the MySQL Command-Line Client)**.
- If there is a Windows ECS, go to **Step 3**.
- If no ECS is available, go to **Step 2**.

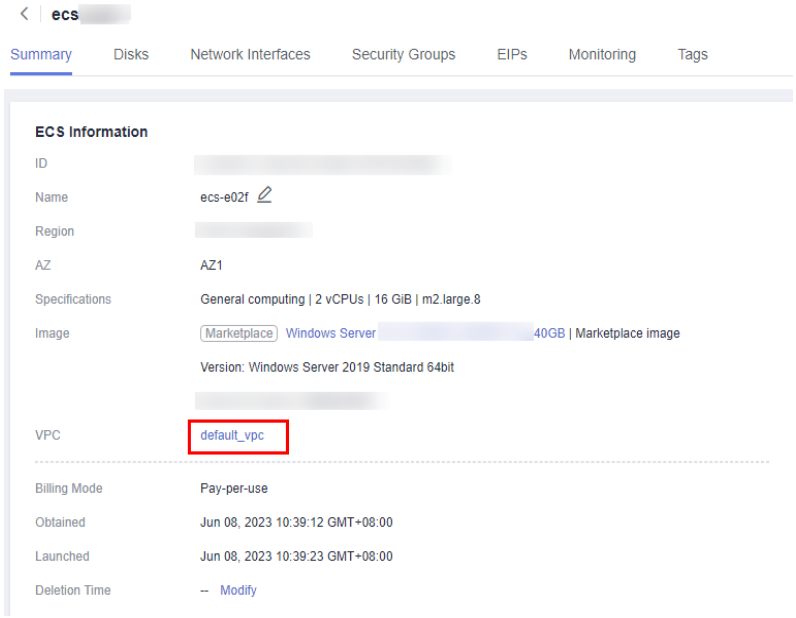
Step 2 Buy an ECS and select Windows as its OS.

To download the mysql client to the ECS, bind an EIP to the ECS.

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

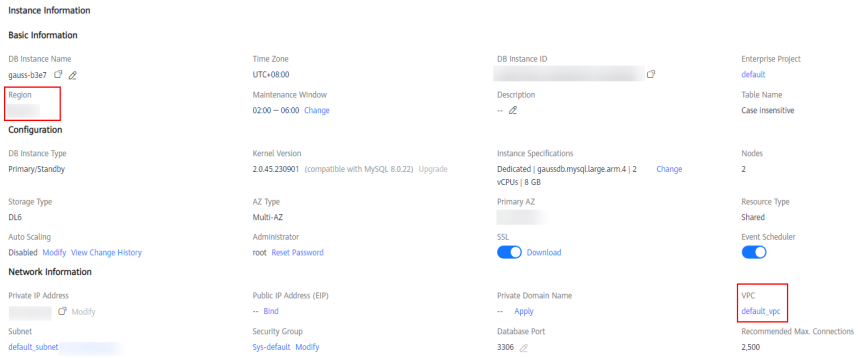
Step 3 On the **ECS Information** page, view the region and VPC of the ECS.

Figure 4-27 Viewing ECS information



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

Figure 4-28 Viewing the region and VPC of the DB instance



----End

Step 2: Bind an EIP to a DB Instance

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

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

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

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

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

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

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

CAUTION

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

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

----End

Step 3 Query the EIP of the DB Instance to Be Connected

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

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

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

Figure 4-29 Viewing the EIP and database port

----End

Step 4: Test Connectivity and Install MySQL-Front

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

```
telnet EIP port
```

Example:

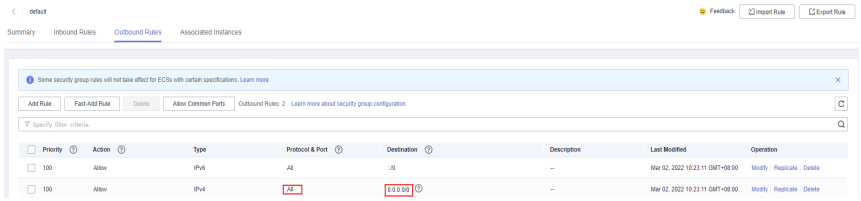
```
telnet 192.168.0.16 3306
```


NOTE

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

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

    Figure 4-30 Configuring rules of an ECS security group

  - 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 the EIP and port of the ECS to the inbound rules. For details, see **Configuring Security Group Rules**.

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

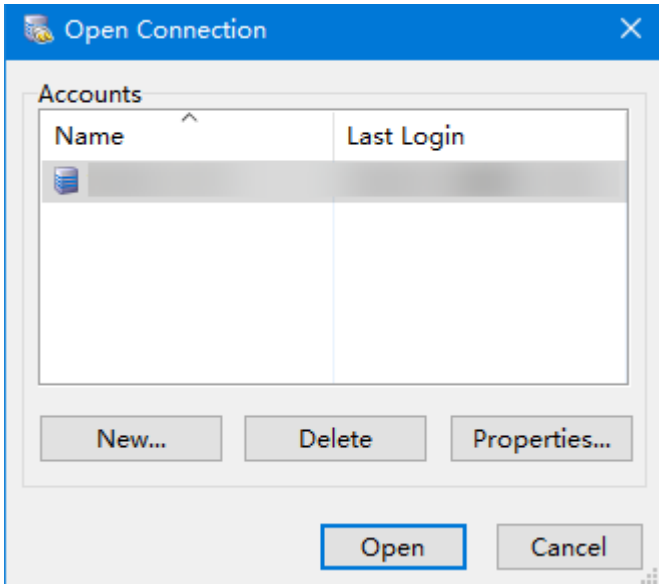
----End

Step 5: Connect to the DB Instance Using MySQL-Front

Step 1 Start MySQL-Front.

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

Figure 4-31 Creating a connection



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

Figure 4-32 Adding an account

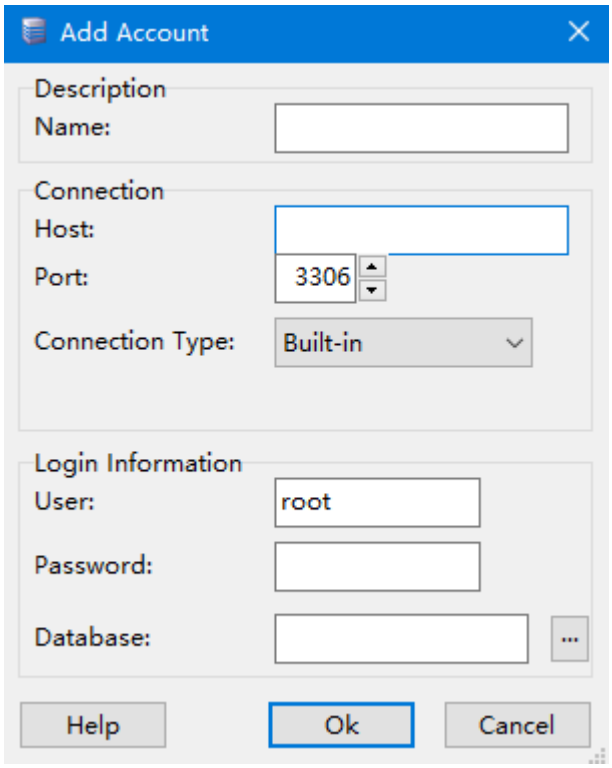
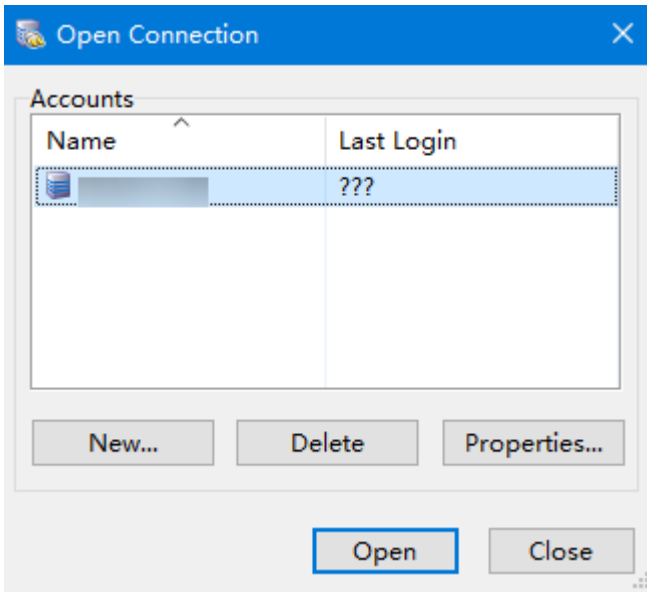


Table 4-5 Parameter description

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

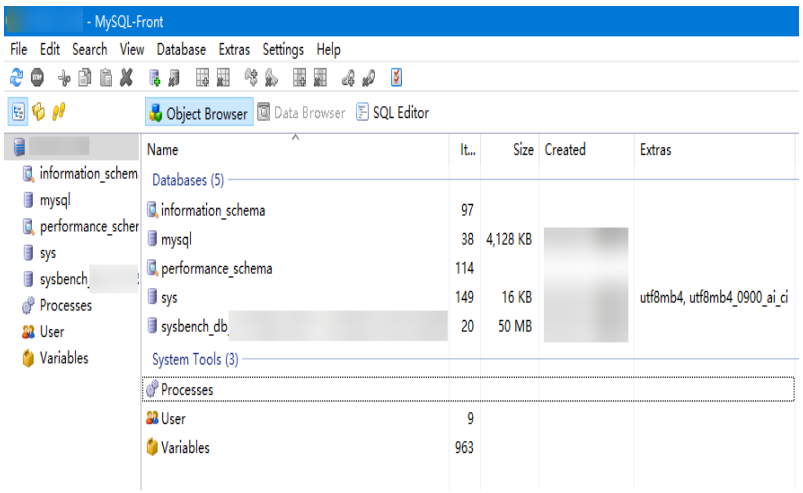
Step 4 In the displayed window, select the connection that you created and click **Open**.

Figure 4-33 Opening a connection



Step 5 Check whether the DB instance has been connected. If the connection information is correct, the DB instance has been connected.

Figure 4-34 Login succeeded



----End

FAQs

What Should I Do If I Can't Connect to My TaurusDB Instance?

4.5 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 is enabled for new TaurusDB instances. SSL encrypts connections to instances but prolongs connection response time and increases CPU usage. Before enabling SSL, evaluate the impact on service performance. For details about how to enable or disable SSL, see **Configuring SSL**.

Prerequisites

Familiarize yourself with:

- Computer basics
- Java programming language
- JDBC knowledge

Connection with the SSL Certificate

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

CAUTION

If the **ssl_type** value of a database user is **x509**, this method is unavailable.

To check the **ssl_type** value of the current user, run the following command:

```
select ssl_type from mysql.user where user = 'xxx';
```

Step 1 Download the CA certificate or certificate bundle.

1. On the **Instances** page, click the instance name to go to the **Basic Information** page.
2. Click **Download** under **SSL**.

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

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

Table 4-6 Parameter description

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

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

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

Step 3 Connect to your TaurusDB instance through JDBC.

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

Table 4-7 Parameter description

Parameter	Description
instance_ip	IP address of the DB instance. • If you are accessing the instance through ECS, *instance_ip* is the private IP address of the instance. You can view the private IP address in the **Network Information** area on the **Basic Information** page. • If you are accessing the instance through a public network, *instance_ip* is the EIP that has been bound to the instance. You can view the EIP in the **Network Information** area on the **Basic Information** page. • If you are accessing the instance through a proxy instance, *instance_ip* is the proxy address. You can view the proxy address on the **Database Proxy** page.
instance_port	Database port of the DB instance. The default port is **3306**. 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. For details about the relationship between **requireSSL** and **sslmode**, see **Table 4-8**.

Parameter	Description
value2	Value of **useSSL**, indicating whether the client uses SSL to connect to the server. It can be either of the following: • **true**: The client uses SSL to connect to the server. • **false**: The client does not use SSL to connect to the server. For details about the relationship between **useSSL** and **sslmode**, see **Table 4-8**.
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. For details about the relationship between **verifyServerCertificate** and **sslmode**, see **Table 4-8**.
truststore_file	Path for storing the truststore file configured in **Step 2**.
password	Password of the truststore file configured in **Step 2**.

Table 4-8 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 your TaurusDB instance through JDBC.

```
jdbc:mysql://instance_ip:instance_port/database_name?useSSL=false
```


Table 4-9 Parameter description

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

Code example (Java code for connecting to a 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
    ```

4.6 Connection Information Management

4.6.1 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and TaurusDB instances that are within the same VPC, have the same security requirements, and are mutually trusted. To ensure database security and reliability, you need to

configure security group rules to allow only specific IP addresses and ports to access the TaurusDB instances.

When you attempt to connect to a TaurusDB instance through a private network, check whether the ECS and TaurusDB 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 TaurusDB instance, respectively.
  - TaurusDB instance: Configure an **inbound rule** for the security group with which the TaurusDB 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 TaurusDB instance.

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

Precautions

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

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

NOTE

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

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

Procedure

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **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, click the security group name under **Security Group**.

Figure 4-35 Configuring security group rules

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

You can click ⊕ to add more inbound rules.

Figure 4-36 Adding inbound rules

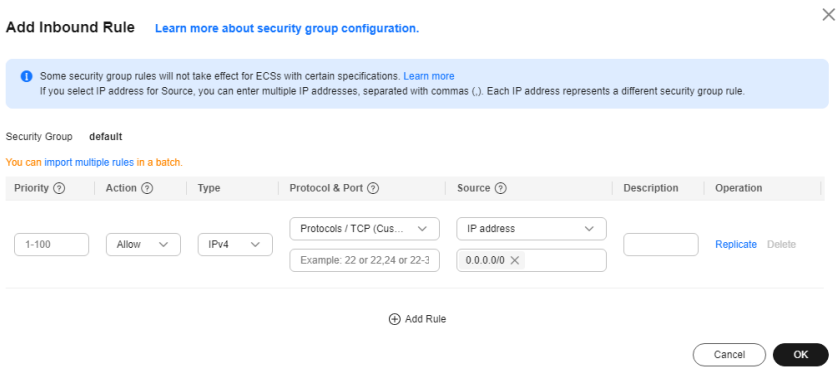


Table 4-10 Inbound rule parameter description

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

4.6.2 Binding or Unbinding an EIP

Scenarios

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

Precautions

- Binding EIPs to DB instances reduces the security of the DB instances. Exercise caution when performing this operation. To achieve a higher transmission rate and security level, you are advised to migrate your applications to the ECS that is in the same region as the TaurusDB instance.

- After an EIP billed on a pay-per-use basis is unbound from a TaurusDB instance, it is still billed. To save money, you can release the EIP or bind it to another DB instance.

Billing

- Traffic generated by the public network is billed. You can unbind the EIP from your DB instance when the EIP is no longer used.

Binding an EIP

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

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

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

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

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

----End

Unbinding an EIP

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 DB instance that has an EIP bound.

Step 5 In the **Network Information** area on the displayed **Basic Information** page, click **Unbind** under **Public IP Address (EIP)**.

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

Step 7 In the **Network Information** area on the **Basic Information** page, check the result.

You can also see the progress and result of unbinding an EIP from a DB instance on the **Task Center** page.

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

----End

APIs

- Binding an EIP
- Unbinding an EIP
- Querying an EIP

4.6.3 Changing a Database Port

You can change the database port of a TaurusDB instance.

Constraints

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

Procedure

Step 1 Log in to the management console.

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

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

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.

Step 6 In the displayed dialog box, enter a new database port and click **OK**.

----End

APIs

Changing a Database Port

4.6.4 Applying for and Changing a Private Domain Name

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

After a TaurusDB instance is created, you can apply for and change the private domain name as needed.

Constraints

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

Applying for a Private Domain Name

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **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 **Apply** under **Private Domain Name**.

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

----End

Changing a Private Domain Name

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

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

Figure 4-37 Modifying a private domain name

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

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

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

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

----End

4.6.5 Configuring and Changing a Private IP Address

Scenarios

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

Constraints

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

Configuring the Private IP Address of a DB Instance

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

Procedure

You can change the private IP address of an existing TaurusDB instance.

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

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

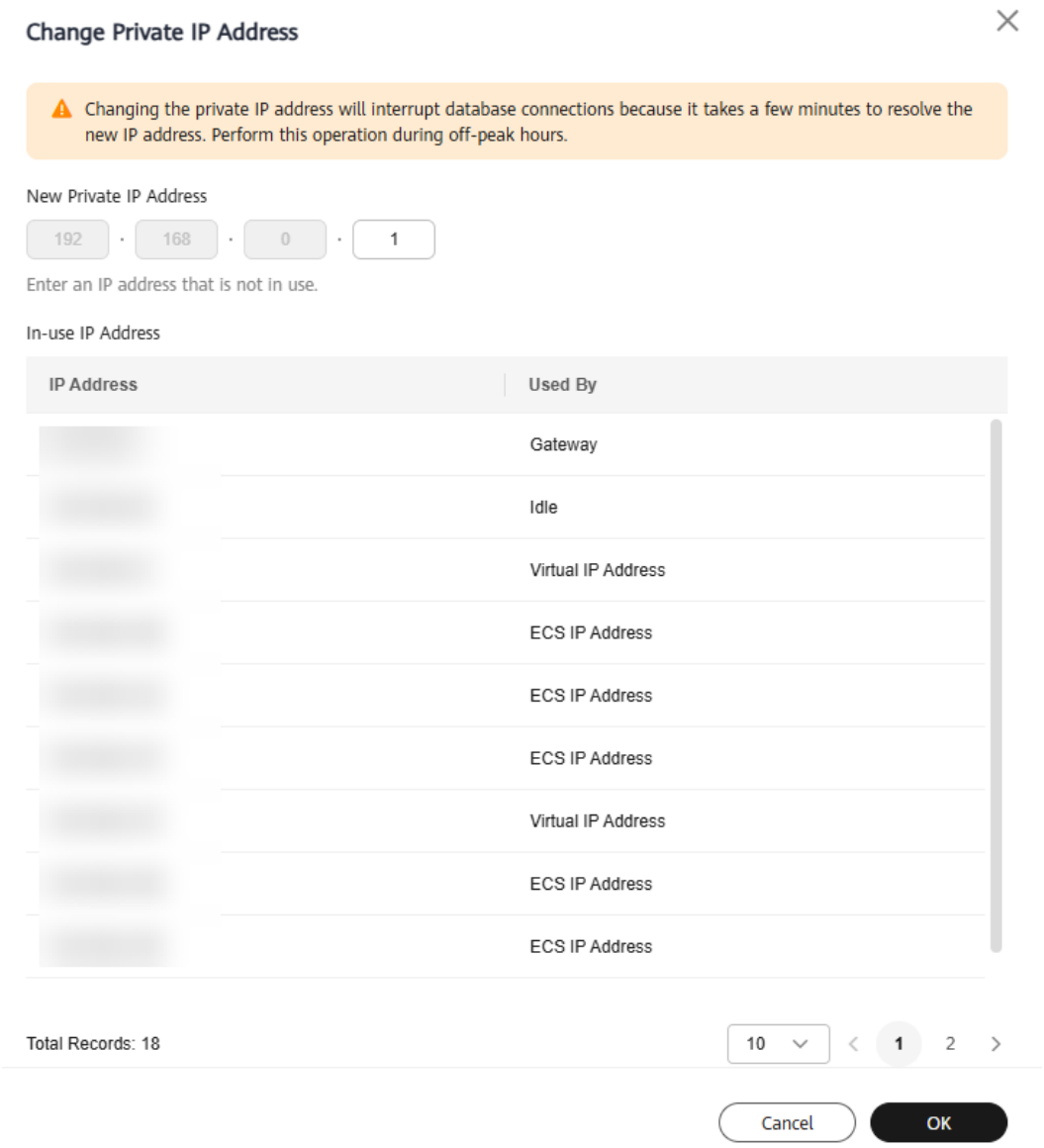
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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 **Modify** under **Private IP Address**.

Step 6 In the displayed dialog box, check the in-use IP addresses.

Figure 4-38 Changing a private IP address



Step 7 Enter an IP address that is not in use and click **OK**.

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

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

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

----End

APIs

Changing a Private IP Address

5 Database Usage

5.1 Usage Guidelines

5.1.1 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.

5.1.2 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 5-1 lists the new keywords and reserved words in TaurusDB.

Table 5-1 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. The absence of a primary key may cause slow execution of the primary database and replication latency.

  If an auto-increment primary key in MySQL reaches the maximum value of its data type, new insertions will fail. For details about how to handle this issue, see **What Should I Do If a MySQL Auto-Increment Primary Key Reaches the Maximum Value?**

  If a MySQL table contains a foreign key and has a foreign key relationship, the error message "ERROR 1451 (23000): Cannot delete or update parent row: a foreign key constraint fails" will be displayed when you attempt to delete the table. For details about how to handle this issue, see **Failed to Delete a Table with a Foreign Key**.
- 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 ('').
- Do not use partitioned tables. If necessary, use multiple independent tables instead. Partitioned tables have the following disadvantages:
  - 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.
  - Partitioned 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.

  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.

5.1.3 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.

  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.

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

  CAUTION

  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.

  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.

  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 neither **Using filesort** nor **Using temporary** is 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. For details about the parameters of an execution plan, see **Table 5-2**.

Table 5-2 EXPLAIN parameters

Parameter	Description
type	Indicates the scan type. • **ALL**: full table scan • **index**: full index scan • **range**: index range scan • **ref**: index lookup using a non-unique key • **eq_ref**: index lookup using a unique key (primary key or unique index) • **const**: MySQL can locate the unique row of data when the query uses index lookup using a primary key or unique index. • **system**: The table has only one row. • **NULL**: The query can be completed without accessing the table.
possible_keys	Indicates the indexes from which MySQL can choose to find the rows in this table. If an index exists on a column involved in a query, the index is listed but may not be used by the query.
key	Indicates the key (index) that MySQL actually decided to use. The key is NULL if no index was chosen. 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.

Parameter	Description
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.
  - For a table with tens of millions or hundreds of millions of data records, create indexes efficiently for it by referring to **How Do I Write Data to or Create Indexes for an Ultra-large Table?**

5.1.4 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 BETWEEN 1 AND 5;**.

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.

  The conversion rules are as follows:

  a. 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.
  b. If both parameters are character strings, they are compared as character strings.
  c. If both parameters are integers, they are compared as integers.
  d. When one parameter is a hexadecimal value and the other parameter is a non-digit value, they are compared as binary strings.
  e. If one parameter is a TIMESTAMP or DATETIME value and the other parameter is a CONSTANT value, they are compared as TIMESTAMP values.
  f. 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.
  g. In other cases, both parameters are compared as FLOATING POINT values.
  h. 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.

5.2 Database Management

5.2.1 Creating a Database

Scenarios

After a 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.

Creating a Database Through TaurusDB

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 **Databases**.

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

Figure 5-1 Creating a database

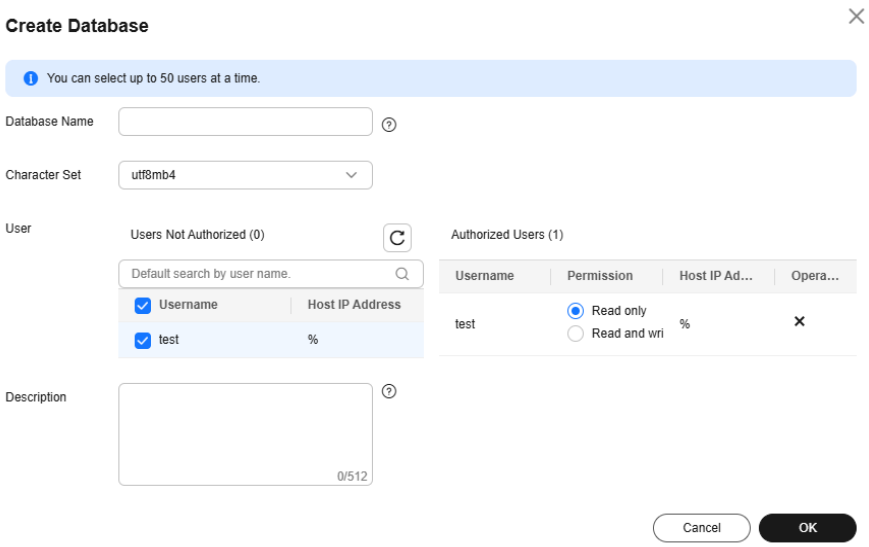


Table 5-3 Parameter description

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

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

----End

Creating a Database Through DAS

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

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

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

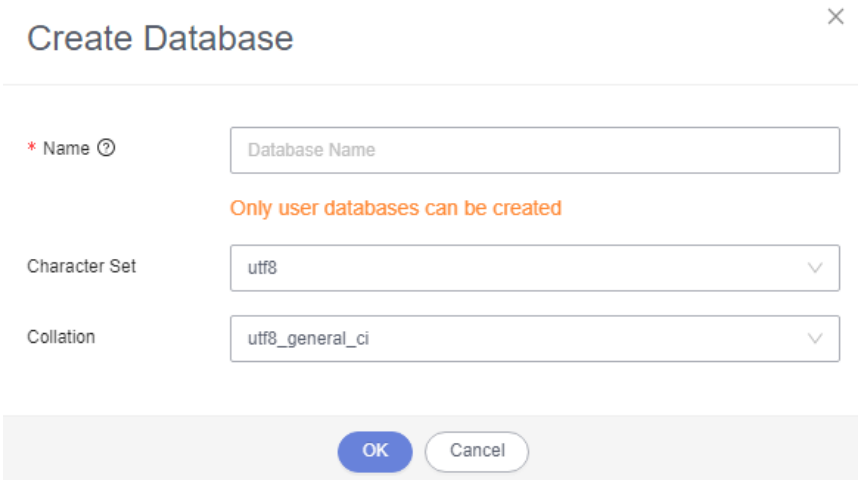
Step 4 On the **Instances** page, locate an instance and click **Log In** in the **Operation** column.

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

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

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

  Figure 5-2 Creating a database

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

  ```
  create database <database-name>;
  ```

----End

APIs

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

5.2.2 Deleting a Database

Scenarios

You can delete databases you have created.

Constraints

- Deleted databases cannot be recovered. Exercise caution when deleting a database.

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

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

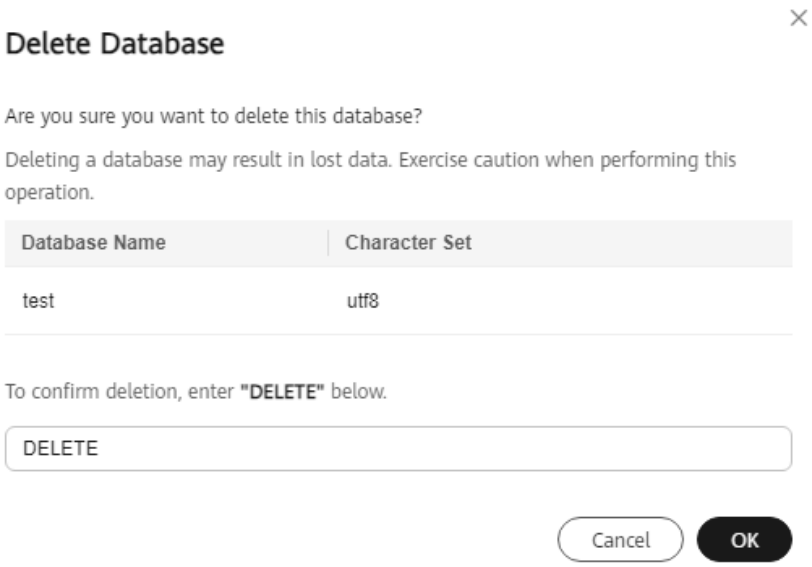
Step 6 On the displayed page, locate a database and click **Delete** in the **Operation** column.

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

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

Step 8 In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

Figure 5-3 Deleting a database



Step 9 Check that the database is no longer displayed on the **Databases** page.

----End

APIs

- Deleting a Database

- **Creating a Database**

5.3 Account Management (Non-Administrator)

5.3.1 Creating an Account

Scenarios

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

You can create an account through TaurusDB or DAS:

- **TaurusDB**: TaurusDB is easy to use. There are no special commands to remember.
- **DAS**: DAS is a powerful platform that offers more flexibility, but you need to be familiar with the creation commands. The process requires a bit more expertise.

Account Types

Table 5-4 Account description

Account Type	Description
Administrator account **root**	Only the administrator account **root** is provided on the instance creation page. For details about the supported permissions, see **Constraints**. **NOTE** Running **revoke**, **drop user**, or **rename user** on **root** may cause service interruption. Exercise caution when running any of these statements.

Account Type	Description
System accounts	To provide O&M services, the system automatically creates system accounts when you create TaurusDB instances. These system accounts are unavailable to you. • rdsAdmin: a management account with superuser permissions, which is used to query and modify instance information, rectify faults, migrate data, and restore data. • rdsRepl: a replication account, which is used to synchronize data from the primary node to read replicas. • rdsBackup: a backup account, which is used to back up data in the background. • rdsMetric: a metric monitoring account, which is used by watchdog to collect database status data. • rdsProxy: a database proxy account, which is used for authentication when the database is connected through the proxy address. This account is automatically created when you enable read/write splitting.
Other accounts	Accounts created through the console or SQL statements. After an account is created, you can assign permissions to it as required. For details, see **Changing Permissions for Accounts**.

Constraints

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

Creating an Account Through TaurusDB

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 **Accounts**.

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

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

Figure 5-4 Creating an account

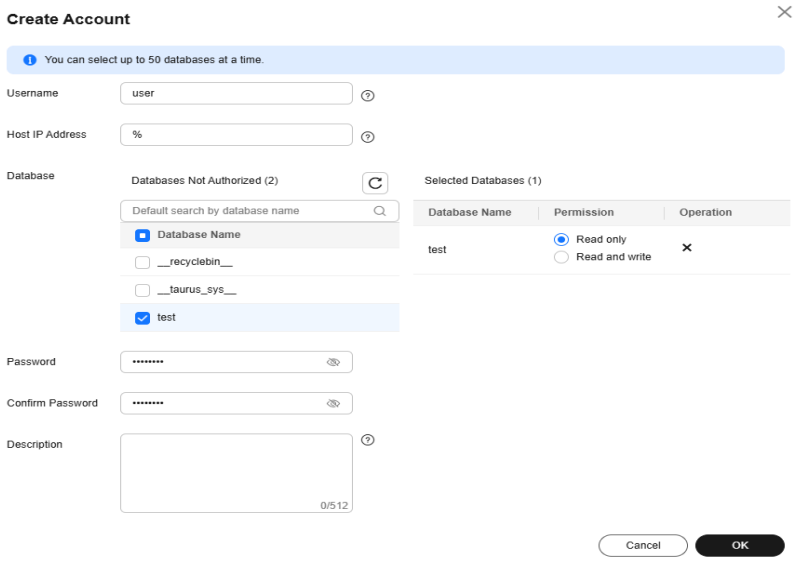


Table 5-5 Parameter description

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

Parameter	Description
Password	The password must: • Consist of 8 to 32 characters. • Contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&|.). • Comply with the values of **validate_password** parameters. To check the password-related parameter values, click an instance name, choose **Parameters** in the navigation pane, and search for **validate_password** in the upper right corner of the page. • Be different from the username or the username spelled backwards.
Confirm Password	The value must be the same as that of **Password**.
Description	The description can consist of up to 512 characters. It cannot contain carriage returns or any of the following special characters: !<"='>&

Step 8 Click **OK**.

Step 9 After the account is created, manage it on the **Accounts** page.

----End

Creating an Account Through DAS

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **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 an account using either of the following methods:

- Choose **SQL Operations** > **SQL Query**. In the displayed SQL window, select the target database and run the following command:
  ```
  create user username;
  ```
- Choose **Account Management** > **User Management** and click **Create User**.
  For detailed operations and parameter settings, see **Creating a User**.

Step 7 After the account is created, manage it on the **Accounts** page.

----End

APIs

- Creating a Database Account
- Querying Database Users
- Modifying the Description of a Database User

5.3.2 Resetting the Password of an Account

Scenarios

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

Constraints

- This operation is not allowed when another operation is being performed on your DB instance.
- After the password is reset, the DB instance will not be rebooted and your permissions will not be changed.
- You can query password reset records on the CTS console. For details, see *Cloud Trace Service User Guide*.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

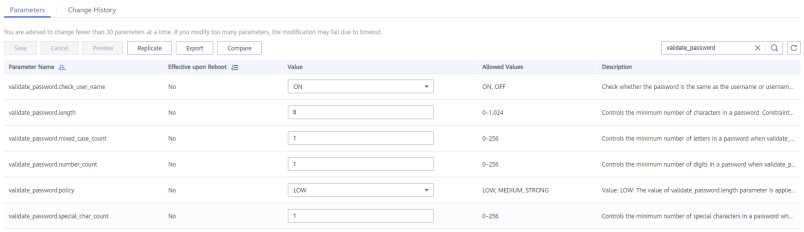
Step 6 In the displayed dialog box, enter a new password, confirm it, and click **OK**.

The password must meet the following requirements:

- It must consist of 8 to 32 characters.
- It must contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&|.).
- It must comply with the values of **validate_password** parameters.

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

Figure 5-5 Checking the password-related parameters



- The password you entered in the **Confirm Password** text box must be the same as that you entered in the **New Password** text box.
- It cannot be the username or the username spelled backwards.

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

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

Step 8 After the password is reset, use the new password to log in to the instance.

----End

APIs

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

5.3.3 Changing Permissions for Accounts

Scenarios

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

Constraints

- This operation is not allowed when another operation is being performed on your DB instance.
- Take care when changing account permissions. Inappropriately configured account permissions can impact the DB instance or workloads.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

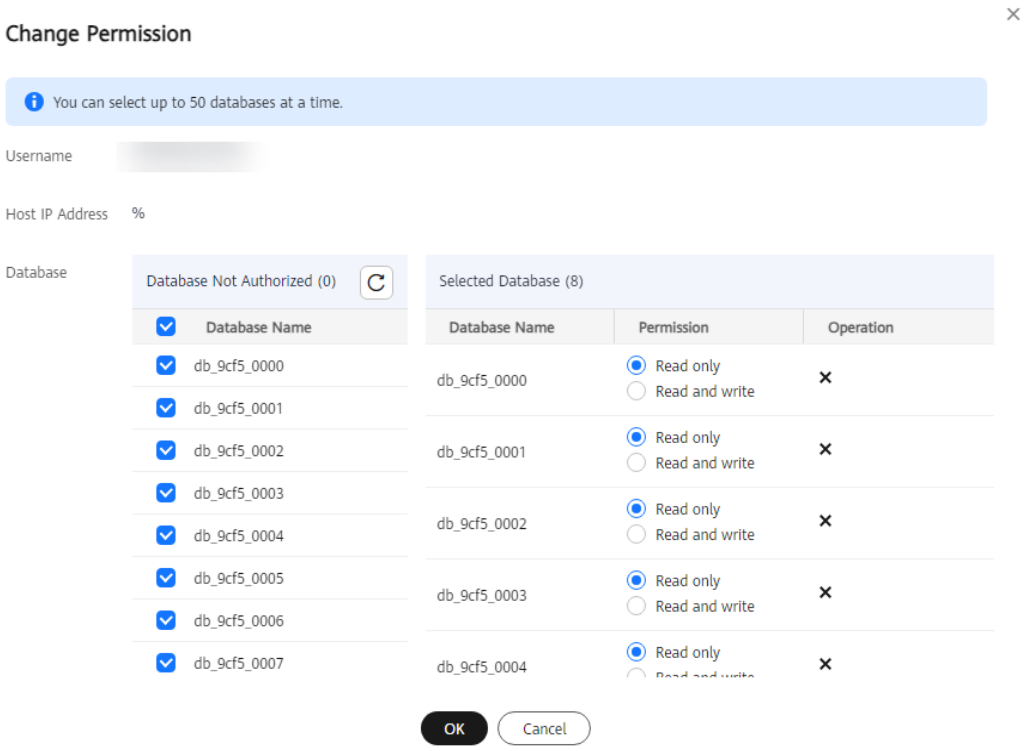
Step 4 On the **Instances** page, click the instance name.

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

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

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

Figure 5-6 Changing permissions



Step 7 On the **Accounts** page, check that the authorized databases and permissions of the account have been changed.

----End

APIs

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

5.3.4 Deleting an Account

Scenarios

You can delete accounts you have created.

Constraints

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

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

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

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

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

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

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

Step 9 Check that the account is no longer displayed on the **Accounts** page.

----End

APIs

Deleting a Database User

6 Data Migration

6.1 Data Migration Schemes

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

Migration Tools

Table 6-1 Migration tools

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

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

Migration Schemes

Table 6-2 Migration schemes

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

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

6.2 Migrating Data to TaurusDB Using mysqldump

You can use mysqldump to migrate data to TaurusDB.

Constraints

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

Preparing for the Migration

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.

     Purchase an ECS and **log in to the ECS**.
   - To connect to an instance through an EIP, you must:
     i. Bind the EIP to the instance. For details, see **Binding an EIP**.
     ii. Ensure that the local device can access the EIP that has been bound to the instance.
2. Install a MySQL client on the prepared ECS or device that can access the TaurusDB instance.
   - Install the MySQL client by following the instructions provided in **How Can I Install a MySQL Client?**
   - Ensure that the MySQL client version is the same as or later than that installed on the TaurusDB instance. The MySQL database or client provides mysqldump and mysql by default.

Exporting Data

Before migrating data from the source database to the TaurusDB instance, you need to export data from the source database first.

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

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

WARNING

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.*.*.* -P 3306 |sed -e 's/DEFINER[ ]*=[ ]*[^*]*\*/\*/' -e 's/DEFINER[ ]*=.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*=.*PROCEDURE/PROCEDURE/' -e 's/DEFINER[ ]*=.*TRIGGER/TRIGGER/' -e 's/DEFINER[ ]*=.*EVENT/EVENT/' > dump-defs.sql
Enter password:
```

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

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

```
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 device that can access the TaurusDB instance and then import the exported SQL files into that instance.

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 TaurusDB instance and view the results.

```
show databases;
```

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

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

----End

6.3 Migrating Data to TaurusDB Using the Export and Import Functions of DAS

Scenarios

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

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

Constraints

- Only one file that is no larger than 1 GB can be imported at a time.

- Only data files in .sql, .csv, or .xlsx format can be imported.
- If data files are exported as a .zip package, they cannot be directly imported. You need to extract the files first.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB cannot be imported.
- Data cannot be exported from or imported to cross-region OBS buckets.

Exporting an Entire Database

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, locate an instance and click **Log In** in the **Operation** column.

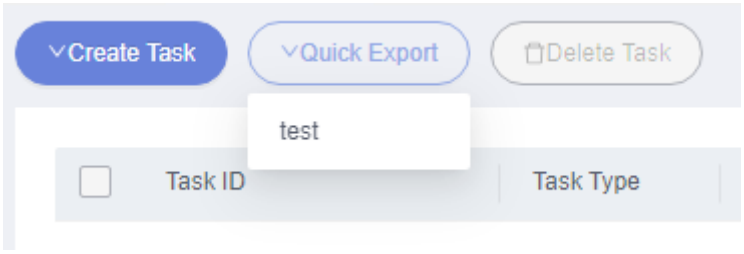
Step 5 On the displayed DAS login page, enter the username and password and click **Log In**.

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

Step 7 Export an entire database.

Method 1: Use the quick export function.

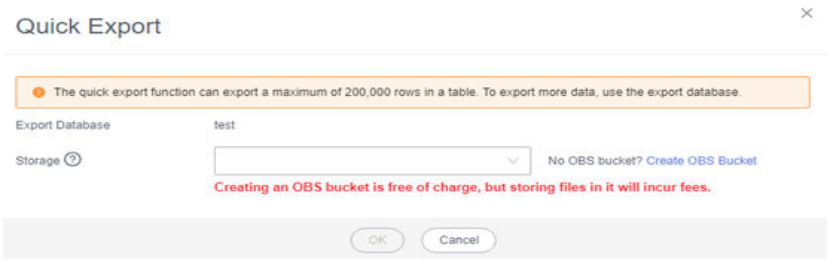
1. Click **Quick Export** and select the database to be exported.



2. In the displayed dialog box, select a storage path and click **OK**.

NOTE

- DAS does not store any data. The exported data files are stored in the OBS bucket that you have created.
- Creating an OBS bucket is free, but you will be billed for storing data in the bucket.



Method 2: Create an export task.

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

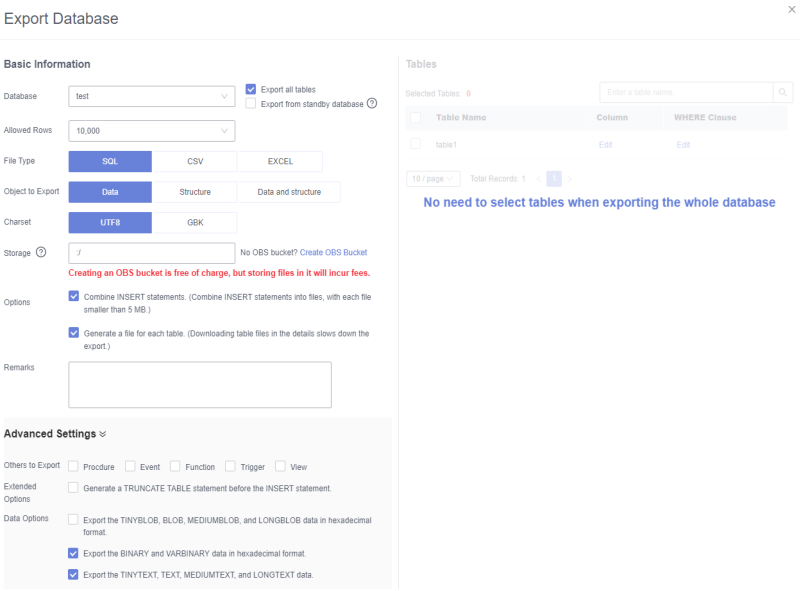


Table 6-3 Parameter description

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

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

3. Click **OK**.

----End

Exporting Some Data Tables

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, locate an instance and click **Log In** in the **Operation** column.

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

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

Step 7 Click **Create Task** > **Export Database**.

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

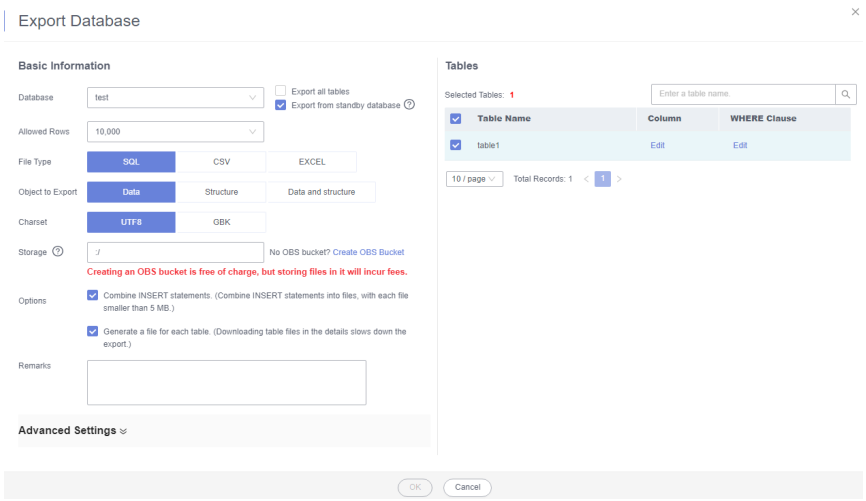


Table 6-4 Parameter description

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

Category	Parameter	Description
	Remarks	-
Advanced Settings	You can configure advanced options as required.	

Step 9 Click **OK**.

----End

Exporting SQL Result Sets

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, locate an instance and click **Log In** in the **Operation** column.

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

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

Step 7 Click **Create Task** > **Export SQL Result**.

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

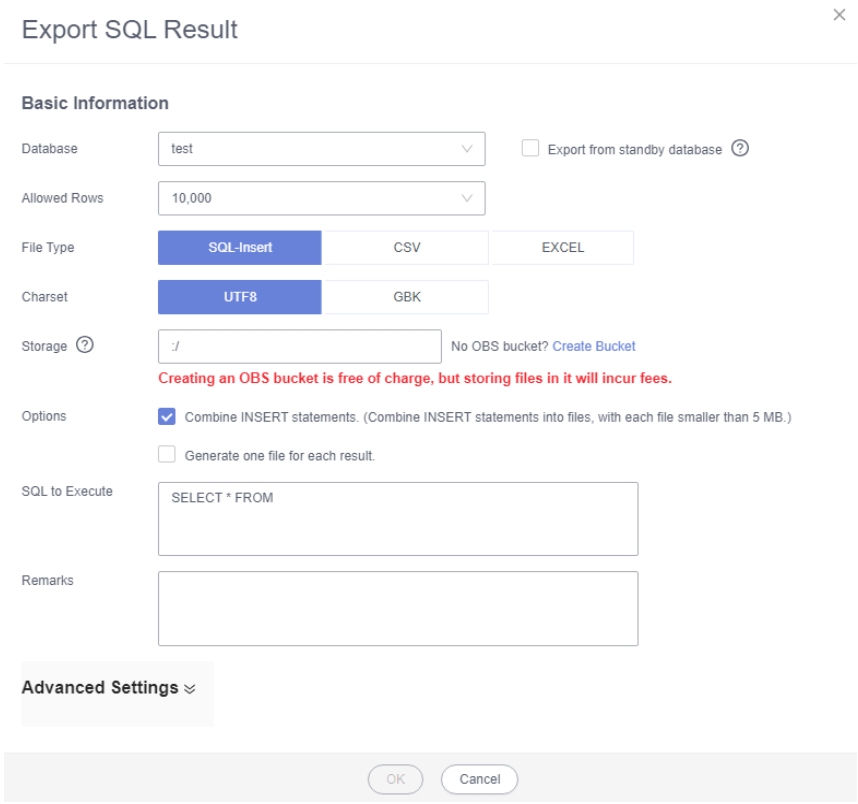


Table 6-5 Parameter description

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

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

Step 9 Click **OK**.

----End

Importing Data

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

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

Step 3 Click  in the upper left corner of the page, choose **Databases** > **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 On the top menu bar, choose **Import and Export** > **Import**.

Step 7 Click **Create Task**.

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

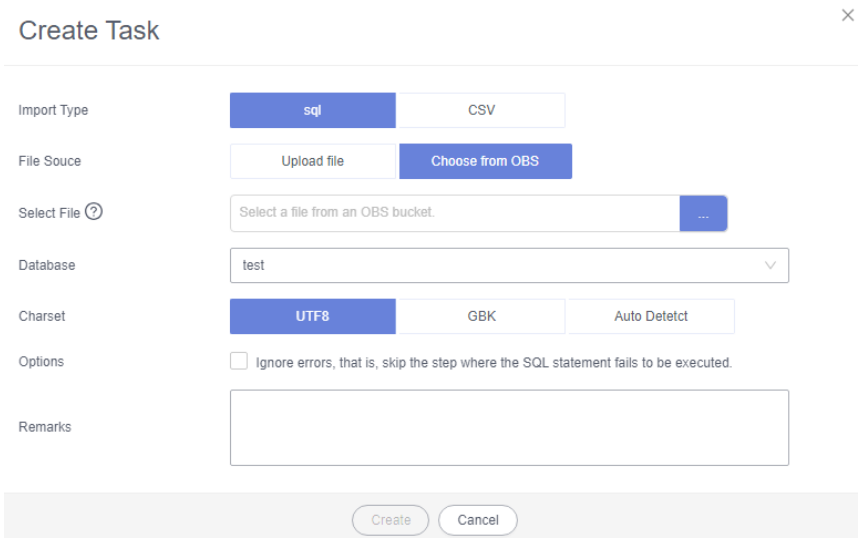


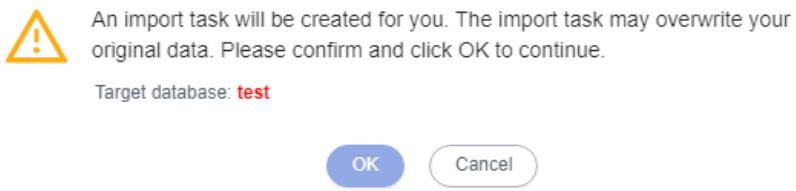
Table 6-6 Parameter description

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

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

Step 9 Click **Create**.

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



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

----End

6.4 Migrating Data to TaurusDB Enterprise Edition (OBT)

You can migrate data from RDS for MySQL and TaurusDB Standard Edition instances to TaurusDB Enterprise Edition instances.

Constraints

- To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- To use this function, you must have the **DRS Administrator** privilege of DRS and the **GaussDB FullAccess** privilege of TaurusDB.
- Data can only be migrated to pay-per-use TaurusDB Enterprise Edition instances.
- This function only supports the migration from RDS for MySQL and TaurusDB Standard Edition to TaurusDB Enterprise Edition.
- The system will only switch your application's connection address from the RDS for MySQL primary node to the TaurusDB primary node.

- This function is not available to RDS for MySQL instances with an EIP bound, with database proxy enabled, or with TDE enabled.

Step 1: Create a Migration Task

1. Go to the **Buy DB Instance** page.
2. On the displayed **Custom Config** page, configure required information and click **Next**.
   - Basic configuration

     Figure 6-1 Basic configuration

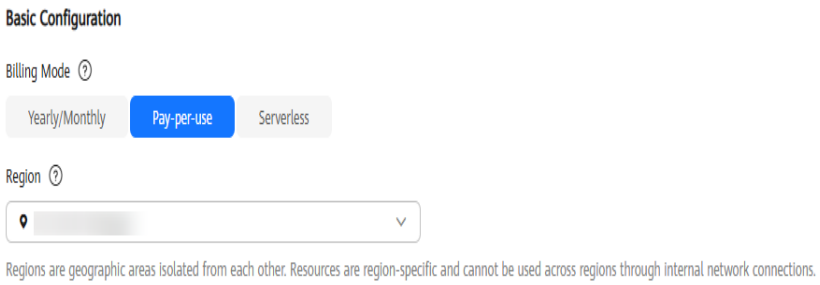
     

     Table 6-7 Basic configuration

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	Region where an instance is deployed. **NOTE** You cannot change the region of an instance once it is purchased.

   - Resource selection

Figure 6-2 Resource selection

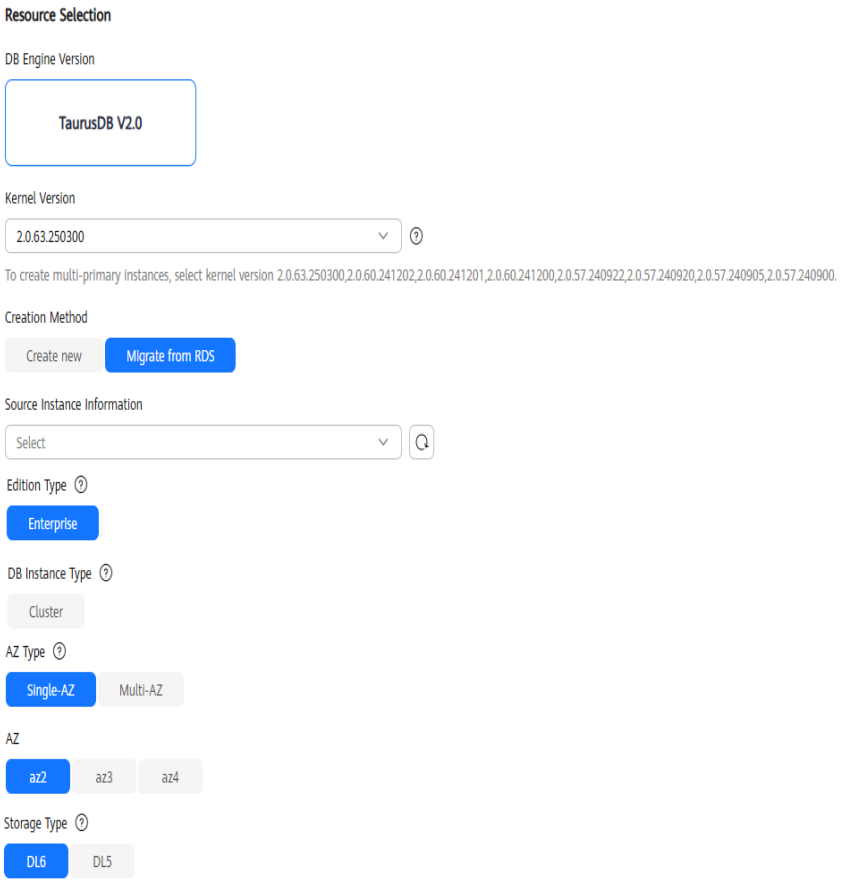


Table 6-8 Resource selection

Parameter	Description
DB Engine Version	Select **TaurusDB V2.0**.
Kernel Version	DB kernel version. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** To specify the kernel version when buying an instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
Creation Method	Select **Migrate from RDS**.
Source Instance Information	Select an RDS for MySQL instance or TaurusDB Standard Edition instance from the drop-down list.

Parameter	Description
DB Instance Type	Only cluster instances are supported. 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.
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. • **Single-AZ**: The primary node and read replicas are deployed in the same AZ. • **Multi-AZ**: The primary node and read replicas are deployed in different AZs to achieve higher availability and reliability. It is suitable for workloads that require cross-AZ DR or are insensitive to cross-AZ latency.

Parameter	Description
Storage Type	• **DL6** The original shared storage. The default storage type of TaurusDB instances created before July 2024 is shared storage, while that of TaurusDB instances created in July 2024 and beyond is **DL6**. DL6-based instances achieve zero RPO with a 3-AZ deployment and deliver better performance and higher peak throughput. They are suitable for core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming. • **DL5** A new type of storage. With Huawei Cloud's hardware and network infrastructure technologies, DL5-based instances maintain the same high availability (zero RPO in the 3-AZ deployment) as DL6-based instances. Although the peak performance of DL5-based instances may be a bit less than what you get with DL6-based instances, the cost per unit of capacity is a lot less. DL5-based instances are suitable for CPU-intensive sub-core business systems, or application modules that need to minimize costs. For more information about storage types, see **Storage Types**.

– Instance options

Figure 6-3 Specifications and storage

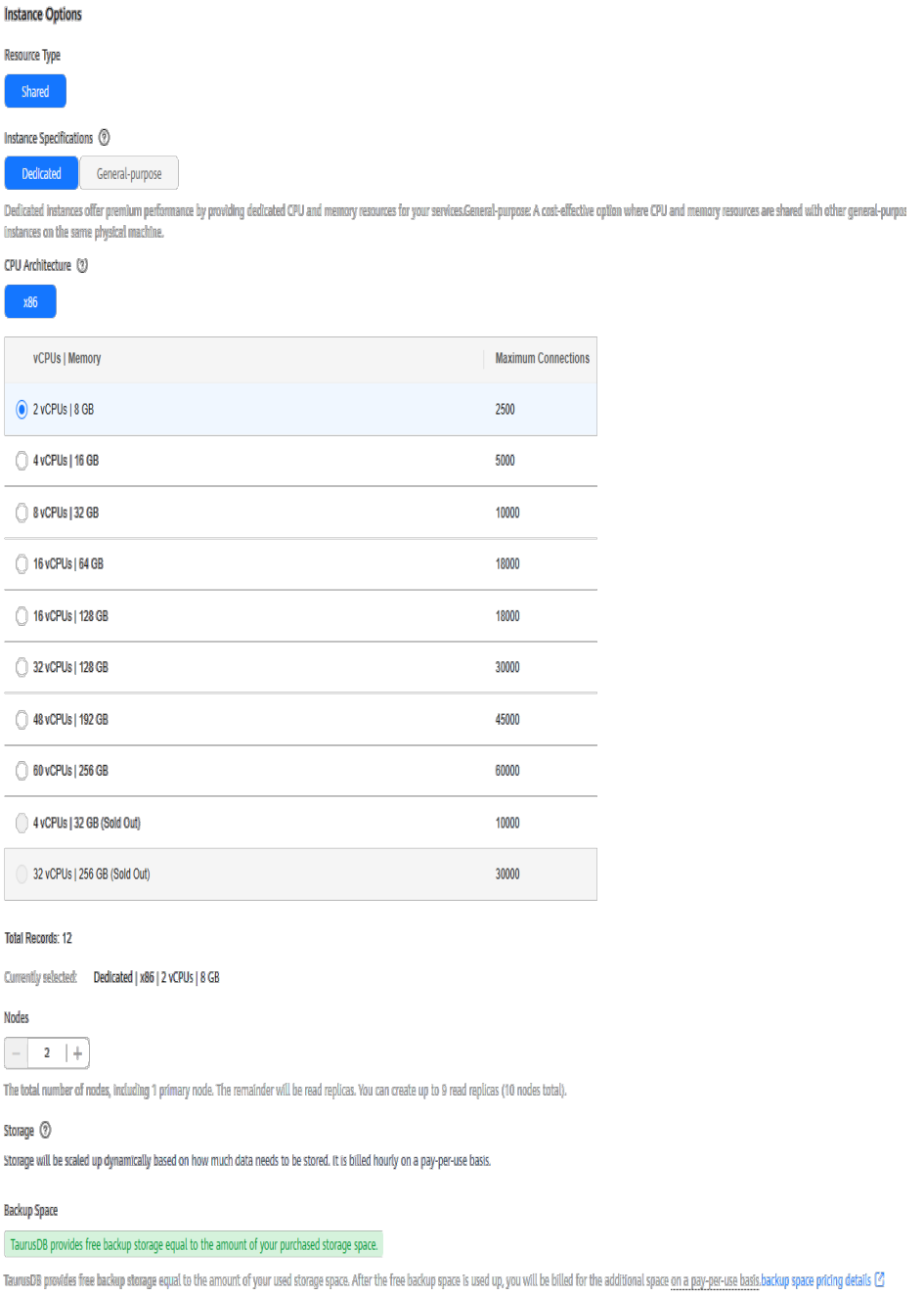


Table 6-9 Specifications and storage

Parameter	Description
Resource Type	Select **Shared**.

Parameter	Description
Instance Specifications	TaurusDB is a cloud-native database that uses the shared storage. To ensure that instances run stably under high read/write pressure, TaurusDB controls the read/write peaks of instances based on instance specifications. For details about how to select instance specifications, see **Performance White Paper**. For more information about instance specifications, see **Instance Specifications**.
CPU Architecture	Select **x86** or **Kunpeng**. • **x86**: x86 instances use Intel® Xeon® Scalable processors and feature robust and stable computing performance. When working on high-performance networks, the instances provide the additional performance and stability that enterprise-class applications demand. • **Kunpeng**: Kunpeng instances use Kunpeng 920 processors and 25GE high-speed intelligent NICs for powerful compute and high-performance networks, making them an excellent choice for enterprises needing cost-effective, secure, and reliable cloud services.
Nodes	This parameter is mandatory for cluster instances. • By default, each instance can contain one primary node and multiple read replicas. • You can create up to 9 read replicas for a pay-per-use instance at a time. • After an instance is created, you can add read replicas as required. Up to 15 read replicas can be added to an instance. For details, see **Adding Read Replicas to a DB Instance**.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage of a pay-per-use instance will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
Backup Space	TaurusDB provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis.

Figure 6-4 Setting instance information

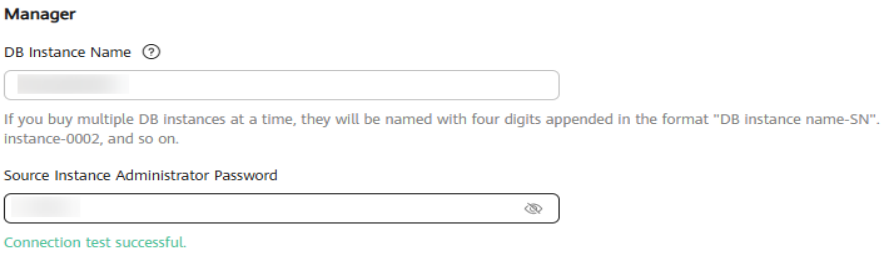


Table 6-10 Instance information

Parameter	Description
DB Instance Name	Enter the TaurusDB instance name at the destination. The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. • If you create multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on. • The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Source Instance Administrator Password	Enter the administrator password of the source instance. After entering the password, click **Test Connection** to verify the password.

– Advanced settings

Figure 6-5 Advanced settings

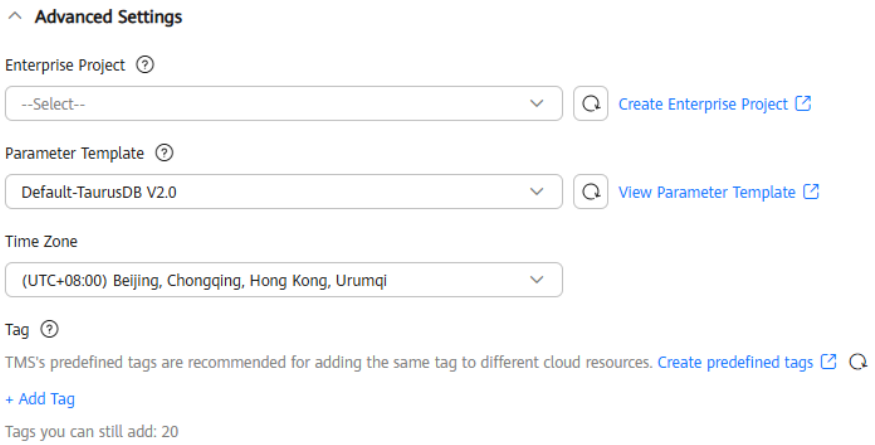
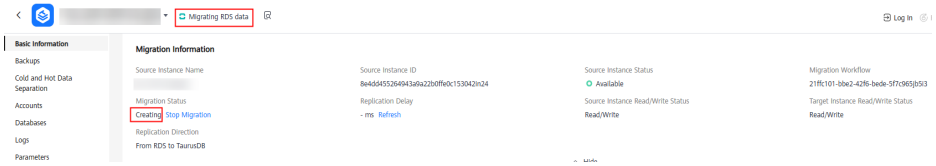


Table 6-11 Advanced settings

Parameter	Description
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can select the default parameter template, the high-performance parameter template, or a custom parameter template in the current region as required. For details about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters of a DB Instance**.
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.
Tag	Tags a DB instance. This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

3. Confirm your settings.
   - If you need to modify your settings, click **Previous**.
   - If you do not need to modify your settings, click **Submit**.
4. After the TaurusDB instance is created, click the instance name to go to the **Basic Information** page. The instance status becomes **Migrating RDS data**, which means the system starts to create a DRS migration task.

Figure 6-6 Creating a migration task

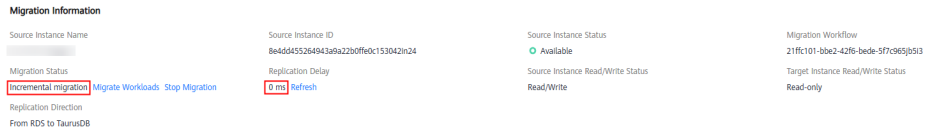


Wait until the migration status changes to **Incremental migration**, which means the DRS migration task has been created.

Step 2: Migrate Workloads

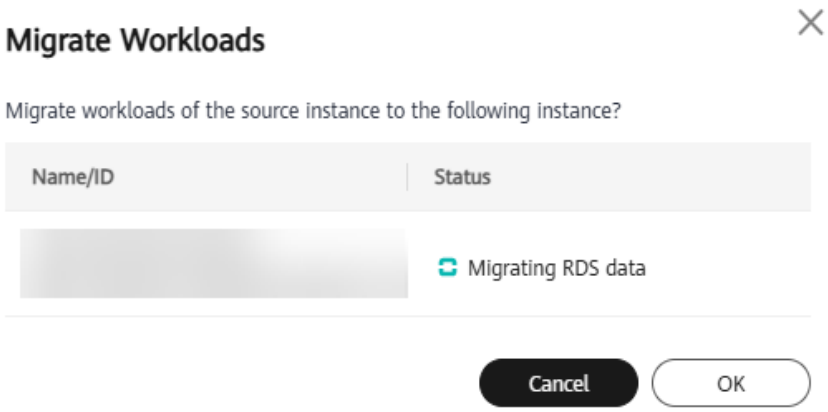
1. On the **Basic Information** page of the TaurusDB instance, wait until the migration status becomes **Incremental migration** and the replication delay is less than 60 seconds, and then migrate workloads.

   Figure 6-7 Observing the migration status and replication delay

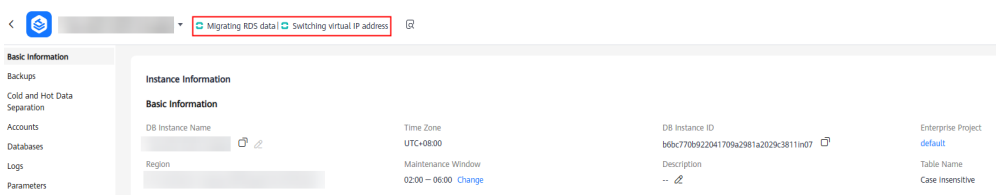
2. Click **Migrate Workloads**.
3. In the displayed dialog box, confirm the instance information and click **OK**.

   Figure 6-8 Migrating workloads

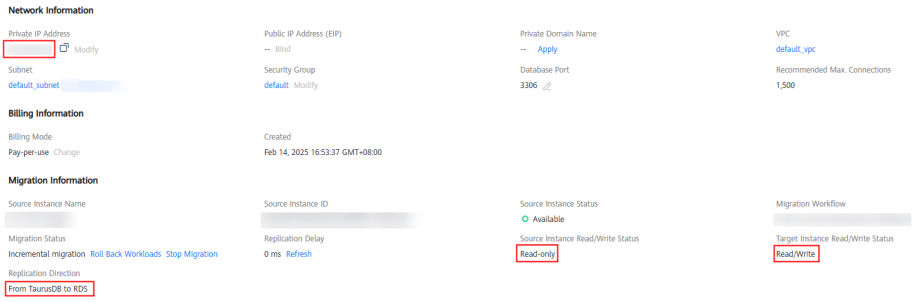
4. On the **Basic Information** page of the TaurusDB instance, check that the instance status is **Migrating RDS data | Switching virtual IP address**.

   Figure 6-9 Checking the TaurusDB instance status

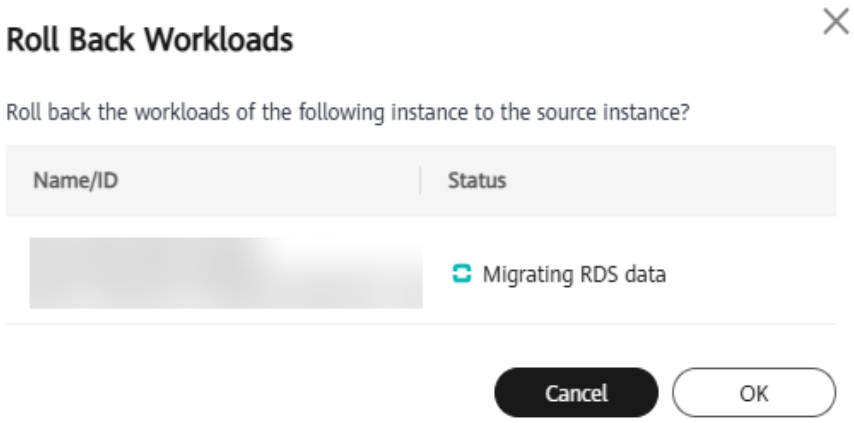
After the migration is complete, the read/write statuses of both the source and destination instances changed, the private IP address changed to that of the source instance, and the replication direction also changed.

Figure 6-10 Checking the read/write statuses and replication direction after the migration



If you do not want to use the TaurusDB instance, click **Roll Back Workloads**. In the displayed dialog box, confirm the instance information and click **OK**.

Figure 6-11 Rolling back workloads



After the workloads are rolled back, related information is restored to the status before the migration.

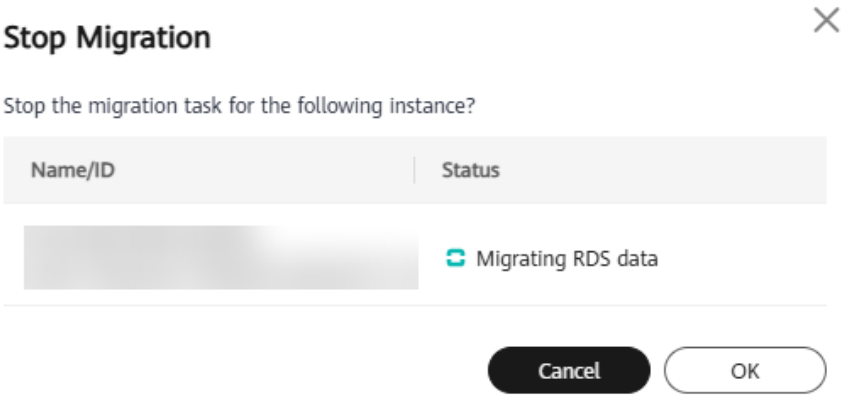
Step 3: Stop the Migration

After the migration is complete, you can stop the migration. After the migration is stopped, the DRS migration task is terminated. Data will not be synchronized between instances, and the migration lock will be released. The TaurusDB instance will be billed and the system triggers a full backup for the instance.

You can perform the following operations to stop the migration:

1. On the **Basic Information** page of the TaurusDB instance, click **Stop Migration**.
2. In the displayed dialog box, confirm the instance information and click **OK**.

Figure 6-12 Confirming the instance information



7 Instance Management

7.1 Viewing the Overall Status of DB Instances

The **Overview** page gives you a bird's eye view of TaurusDB instances, including instances by status, alarms, and intelligent diagnosis.

Functions

Table 7-1 lists the functions of the **Overview** page.

Table 7-1 Function description

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

Instances by Status

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Instances by Status** area, check the status of all TaurusDB instances under the current account.

Figure 7-1 Checking instances by status

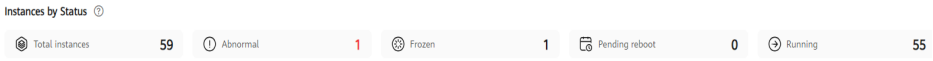


Table 7-2 Status description

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

----End

Alarms

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

1. In the upper right corner of the **Alarms** area, click **Create Alarm Rule** to access the Cloud Eye console.
   - By default, the system has a built-in alarm rule, which can be modified, disabled, and deleted. For details, see **Modifying an Alarm Rule**.
   - Click **Create Alarm Rule** to create an alarm rule to monitor a metric or event for instances. For details, see **Creating an Alarm Rule**.
2. In the upper right corner of the **Alarms** area, select a time window and view alarm details.
   - The time window can be **Last 1 hour**, **Last 6 hours**, **Last 12 hours**, **Last day**, **Last week**, or **Last month**.

- The **Alarm Severity** area displays the total number of alarms and the number of alarms of each severity. Alarm severities include **Critical**, **Major**, **Minor**, and **Warning**.
- The **Top 5 Instances by Total Number of Alarms** area displays alarm statistics of the top 5 instances with the largest number of alarms. You can hover over an instance to view the number of alarms of each severity.
- For details about critical alarms, see **Table 7-3**.

Table 7-3 Critical alarm description

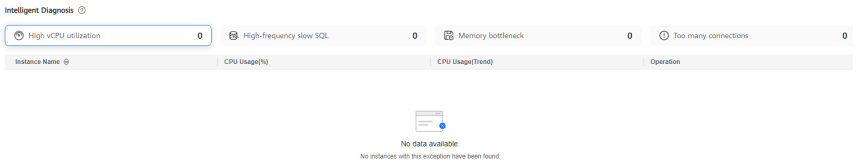
Parameter	Description
Instance Name	Name of the instance where the alarm was reported. After the page is refreshed, the latest alarm details will be displayed in real time.
Status	You can view active alarms of all instances in the current region, including alarms in the **Alarm** (metric) and **Triggered**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

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

Intelligent Diagnosis

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

Figure 7-2 Health diagnosis



Click an abnormal diagnosis item to view the abnormal instances and related metric data.

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

For details about supported diagnosis items and their handling suggestions, see **Table 7-4**.

Table 7-4 Intelligent diagnosis details

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

Diagnosis Item	Metric	Metric Description	Handling Suggestion	Reference
Memory bottleneck	Memory Usage	Memory usage of the monitored object	• Upgrade instance specifications. • Optimize SQL statements to reduce the use of temporary tables. • Reconnect sessions at a specific interval to release memory of the sessions.	**How Do I Handle a Large Number of Temporary Tables Being Generated for Long Transactions and High Memory Usage?**
High-frequency slow SQL	Slow Query Logs(Count/min)	Number of TaurusDB slow query logs generated per minute	• Optimize slow SQL statements based on the execution plan. • Upgrade vCPUs.	**How Do I Handle Slow SQL Statements Caused by Inappropriate Composite Index Settings?**

Diagnosis Item	Metric	Metric Description	Handling Suggestion	Reference
Too many connections	Total Connections(Count)	Total number of connections that connect to the TaurusDB server	• Check whether applications are connected, optimize the connections, and release unnecessary connections. • Check the instance specifications and upgrade them if needed.	What Do I Do If There Are Too Many Database Connections?
	Current Active Connections(Count)	Number of active connections		
	Connection Usage	Percent of used TaurusDB connections to the total number of connections		

7.2 Viewing Metrics

The **Metrics** page allows you to monitor TaurusDB instances.

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

Viewing Real-Time Metrics

1. Log in to the management console.
2. Click in the upper left corner and select a region and project.
3. Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.
4. In the navigation pane, choose **Metrics**.
5. View the real-time performance metrics of a created TaurusDB instance under the current account.

Figure 7-3 Viewing real-time metrics

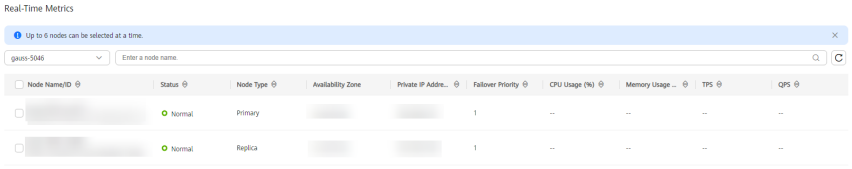


Table 7-5 Parameter description

Parameter	Description
Node Name/ID	Only monitoring data for the nodes of a created TaurusDB instance is displayed.
Status	The value can be: • **Normal**: Real-time monitoring data is displayed. **NOTE** The monitoring data and charts are available for a new instance after the instance runs for about 10 minutes. • **Abnormal**: There is no monitoring data. The default values for all metrics are 0. The monitoring data is available only after the instance becomes normal. • **Stopped**: There is no monitoring data. The default values for all metrics are 0. The monitoring data is available only after the instance is started.
Node Type	The value can be: • Primary • Replica
Availability Zone	AZ where a node is located
Private IP Address	Private IP address of a node
Failover Priority	Failover priority of a node
Metrics	For details about metric description and handling suggestions for abnormal metrics, see **Table 7-6**. The following metrics are available: • CPU Usage • Memory Usage • TPS • QPS

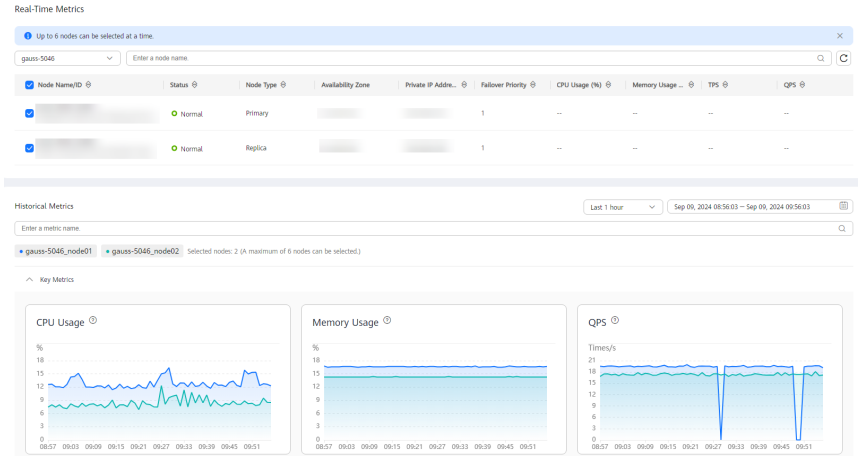
Table 7-6 Monitoring items

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

Viewing Historical Metrics

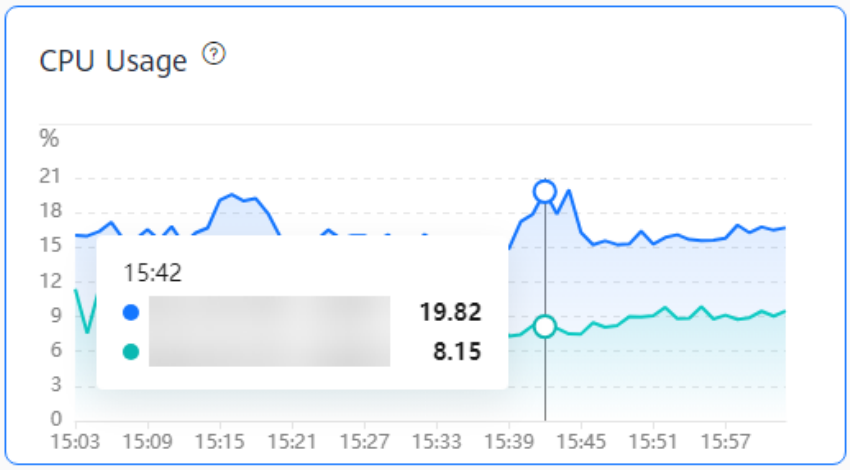
Select one or more nodes in the real-time metric list and then view their historical metrics in the **Historical Metrics** area.

Figure 7-4 Viewing historical metrics



- You can view the metrics of up to six nodes at a time.
- You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 24 hours, or last 7 days. You can also configure a time period.
- You can move the cursor to a point in time of a chart to view the performance metric at that point in time.

Figure 7-5 Viewing a performance metric at a point in time



7.3 Instance Lifecycle Management

7.3.1 Changing a DB Instance or Node Name

Scenarios

You can change the name of a TaurusDB instance or its node for easy identification.

Changing a DB Instance Name

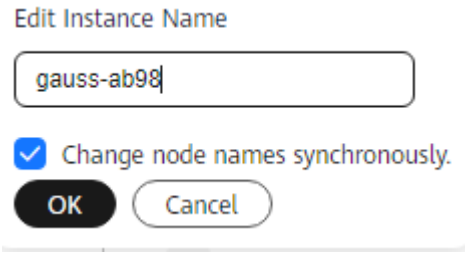
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

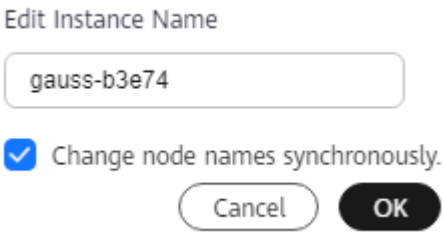
Step 4 On the **Instances** page, locate a DB instance and click in the **Name/ID** column to edit the DB instance name.

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



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

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



- The instance name must start with a letter and consist of 4 to 64 characters. Only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_) are allowed.
- When changing the instance name, you can determine whether to select **Change node names synchronously** as required. If this option is selected, the names of the corresponding nodes are changed when the instance name is changed. If this option is not selected, only the instance name is changed, and the corresponding node names are not changed.
- If you want to submit the change, click **OK**. If you want to cancel the change, click **Cancel**.

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

----End

Changing a Node Name

Step 1 Log in to the management console.

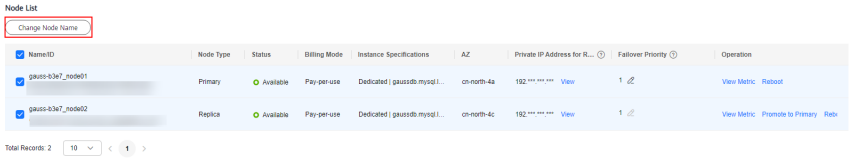
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **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, select one or more nodes, click **Change Node Name**.

Figure 7-8 Changing node names



Alternatively, click next to a node name to edit the node name.

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

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

Step 7 Check that the node name has been changed.

----End

APIs

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

7.3.2 Modifying a DB Instance Description

Scenarios

After a TaurusDB instance is created, you can add a description for it.

Modifying a DB Instance Description

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, locate an instance and click  in the **Description** column to edit the instance description.

- The instance description can contain up to 64 characters, and cannot start with or end with a space. Only letters, digits, hyphens (-), underscores (_), periods (.), and spaces are allowed.
- To submit the modification, click **OK**. To cancel the modification, click **Cancel**.

Alternatively, click the instance name to go to the **Basic Information** page. Locate **Description** in the **Instance Information** area, and click  to edit the instance description.

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

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

----End

APIs

- **Modifying the Description of a DB Instance**
- **Querying DB Instances**
- **Querying Details of a DB Instance**
- **Querying Details of DB Instances in Batches**

7.3.3 Rebooting a DB Instance or Node

Scenarios

You may need to reboot a DB instance or node for maintenance purposes. For example, after modifying some parameters, you may need to reboot your instance to apply the modifications. You may need to reboot a node to resolve database connection issues.

Constraints

Table 7-7 Constraints

Scenario	Description
Rebooting a DB instance	• If the DB instance status is **Abnormal**, the reboot may fail. • To shorten the time required, reduce database activities during the reboot to reduce rollback of transit transactions. • Rebooting a DB instance will interrupt services briefly. During this period, the instance status is **Rebooting**. • A DB instance will be unavailable when it is being rebooted. Rebooting a DB instance will clear the cached memory in it. To prevent traffic congestion during peak hours, you are advised to reboot the DB instance during off-peak hours.
Rebooting a node	• Nodes in the **Abnormal** state can be rebooted. • To shorten the time required, reduce database activities during the reboot to reduce rollback of transit transactions. • Rebooting a node will interrupt services briefly. During this period, the node status is **Rebooting node**. • A node will be unavailable when it is being rebooted. You are advised to reboot the node during off-peak hours and ensure that your applications support automatic reconnection. • If a parameter of your DB instance is modified, you need to first reboot the DB instance to apply the modification, and then reboot a node of the DB instance.

Rebooting a DB Instance

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **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 (if any) are also rebooted.

Step 5 In the displayed dialog box, set **Scheduled Time** and click **OK**.

Figure 7-9 Rebooting a DB instance

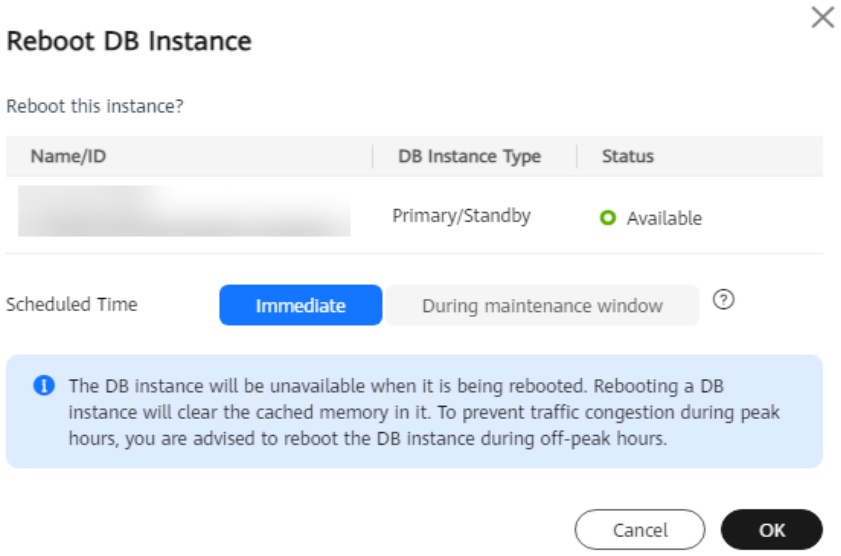


Table 7-8 Rebooting a DB instance

Parameter	Description
Scheduled Time	You can reboot a DB instance immediately or during the maintenance window. • **Immediate**: The DB instance will be rebooted immediately. • **During maintenance window**: The DB instance will be rebooted during a maintenance window. The maintenance window is 02:00–06:00 by default and you can change it as required. Changing the maintenance window will not affect the timing that has already been scheduled. A reboot task configured during a current maintenance window will not be executed until the next maintenance window.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

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

Step 7 View the reboot progress on the **Task Center** page. If the task status becomes **Completed** and the instance status becomes **Available**, the DB instance has been rebooted successfully.

----End

Rebooting a Node

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

Step 5 In the **Node List** area, select the target node and click **Reboot** in the **Operation** column.

Step 6 In the displayed dialog box, set **Scheduled Time** and click **OK**.

Table 7-9 Rebooting a node

Parameter	Description
Scheduled Time	You can reboot a node immediately or during the maintenance window. • **Immediate**: The node will be rebooted immediately. • **During maintenance window**: The node will be rebooted during a maintenance window. The maintenance window is 02:00–06:00 by default and you can change it as required. Changing the maintenance window will not affect the timing that has already been scheduled. A reboot task configured during a current maintenance window will not be executed until the next maintenance window.

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

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

Step 8 View the reboot progress on the **Task Center** page. If the task status becomes **Completed** and the node status becomes **Available**, the node has been rebooted successfully.

----End

APIs

- Rebooting a DB Instance
- Rebooting a Node

7.3.4 Exporting DB Instance Information

Scenarios

You can export DB instance information for further analysis.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, click **Export Instance Info** above the instance list. In the displayed dialog box, select the items to be exported and click **OK**.

Step 5 Check the .csv file locally after the export task is complete.

----End

7.3.5 Deleting a DB Instance

Scenarios

You can manually delete a DB instance billed on a pay-per-use or serverless basis on the **Instances** page.

Constraints

- Instances cannot be deleted when operations are being performed on them.
- If you delete a DB instance, its automated backups are also deleted and you are no longer billed for them. Manual backups are still retained and will incur additional costs.
- If you delete a DB instance, its read replicas are also deleted.
- If a backup of a DB instance is being used to restore data, the DB instance cannot be deleted.
- Deleted DB instances cannot be recovered and their resources will be released immediately. To retain data, back up the data first and then delete the DB instances.
- Deleted DB instances will be moved to the recycle bin, but will be permanently deleted after a length of time determined by the recycling policy.

Procedure

Step 1 Log in to the management console.

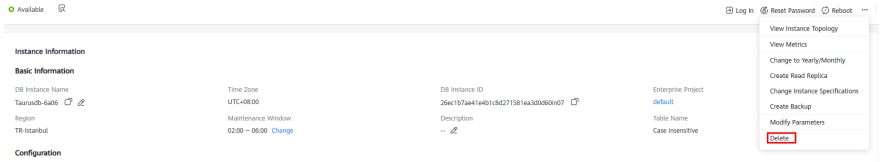
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Alternatively, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click ··· and choose **Delete**.

Figure 7-10 Deleting a DB instance on the **Basic Information** page



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

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

Step 6 In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

Step 7 Refresh the **Instances** page later to check that the deletion is successful.

----End

APIs

- **Deleting a DB Instance**
- **Deleting a Read Replica**
- **Querying DB Instances**

7.3.6 Rebuilding a DB Instance in the Recycle Bin

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

The recycle bin is enabled by default and cannot be disabled.

Billing

Yearly/monthly, pay-per-use, and serverless DB instances in the recycle bin are no longer billed, but manual backups retained for these DB instances are still billed. You can **delete the manual backups** to reduce costs. For details about backup pricing, see **How Is TaurusDB Backup Data Billed?**

Modifying the Recycling Policy

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

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

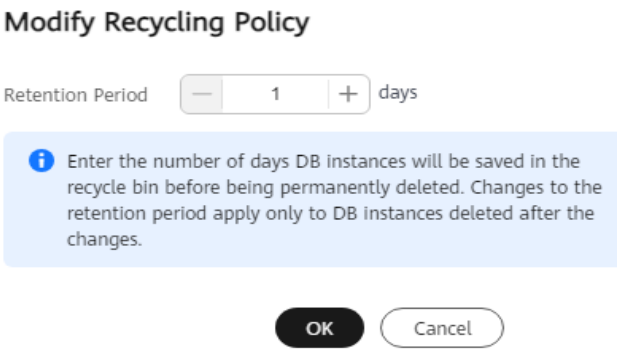
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set **Retention Period** (value range: 1 to 7, in days).

NOTE

The new recycling policy applies only to DB instances deleted after the changes.

Figure 7-11 Modifying the recycling policy



Step 5 Click **OK**.

Step 6 After the modification is successful, click **Modify Recycling Policy** on the **Recycle Bin** page to view the new recycling policy.

----End

Rebuilding a DB Instance

You can rebuild DB instances in 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 and choose **Databases** > **TaurusDB**.

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

Step 5 On the displayed **Rebuild DB Instance** page, set required parameters by referring to section "Buying a DB Instance".

Step 6 Click **Next**.

Step 7 Confirm the information and click **Submit**.

Step 8 Check the progress of task **Restoring to a new TaurusDB instance** on the **Task Center** page and refresh the **Instances** page later. If the instance status changes from **Creating** to **Available**, the DB instance is successfully rebuilt.

----End

7.4 Configuration Changes

7.4.1 Changing the vCPUs and Memory of a DB Instance or Node

Scenarios

You can upgrade or downgrade the specifications (vCPUs and memory) of a yearly/monthly or pay-per-use DB instance or node as needed.

Constraints

- A DB instance or node cannot be deleted when its specifications are being changed.
- You can change the specifications of a DB instance or just a single node within the instance. To change the specifications of a single node, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- Instance specifications can only be changed from the general-purpose edition to the dedicated edition.
- You can change the specifications of yearly/monthly or pay-per-use DB instances immediately or during a maintenance window. Serverless DB instances do not support specification changes.
- If you want to change instance specifications during a maintenance window, you can cancel the task before it starts. Once started, the task cannot be canceled.
- During an instance specification change, a read replica will be promoted to primary. To prevent service interruptions, perform the operation during off-peak hours.
- The time required for modifying specifications depends on factors such as the number of nodes, database load, and number of database tables.
- Changing instance specifications will change the private IP addresses for read of the primary node and read replicas. The connection addresses in your application need to be changed to prevent your services from being affected. You are advised to use the private IP address of a DB instance to connect your application.
- The specifications of the primary node and read replicas can be changed separately. When the specifications of the primary node are changed separately, the specifications of **synchronous nodes** are also changed.
- After the instance specifications are changed, the system will change the values of the following parameters accordingly:
  - **innodb_buffer_pool_size**
  - **innodb_log_buffer_size**
  - **max_connections**
  - **innodb_buffer_pool_instances**

  - **innodb_page_cleaners**
  - **innodb_parallel_read_threads**
  - **innodb_read_io_threads**
  - **innodb_write_io_threads**
  - **threadpool_size**
- The default value of **innodb_parallel_select_count** is **OFF** for instance with 16 vCPUs or less and **ON** for instances with more than 16 vCPUs.

  If you have modified this parameter before, the parameter value remains unchanged after the specifications are changed. Or, the default value is used.

Billing

Table 7-10 Billing

Billing Mode	Operation	Impact on Price
Yearly/ Monthly	Specification upgrade	After instance specifications are upgraded, the new instance specifications take effect in the original usage period. You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month TaurusDB cluster instance (instance specifications: dedicated, 2 vCPUs | 8 GB, 2 nodes; storage: DL6, 10 GB) in CN-Hong Kong on April 1, 2025. The instance price was $296 USD per month. On April 15, 2025, you changed the instance specifications to 4 vCPUs | 8 GB. The instance price became $586 USD per month. **Price difference = Price for the new instance specifications x Remaining period – Price for the original instance specifications x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Price difference = $586 USD x 0.5 – $296 USD x 0.5 = $145 USD

Billing Mode	Operation	Impact on Price
	Specification downgrade	After instance specifications are downgraded, the new instance specifications take effect in the original usage period. TaurusDB refunds the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month TaurusDB cluster instance (instance specifications: general-purpose, 4 vCPUs | 8 GB, 2 nodes; storage: DL6, 10 GB) in CN-Hong Kong on April 1, 2025. The instance price was $257.52 USD per month. On April 15, 2025, you changed the instance specifications to 2 vCPUs | 4 GB. The instance price became $132.72 USD per month. **Refunded fees = Price for the original instance specifications x Remaining period – Price for the new instance specifications x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and refunded fees are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Refunded fees = $257.52 USD x 0.5 – $132.72 USD x 0.5 = $62.4 USD
Pay-per-use	Specification upgrade	After instance specifications are changed, the new instance specifications are billed by hour. For details, see **Product Pricing Details**.
	Specification downgrade	

Changing the Specifications of a DB Instance

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

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

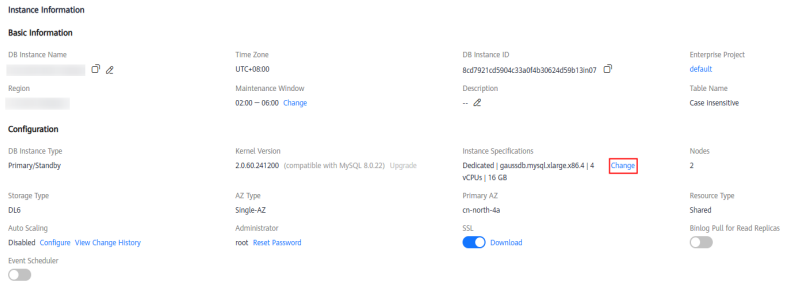
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

You can also enter the page for changing instance specifications in either of the following ways:

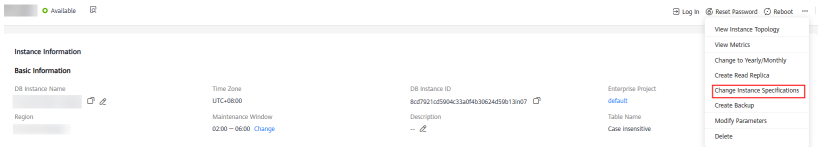
- Click the instance name to go to the **Basic Information** page. Click **Expand**. In the **Configuration** area, click **Change** under **Instance Specifications**.

  Figure 7-12 Changing specifications in the **Configuration** area

- Click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click ··· and choose **Change Instance Specifications**.

  Figure 7-13 Changing specifications on the **Basic Information** page

Step 5 On the displayed page, select the desired specifications. You can scale up or down the specifications as required.

You can change the specifications immediately or during the maintenance window.

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

Step 6 Click **Next**. On the displayed page, confirm the specifications.

- If you need to modify your settings, click **Previous** to go back to the page where you specify details.
- For pay-per-use instances, click **Submit**.

  To view the cost incurred by the instance specifications change, access the **Billing Center** page and then choose **Billing** > **Dashboard** in the navigation pane.
- For yearly/monthly instances:
  - Scaling down the specifications: click **Submit**.

    The refund is automatically returned to your account. You can access the **Billing Center** page and then choose **Orders** > **My Orders** in the navigation pane to view the details.

  - Scaling up the specifications: click **Submit**. The scaling starts only after the payment is successful.

Step 7 View the results.

Changing the instance specifications takes 5–15 minutes. During this period, the status of the instance on the **Instances** page is **Changing instance specifications**. After a few minutes, you can click the instance name to view the new instance specifications on the displayed **Basic Information** page.

----End

Changing the Specifications of the Primary Node

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

Step 2 In the node list, locate the primary node and click **Change Instance Specifications** in the **Operation** column.

Step 3 On the displayed page, select the desired specifications. You can scale up or down the specifications as required.

You can change the specifications immediately or during the maintenance window.

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

Step 4 Click **Next**. On the displayed page, confirm the specifications.

- If you need to modify your settings, click **Previous** to go back to the page where you specify details.
- For pay-per-use instances, click **Submit**.

  To view the cost incurred by the instance specifications change, access the **Billing Center** page and then choose **Billing** > **Dashboard** in the navigation pane.
- For yearly/monthly instances:
  - Scaling down the specifications: click **Submit**.

    The refund is automatically returned to your account. You can access the **Billing Center** page and then choose **Orders** > **My Orders** in the navigation pane to view the details.
  - Scaling up the specifications: click **Submit**. The scaling starts only after the payment is successful.

Step 5 View the results.

Check that the status of the primary node is **Changing instance specifications**. After a few minutes, view the node specifications on the **Basic Information** page to check that the change is successful. This process takes 5 to 15 minutes.

----End

Changing the Specifications of a Read Replica

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

Step 2 In the node list, locate a read replica and click **Change Instance Specifications** in the **Operation** column.

Step 3 On the displayed page, select the desired specifications. You can scale up or down the specifications as required.

You can change the specifications immediately or during the maintenance window.

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

Step 4 Click **Next**. On the displayed page, confirm the specifications.

- If you need to modify your settings, click **Previous** to go back to the page where you specify details.
- For pay-per-use instances, click **Submit**.

  To view the cost incurred by the instance specifications change, access the **Billing Center** page and then choose **Billing** > **Dashboard** in the navigation pane.
- For yearly/monthly instances:
  - Scaling down the specifications: click **Submit**.

    The refund is automatically returned to your account. You can access the **Billing Center** page and then choose **Orders** > **My Orders** in the navigation pane to view the details.
  - Scaling up the specifications: click **Submit**. The scaling starts only after the payment is successful.

Step 5 View the results.

Check that the node status is **Changing instance specifications**. After a few minutes, view the node specifications on the **Basic Information** page to check that the change is successful. This process takes 5 to 15 minutes.

----End

Follow-up Operations

Return to the instance list. In the navigation pane, choose **Task Center** and check the progress of the change task.

- If you have selected **Upon submission** for **Scheduled Time**:

  On the **Instant Tasks** page, search for **Changing the specifications of a TaurusDB instance** and check the execution progress. Instant tasks cannot be canceled.
- If you have selected **Maintenance Window** for **Scheduled Time**:

  On the **Scheduled Tasks** page, search for the instance ID and check the execution status of the change task.

If the task is in the **To be executed** state, you can click **Cancel** to cancel the task.

For details, see **Viewing a Task**.

APIs

- **Changing DB Instance Specifications**
- **Promoting a Read Replica to Primary**
- **Querying Database Specifications**
- **Querying Details of a DB Instance**

7.4.2 Changing the Storage Space of a DB Instance

Scenarios

If the original storage space of your yearly/monthly DB instance is insufficient or redundant as your workloads change, you can scale up or down the storage.

Constraints

- The storage of pay-per-use DB instances grows as needed, so you cannot manually scale up their storage. The storage of pay-per-use DB instances is not limited.
- When you purchase a yearly/monthly DB instance, you need to select storage for it as needed. If your purchased storage cannot meet your requirements, TaurusDB will automatically scale up the storage as needed and you will be billed on a pay-per-use basis for additional storage. If services requirements decrease later, the system preferentially scales down the storage that was automatically scaled up.

  For example, you purchased 10 GB of storage when purchasing a DB instance. Later, as workloads increased, TaurusDB automatically scaled up the storage to 18 GB as needed and you would be billed on a pay-per-use basis for the additional 8 GB of storage. Then, you manually scaled up the storage to 20 GB, which could meet your requirements. The 8 GB of storage that was scaled up by the system will be scaled down. You would only pay for the 20 GB of storage at yearly/monthly rates.
- The system changes storage of your DB instance as your services change, but you can change storage only by a multiple of 10 GB.
- During a storage change, services including backup service are not interrupted.
- You can change the storage space of a DB instance numerous times.
- If the storage space of a DB instance is being changed, you cannot reboot or delete the DB instance.

Billing

Table 7-11 Billing

Billing Mode	Operation	Impact on Price
Yearly/ Monthly	Storage scale-up	You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month TaurusDB cluster instance (instance specifications: dedicated, 2 vCPUs | 8 GB, 2 nodes; storage: DL6, 10 GB) in CN-Hong Kong on April 1, 2025. The unit price of storage space is $0.6 USD/GB per month. On April 15, 2025, you scaled up the storage by 50 GB. The total storage after scale-up is 60 GB. **Price difference = Scale-up volume x Unit price x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Price difference = 50 GB x $0.6 USD/GB x 0.5 = $15 USD

Procedure

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

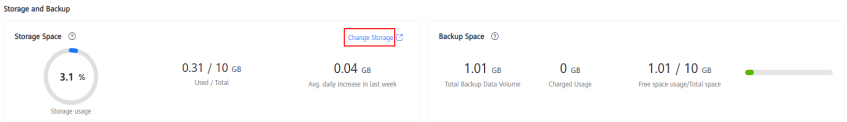
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate the instance and choose **More** > **Change Storage** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Storage and Backup** area, locate **Storage Space** and click **Change Storage**.

Figure 7-14 Scaling storage space



Step 5 Select the new storage space and click **Next**.

Storage space can be scaled up to 128,000 GB only by a multiple of 10 GB. Price after scaling is displayed in the lower left corner of the page.

Storage space can be scaled down to 40 GB only by a multiple of 10 GB. Refund price is displayed in the lower left corner of the page.

NOTE

To reduce the storage space of a DB instance to 10 GB, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Step 6 Confirm your settings.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** to go to the payment page and complete the payment.

Step 7 View the new storage.

On the **Instances** page, click the instance name to go to the **Basic Information** page and view the new storage.

----End

APIs

- **Scaling up Storage of a Yearly/Monthly DB Instance**
- **Querying Database Specifications**
- **Querying Details of a DB Instance**

7.4.3 Configuring Auto Scaling Policies for a DB Instance

Scenarios

You can configure auto scaling policies for your pay-per-use and yearly/monthly DB instances on the **Basic Information** page. When configuring auto scaling policies, you can enable or disable **Auto Scale-up** or **Auto Scale-down**. The scaling types include changing instance specifications and the number of read replicas.

Constraints

- This function is only available for pay-per-use and yearly/monthly DB instances.
- To set **Scaling Type** to **Number of read replicas** for a yearly/monthly DB instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. The read replicas that are automatically added or deleted will be billed based on a pay-per-use basis.
- To configure auto scaling policies, you must have the iam:agencies:listAgencies permission. If you do not have this permission, **create a custom policy**.

- Changing DB instance specifications will briefly interrupt services.
- If you want to set **Scaling Type** to **Number of read replicas**, there must be only one proxy instance. For details, see **Creating a Proxy Instance for Read/Write Splitting**.
- The system will delete or add read replicas. To prevent your services from being affected, you are advised not to use an IP address for read to connect to your applications.
- The pricing standard for auto scaling is the same as that for manual scaling. For details, see **Billing**.

Billing

- Pay-per-use instances

  The instance specifications and number of read replicas can be automatically changed.

  Pricing is listed on a per-hour basis, but bills are calculated down to the second. The old order automatically becomes invalid.

  To view the cost incurred by auto scaling, access the **Billing Center** page and then choose **Billing** > **Dashboard** in the navigation pane.
- Yearly/Monthly instances

  The instance specifications and number of read replicas can be automatically changed.

  You will be billed for the new specifications. For details, see **TaurusDB Pricing Details**.

  If the new specifications are less than the specifications that you purchased, the refund is automatically returned to your account. You can access the **Billing Center** page and then choose **Orders** > **My Orders** in the navigation pane to view the details.

Table 7-12 Pricing description for yearly/monthly instances

Billing Item	Description
Specifications after scale-up	You need to pay the following fee: Price of new specifications x Remaining duration x Number of nodes – Price of original specifications x Remaining duration x Number of nodes Note: Remaining duration = Number of remaining days in a calendar month/Total number of days in the calendar month Example: A customer placed and paid a monthly order for a TaurusDB instance on April 1, 2023. The instance contains 2 nodes and its specifications are 2 vCPUs and 8 GB of memory. The total subscription period would be 30 days and the instance would expire on April 30, 2023. On April 18, 2023, the instance specifications automatically scaled up to 4 vCPUs and 16 GB of memory. The remaining duration is 0.4 (12/30). The monthly price of the new specifications is $290 USD and that of the old specifications is $145 USD, so you need to pay for $116 USD (290 x 0.4 x 2 – 145 x 0.4 x 2).
Specifications after scale-down	For pricing details, see **Unsubscriptions**.
Added read replicas	New read replicas are billed based on the actual usage duration.
Deleted read replicas	Deleted read replicas are no longer billed.

Modifying Auto Scaling Policies

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

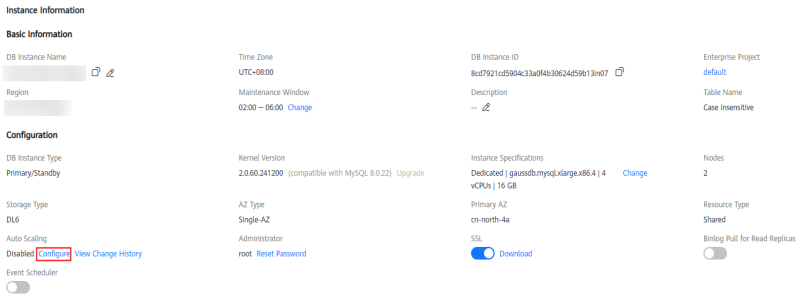
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Configuration** area, click **Configure** under **Auto Scaling**.

Figure 7-15 Modifying auto scaling policies



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

Figure 7-16 Modifying the auto scaling policy

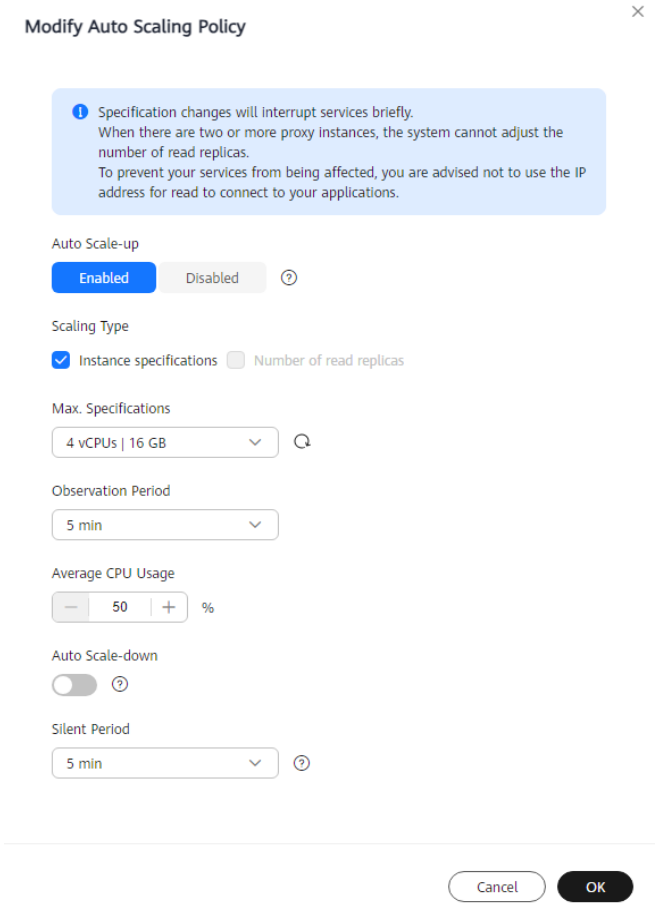


Table 7-13 Parameter configuration

Parameter	Parameter description
Auto Scale-up	You can enable or disable it as needed.

Parameter	Parameter description
Scaling Type	You can select one or more scaling types. • Instance specifications • Number of read replicas – The read replicas that are automatically added or deleted will be billed based on a pay-per-use basis. – After **Auto Scale-up** is enabled, read replicas that are automatically added cannot be promoted to primary. – If you deselect **Number of read replicas**, pay-per-use nodes created in the current instance will be automatically deleted. Exercise caution when performing this operation. **CAUTION** The account balance must be sufficient, or scaling up the specifications or adding read replicas may fail.
Observation Period	• Once auto scale-up is enabled, if the system observes any increases in the average CPU usage over the preset value, it upgrades the specifications or adds read replicas based on the read and write traffic. The system enters a silent period after each scale-up. • The minimum observation period is 5 minutes.
Average CPU Usage	Threshold for triggering an auto scale-up. If the average CPU usage of the primary node exceeds the preset value, an auto scale-up is triggered. Allowed range: 50%–100%
Max. Specifications	Maximum specifications after the final auto scale-up. The specifications can only be scaled up gradually and the system enters the silent period after each scale-up.
Max. Read Replicas	Only one read replica can be added at a time.
Replica Read Weight	If you have enabled read/write splitting, the new read replicas are automatically associated with the proxy instance.
Auto Scale-down	You can enable or disable it as needed. 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**.

Step 8 Refresh the **Basic Information** page later and check that auto scaling is enabled.

----End

Viewing Change History

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Configuration** area, click **View Change History** under **Auto Scaling**.

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

----End

7.4.4 Configuring Autoscaling for a DB Instance (OBT)

Scenarios

As your workloads grow, the used storage of a yearly/monthly DB instance may exceed the initially purchased storage. Additional storage will be billed on a pay-per-use basis. To better adapt to this change, TaurusDB provides storage autoscaling. When the required storage exceeds the initially purchased storage, the system will automatically scale up the storage and additional storage will be billed based on the storage prices for yearly/monthly subscriptions in **TaurusDB Pricing Details**. This meets workload growth needs while controlling costs effectively.

This section describes how to configure autoscaling after a DB instance is purchased.

Constraints

- To configure autoscaling, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- You can only configure autoscaling when your account balance is greater than or equal to $0 USD and the instance status is **Available**.
- If a yearly/monthly DB instance has pending orders, its storage cannot autoscale.
- The maximum allowed storage for a DB instance is 128,000 GB.
- If the product of the current storage and the increment is greater than the autoscaling limit, the storage can only autoscale to the limit.
- Storage autoscaling is unavailable when the DB instance is in any of the following statuses: changing instance specifications, upgrading a minor version, and rebooting.

Billing

After the storage is automatically scaled up, you will be billed for the new storage based on the time remaining in the requested period of your instance. For details, see the storage prices for yearly/monthly subscriptions in **TaurusDB Pricing Details**.

Example:

If you have selected 40 GB storage when buying an instance but the storage used later has reached 60 GB, you can configure autoscaling as required.

- If autoscaling is not configured, 20 GB (60 GB – 40 GB) will be billed on a pay-per-use basis.
- If autoscaling is configured, the new storage will be billed on a yearly/monthly basis, and any additional storage will still be billed on a pay-per-use basis.

  For example, if the storage autoscales from 40 GB to 50 GB, 50 GB will be billed on a yearly/monthly basis, and the remaining 10 GB (60 GB – 50 GB) will be billed on a pay-per-use basis. If the storage autoscales to 70 GB, 70 GB will be billed on a yearly/monthly basis. Since the used storage is less than the new storage, no additional pay-per-use billing is required.

Procedure

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

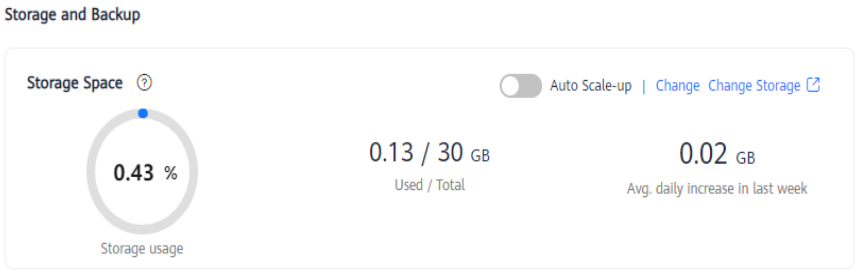
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Storage and Backup** area, click before **Auto Scale-up**.

Figure 7-17 Viewing storage space



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

Figure 7-18 Configuring autoscaling

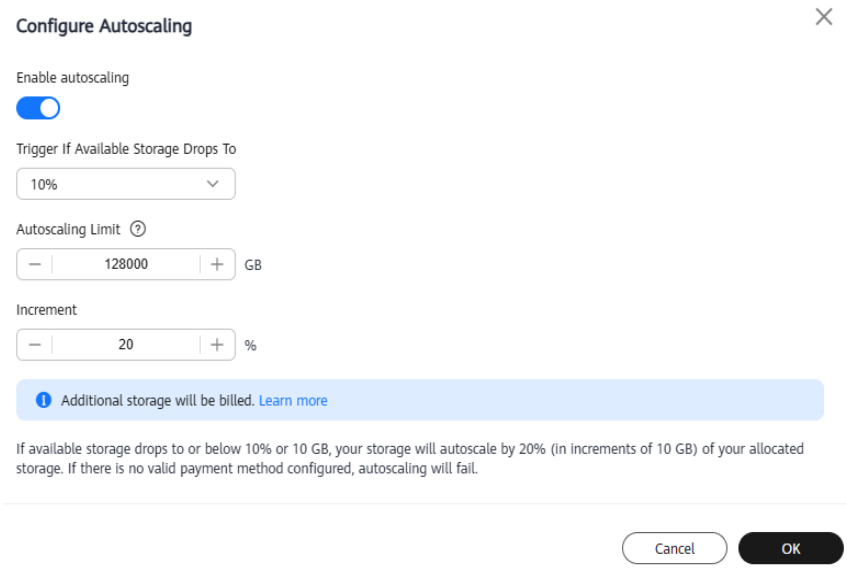


Table 7-14 Parameter description

Parameter	Description
Enable autoscaling	If you toggle on this switch, autoscaling is enabled.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered. Value range: **10%**, **15%**, and **20%**
Autoscaling Limit (GB)	The default value is **128000**. The value must be a multiple of 10 and greater than both the total storage and the used storage of the instance.
Increment	Percentage of allocated storage that is automatically scaled up each time. Value range: 5%–50%

Step 7 Click **OK**.

Step 8 In the **Storage and Backup** area on the **Basic Information** page, check that **Auto Scale-up** is enabled.

----End

7.4.5 Changing the Maintenance Window of a DB Instance

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required. To prevent service interruption, set the maintenance window to off-peak hours.

Constraints

Before maintenance is performed, TaurusDB will send SMS messages and emails to the contact person that has been set in the Huawei Cloud account.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

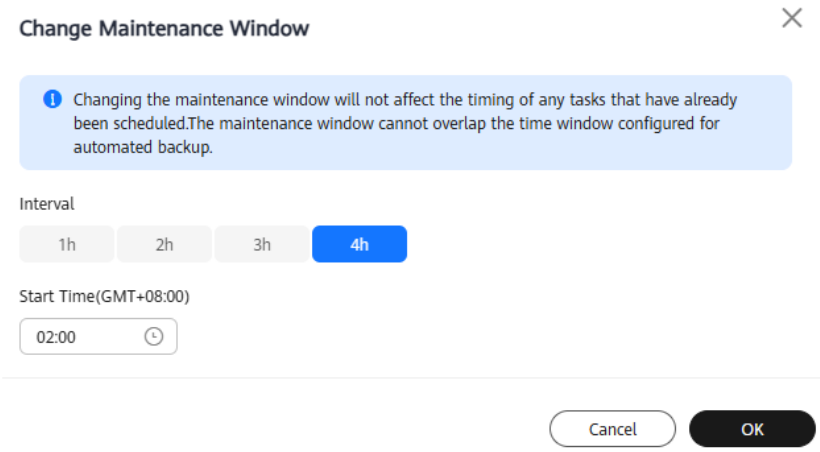
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 On the **Instance Information** page, click **Change** under **Maintenance Window**.

Figure 7-19 Changing a maintenance window

Step 6 In the displayed dialog box, select a maintenance window and click **OK**.

Figure 7-20 Changing a maintenance window



NOTE

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

Step 7 On the **Basic Information** page, check that the maintenance window is updated.

----End

APIs

Changing a Maintenance Window

7.4.6 Customizing Displayed Items of the Instance List

Scenarios

You can customize instance information items displayed on the **Instances** page based on your requirements.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, click and select items displayed in the instance list.

- The following items are displayed by default and cannot be hidden: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine**, **Status**, **Billing Mode**, **Private IP Address**, **Storage Type**, and **Operation**.
- The following items can be displayed or hidden: **Private IP Address for Read**, **Proxy Address**, **Private Domain Name**, **Enterprise Project**, **Created**, and **Database Port**.

----End

7.4.7 Enabling or Disabling Event Scheduler

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

Disclaimer

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

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

- The event scheduler is not triggered.
- Due to the particularity of the event scheduler, the actual execution may be different from what you expected.
- The event scheduler may impact analysis and judgment for issues with database usage.
- Heterogeneous disaster recovery cannot be used.
- Other unknown issues.

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

Constraints

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

Enabling Event Scheduler

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

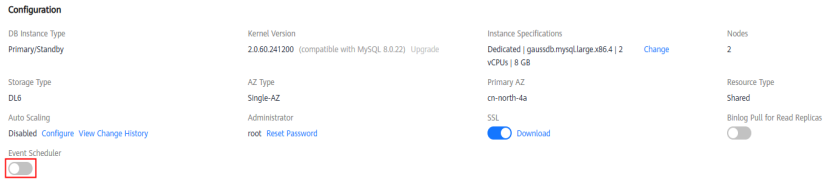
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

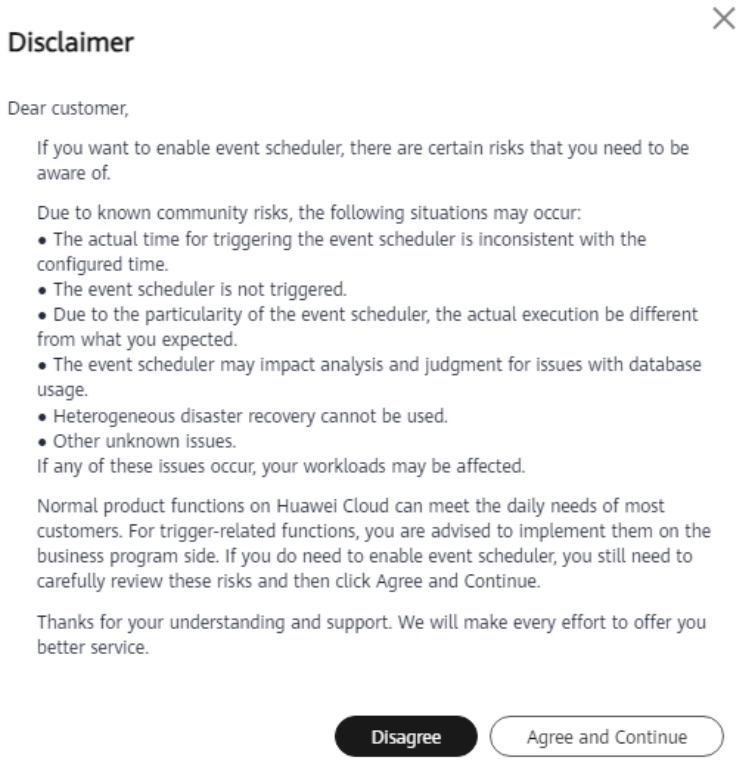
Step 5 Expand **Instance Information**. In the **Configuration** area, click under **Event Scheduler**.

Figure 7-21 Enabling event scheduler



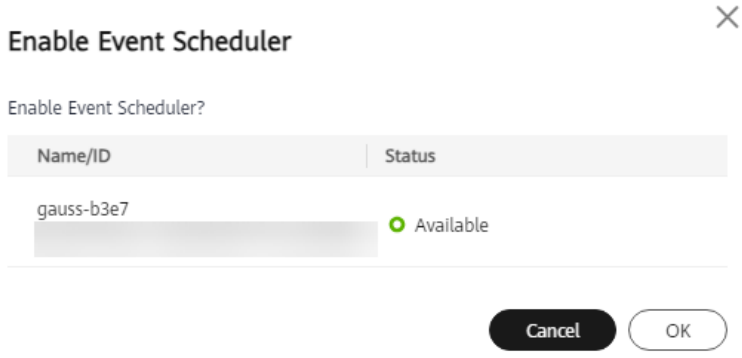
Step 6 In the displayed dialog box, read the disclaimer and click **Agree and Continue**.

Figure 7-22 Reading the disclaimer



Step 7 In the displayed dialog box, confirm the instance information and click **OK**.

Figure 7-23 Confirming information



Step 8 On the **Basic Information** page, expand **Instance Information** and check that the event scheduler status becomes .

----End

Disabling Event Scheduler

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

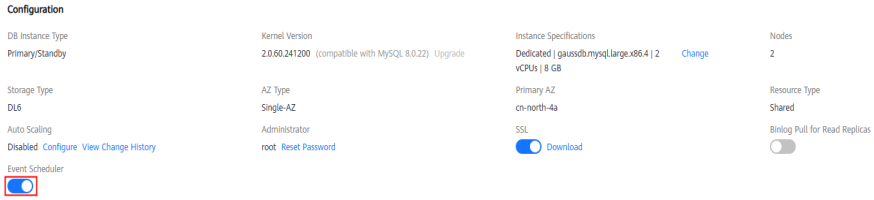
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

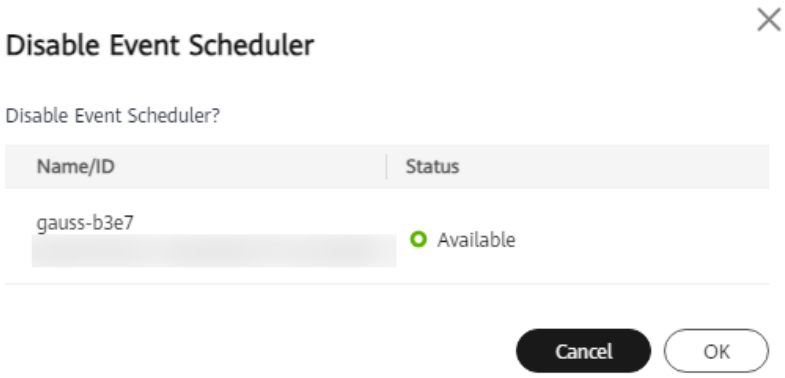
Step 5 Expand **Instance Information**. In the **Configuration** area, click under **Event Scheduler**.

Figure 7-24 Disabling event scheduler



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

Figure 7-25 Confirming information



Step 7 On the **Basic Information** page, expand **Instance Information** and check that the event scheduler status becomes .

----End

7.4.8 Updating the OS of a DB Instance

To improve database performance and security, the OS of your TaurusDB instance needs to be updated timely.

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

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

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

8 Version Upgrades

8.1 Upgrading the Minor Version of a DB Instance

Scenarios

You can upgrade the minor version of your DB instance to improve performance, optimize functions, and fix bugs.

When a new minor version is released on Huawei Cloud, you will see **Upgrade** in the **DB Engine Version** column on the **Instances** page. You can click **Upgrade** to go to the minor version upgrade page.

For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**.

Upgrade Scenarios

Table 8-1 Upgrade scenarios

Upgrade Information	Upgrade Scenario	Description
Kernel version	There are three scenarios for upgrading a minor version based on different kernel versions: • Scenario 1: If the instance's kernel version is below 2.0.51.240305, it will be upgraded to 2.0.51.240305 first, followed by a minor version pre-upgrade and then a minor version upgrade. • Scenario 2: If the instance's kernel version is 2.0.51.240305, it will undergo a minor version pre-upgrade first. After that, you need to click **Upgrade** again to start a minor version upgrade. **The following operations use this upgrade method as an example.** • Scenario 3: If the instance's kernel version is above 2.0.51.240305, clicking **Upgrade** will directly trigger a minor version upgrade.	• If the kernel version is 2.0.51.240305 or earlier, the upgrade involves changes to the underlying data dictionary and may take a long time. • For scenario 2, binlog needs to be enabled before the upgrade. Ensure that the value of **rds_global_sql_log_bin** is **ON**, the value of **binlog_expire_logs_seconds** is at least **86400**, and the value of **default_collation_for_utf8mb4** is **utf8mb4_0900_ai_ci**.
Upgrade time	There are two ways to upgrade a minor version based on different upgrade times. • Upon submission: The system upgrades the minor version upon your manual submission of the upgrade request. • In maintenance window: The system upgrades the minor version during the maintenance window you have specified. For details about how to change the maintenance window, see **Changing the Maintenance Window of a DB Instance**.	If the kernel version of your instance has potential risks or major defects, has expired, or has been brought offline, the system will notify you by SMS message or email and deliver an upgrade task during the maintenance window.

Constraints

Table 8-2 Constraints

Phase	Constraint
Before the upgrade	• To upgrade the kernel version from 2.0.51.240305 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 DB Instance**. • If the **rds_global_sql_log_bin** parameter was recently set to **ON**, you need to connect to the database and run the following command to check whether binlog is enabled for all threads. `select @@session.rds_sql_log_bin_inconsistent_count;` • If the replication latency between the primary node and read replicas is longer than 300 seconds, the minor version cannot be upgraded. • If you want to upgrade the minor version of your instance from 8.0.18 to 8.0.22 and there are more than 1,000 partitions, the upgrade may fail. Contact Huawei Cloud engineers to check the version compatibility before the upgrade.

Phase	Constraint
During the upgrade	• Scenario 1 or 3: If the primary node and read replicas of an 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. • Scenario 1 or 3: The upgrade will cause the instance to reboot and briefly interrupt workloads for about 30 to 90 seconds. To minimize the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection. • Scenario 2: During the minor version pre-upgrade, the instance will not reboot, which means workloads will not be affected. • Scenario 2: The minor version upgrade will cause the instance to reboot, so workloads will be interrupted. The upgrade takes 15 to 20 minutes, and workloads are intermittently interrupted for 1 to 2 minutes. To minimize the impact, perform the upgrade during off-peak hours. • If an instance contains a large number of table partitions (more than 1 million), it may take more than 2 hours to reboot the instance. • When you upgrade the minor version of an instance, minor versions of read replicas (if any) will also be upgraded. During the upgrade, the read replicas will reboot. To minimize the impact, perform the upgrade during off-peak hours. Minor versions of read replicas cannot be upgraded separately. Once you upgrade the minor version, the instance will be upgraded to the latest minor version. A minor version upgrade cannot be rolled back after the upgrade is complete. • DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a minor version upgrade. • If the message "Upgrade failed. Reduce the service load of the instance and click Retry" is displayed on the console, the upgrade fails due to heavy service load. Reduce the service load or retry the upgrade during off-peak hours.

Phase	Constraint
After the upgrade	• Scenario 2: If the upgrade fails, the instance will be rolled back to a previous version and its data will be restored to a new instance named *original_instance_name*_**copy**. The original instance will become unavailable after the rollback, and the new instance will start to incur charges once the original instance is deleted. • Data cannot be restored to a point in time when a data dictionary upgrade was in progress. By default, the system searches for the most recent available point in time instead. • During the period between a data dictionary upgrade and the next full backup, data of the instance cannot be restored to the original instance or an existing instance. • When restoring backups, ensure that the data dictionary version of the target instance is the same as that of the source instance.

Upgrading the Minor Version of a Single DB Instance

Step 1 Log in to the management console.

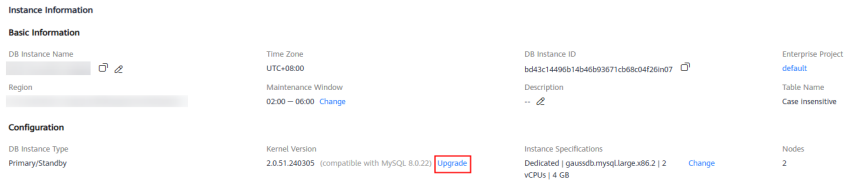
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Instance Information** area, click **Upgrade** under **Kernel Version**.

Figure 8-1 Upgrading the minor version on the **Basic Information** page

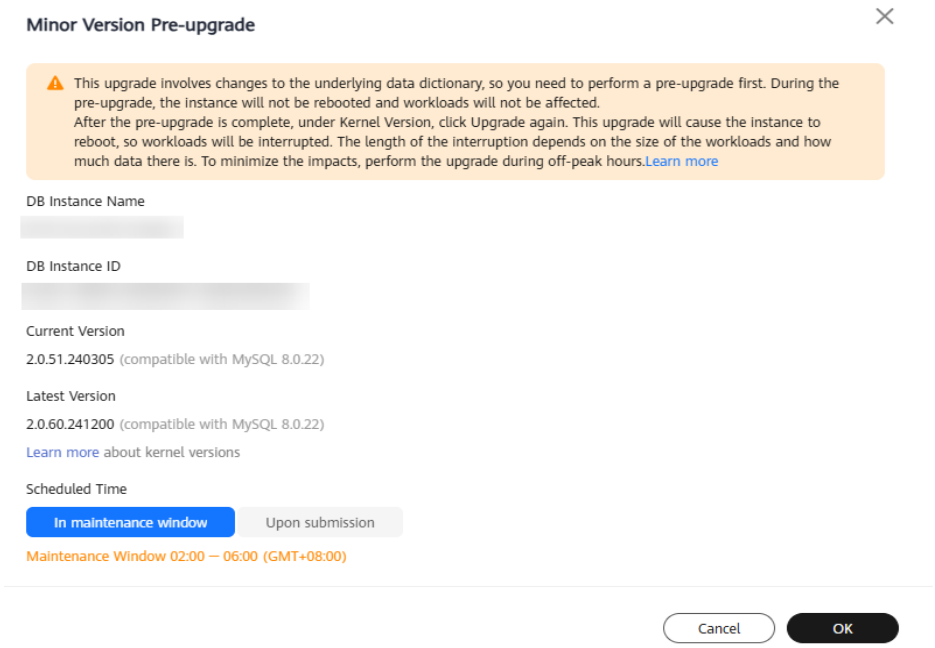


Alternatively, go to the **Instances** page and click **Upgrade** in the **DB Engine Version** column.

Figure 8-2 Upgrading the minor version on the **Instances** page

Step 6 In the displayed dialog box, set **Scheduled Time** and click **OK**.

Figure 8-3 TaurusDB minor version pre-upgrade

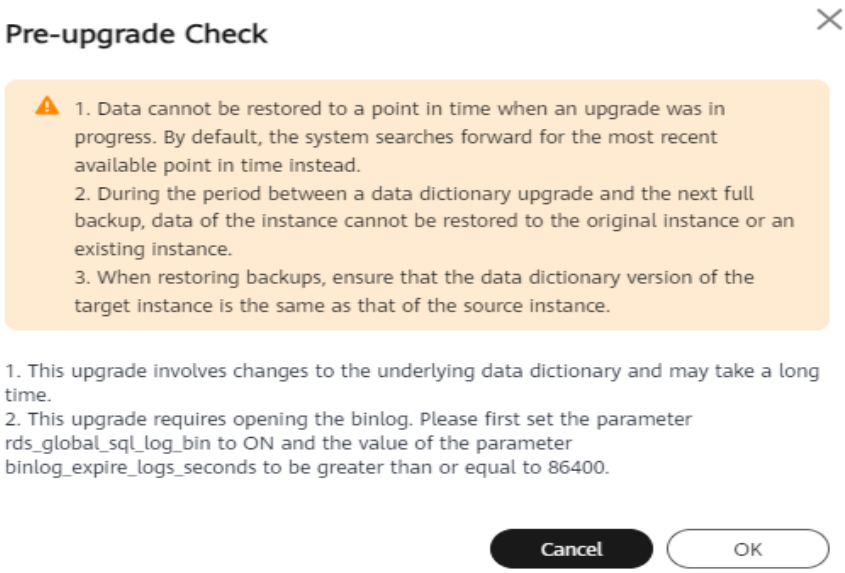


- **Upon submission**: The system upgrades the minor version immediately after your submission of the upgrade request. On the **Task Center** page, click **Instant Tasks** and view the task progress.
- **In maintenance window**: The system upgrades the minor version during the maintenance window you have specified. After the operation is complete, on the **Task Center** page, click **Scheduled Tasks** and view the information about the upgrade task.

Step 7 The system will automatically conduct a pre-upgrade check.

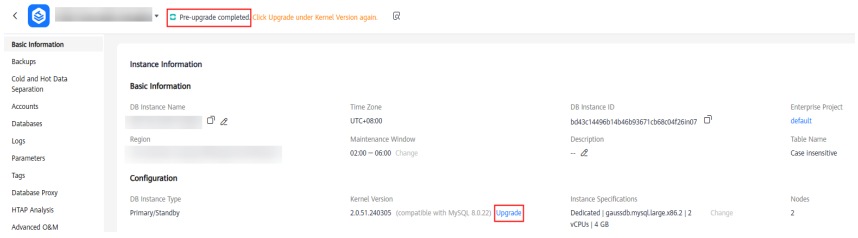
In the displayed dialog box, read the message carefully and click **OK** to proceed with the upgrade.

Figure 8-4 Pre-upgrade check



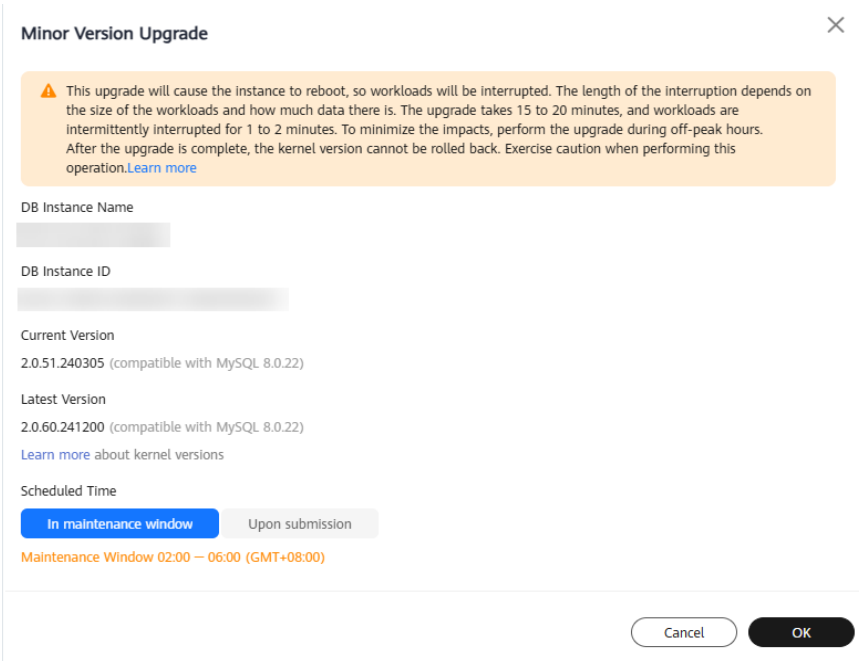
Step 8 After the pre-upgrade is complete, the instance status becomes **Pre-upgrade completed**, which means no other operations are allowed. On the **Basic Information** page, click **Upgrade** under **Kernel Version** again.

Figure 8-5 Upgrading the kernel version



Step 9 In the displayed dialog box, click **OK**. The system will automatically conduct a pre-upgrade check.

Figure 8-6 Minor version upgrade



Step 10 Confirm the check results and click **OK**.

Step 11 After the upgrade is complete, the instance status becomes **Kernel version upgrade to be confirmed**, which means no other operations are allowed. To make the instance available again, go to the **Basic Information** page and click **Apply Upgrade** under **Kernel Version**.

Figure 8-7 Checking the instance status

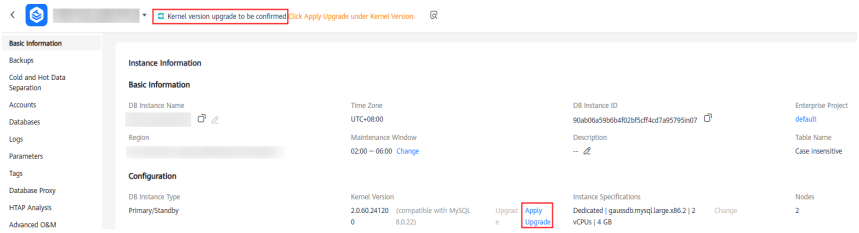
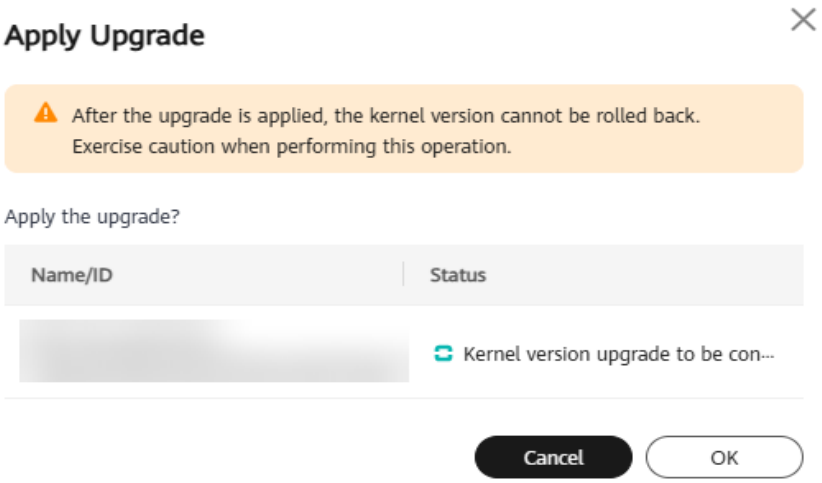


Figure 8-8 Applying the upgrade



----End

Upgrading the Minor Versions of Multiple DB Instances at a Time

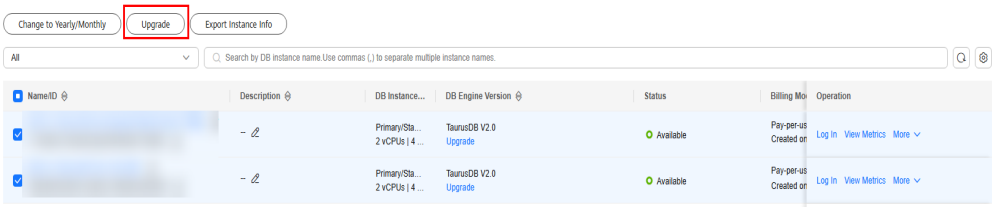
Step 1 **Log in to the management console.**

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, select the desired DB instances and click **Upgrade** in the upper left corner of the list.

Figure 8-9 Upgrading the minor versions of multiple DB instances at a time

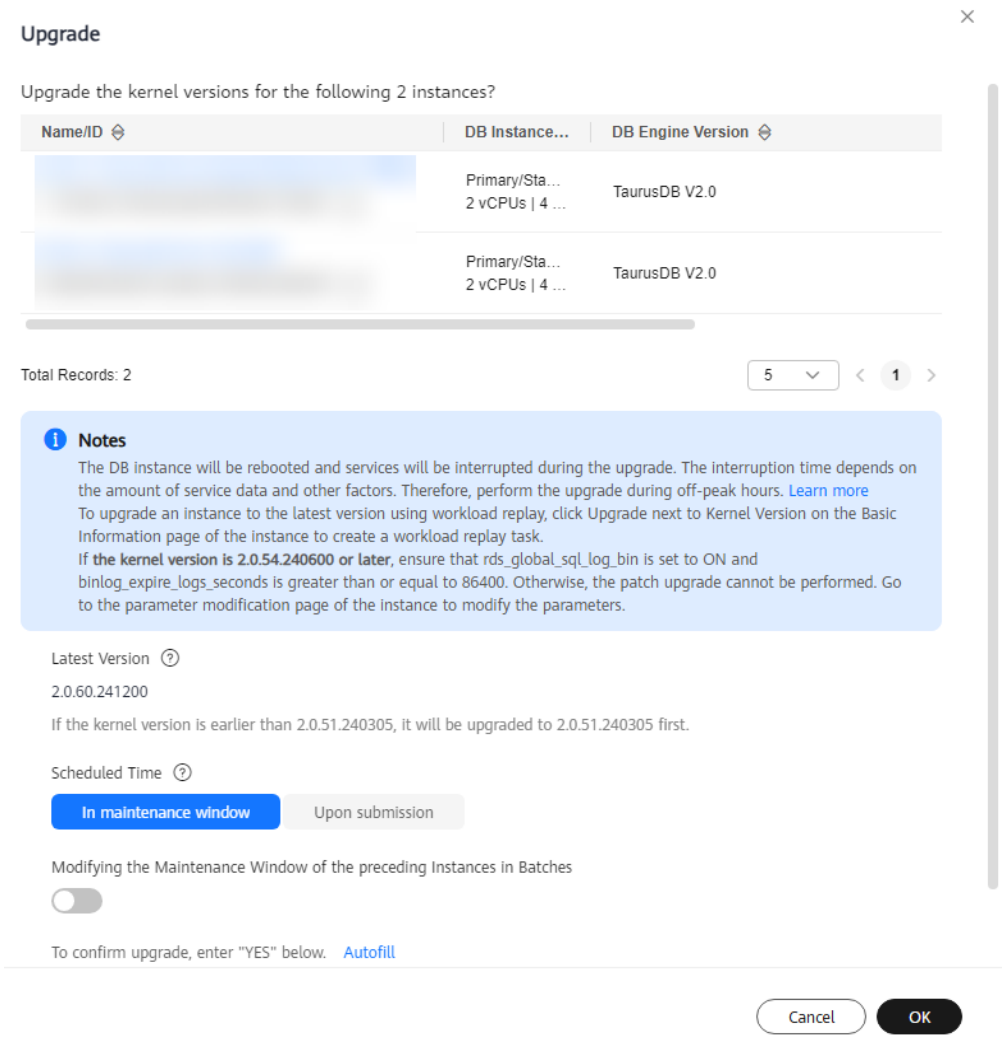


NOTICE

A maximum of 100 DB instances can be selected at a time.

Step 5 In the displayed dialog box, confirm the information about the DB instances to be upgraded and set **Scheduled Time**.

Figure 8-10 Selecting a scheduled time



- **Upon submission**: The system upgrades the minor version immediately after your submission of the upgrade request. On the **Task Center** page, click **Instant Tasks** and view the task progress.
- **In maintenance window**: The system upgrades the minor version during the maintenance window you have specified. After the operation is complete, on the **Task Center** page, click **Scheduled Tasks** and view the information about the upgrade task.

Step 6 Confirm the information, enter **YES** in the text box as prompted, and click **OK**.

CAUTION

- Wait for 2 to 5 minutes and check whether the upgrade has been started for the DB instance. If the upgrade has not been started, check whether the value of **rds_global_sql_log_bin** is **ON** and the value of **binlog_expire_logs_seconds** is at least **86400**. If the parameters are not correctly configured, the upgrade cannot be performed.
- If the parameters are correctly configured but the upgrade has not started, it could be due to that the value of **rds_sql_log_bin_inconsistent_count** is not **0**. Wait until this value becomes **0** before proceeding with the upgrade.

----End

Follow-up Operations

Return to the instance list. In the navigation pane, choose **Task Center** and check the progress of the upgrade task.

- If you have selected **Upon submission** for **Scheduled Time**:

  On the **Instant Tasks** page, search for **Upgrading a DB instance version** and check the execution progress. Instant tasks cannot be canceled.
- If you have selected **In maintenance window** for **Scheduled Time**:

  On the **Scheduled Tasks** page, search for the instance ID and check the execution status of the upgrade task.

  If the task is in the **To be executed** state, you can click **Cancel** to cancel the task.

For details, see **Viewing a Task**.

APIs

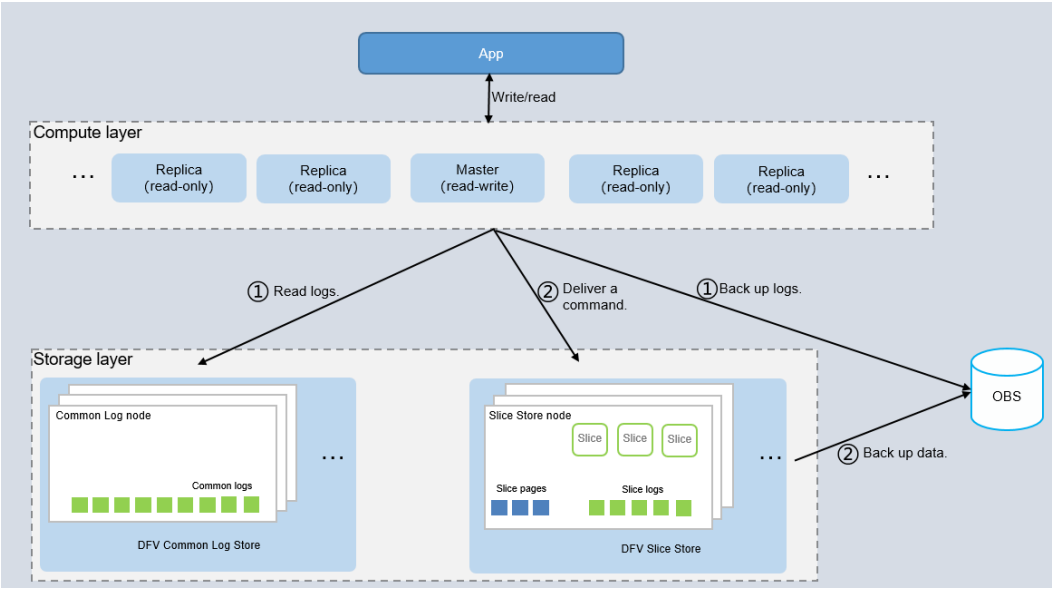
Upgrading the Kernel Version of a DB Instance

9 Data Backups

9.1 Backup Principles

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

Figure 9-1 Backup principles



As shown in **Figure 9-1**, the creation of backups involves in the compute layer and storage layer.

- The primary node at the compute layer reads the logs of the Common Log nodes at the storage layer and backs them up to OBS.
- The primary node at the compute layer delivers a command for backing up data to the Slice Store nodes at the storage layer. The Slice Store nodes back up data to OBS.

During the creation of a backup, the CPU usage and memory usage of the primary node of your instance increase slightly, but you will not notice anything at the storage layer. The final backup is stored in OBS as multiple data files and does not use up any of the disk space of the instance.

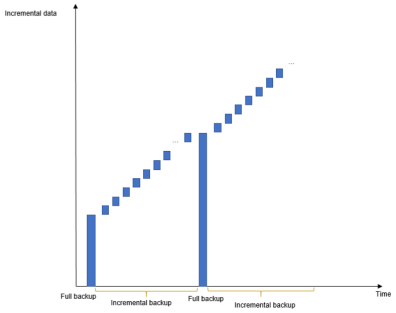
9.2 Backup Types

Based on different dimensions, there are the following backup types in TaurusDB.

- Full backups and incremental backups based on data volume

Table 9-1 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. When there are many writes or the DB instance is overloaded, the upload of incremental backups may be delayed. • Incremental backups are automated backups. • Incremental backups are created based on the most recent full backup, as shown in Figure 9-2, so the most recent full backup that exceeds the retention period is still retained. For details, see the following example. **Figure 9-2** Incremental data restoration 
How to View	Click the instance name. On the **Backups** page, click the **Full Backups** tab and view the backup size.	Click the instance name. On the **Backups** page, click the **Incremental Backups** tab and view the backup size.

- Automated backups and manual backups based on backup methods

Table 9-2 Comparison between automated backups and manual backups

Backup Type	Automated backups	Manual backups

Description	• You can set an automated backup policy on the management console, and the system will automatically back up your instance data based on the time window and backup cycle you set in the backup policy and will store the data for the retention period you specified. • Automated backups cannot be manually deleted. To delete them, you adjust the retention period specified in your backup policy. Retained backups (including full and incremental backups) will be automatically deleted at the end of the retention period.	• Manual backups are user-initiated full backups of your DB instance. They are retained until you delete them manually. • Regularly backing up your DB instance is recommended, so if your DB instance becomes faulty or data is corrupted, you can restore it using backups.
Enabled by Default	Yes	Yes
Retention Period	Automated backups are retained for the retention period you specified. The backup retention period ranges from 1 to 732 days.	Manual backups are always retained until you delete them manually.
Configuration	**Configuring a Same-Region Backup Policy**	**Creating a Manual Backup**

- Same-region backups and cross-region backups based on backup regions

  NOTE

  To configure a cross-region backup policy, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Table 9-3 Comparison between same-region backups and cross-region backups

Backup Type	Same-region backups	Cross-region backups
Description	Backups are stored in the same region as your DB instance.	Backups are stored in a different region from that of your DB instance.
Enabled by Default	Yes	No
Retention Period	Same-region backups are retained for the retention period you specified. The backup retention period ranges from 1 to 732 days. **NOTE** You can request to extend the retention period to up to 3,660 days by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.	Cross-region backups are retained for the retention period you specified. The backup retention period ranges from 1 to 1,825 days.
Characteristic	Backups are stored in the same region as your DB instance. Same-region backup (automated backup) is enabled by default and cannot be disabled.	Backups are stored in a region different from the region where your DB instance is located. After you enable cross-region backup, the backups are automatically stored in the region you specify.
Configuration	**Configuring a Same-Region Backup Policy**	**Configuring a Cross-Region Backup Policy**
How to View	• If cross-region backup is enabled: Click **Backups** in the navigation pane. On the **Same-Region Backups** tab, view the backup size. • If cross-region backup is not enabled: Click **Backups** in the navigation pane and view the backup size.	Click **Backups** in the navigation pane. On the **Cross-Region Backups** tab, locate a DB instance and click **View Cross-Region Backup** in the **Operation** column.

9.3 Backup Space and Billing

Concepts

- Full backup: All data is backed up even if no data has changed since the last backup.
- Incremental backup: The system automatically backs up data that has changed since the last automated backup or incremental backup in logs every 5 minutes. The logs can be used to restore data to a point in time.
- Differential backup: The system backs up data that has changed since the most recent full backup or differential backup in to physical files. Physical files cannot be used for log replay.
- Billed space: backup space that you are billed for
- Logical space: space occupied by full backups
- Physical space: the amount of data that is backed up to OBS

NOTE

After you purchase a DB instance, the logical space is the same as the physical space. When a backup starts in a backup chain, the physical space stores the data of the first full backup and subsequent differential backups.

Backup Space Calculation Methods

There is a default backup chain (where there are seven backups). The first automated backup is a full backup, and subsequent automated backups are differential backups.

In a backup chain, the backup space is released only after all full backups and differential backups are deleted.

Billed space is calculated as follows:

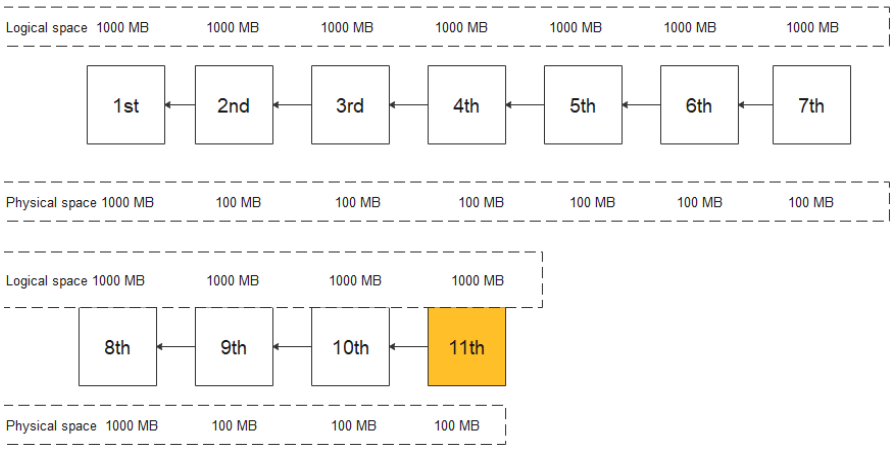
Billed space = Min(Logical space, Physical space) – Free space = Min(Logical space, Physical space) – Storage space x 100%

- Logical space: Total size of the logical space – Logical size of the expired backup
- Physical space: Size of the first full backup + Total size of subsequent differential backups
- Free space: There is free backup storage up to 100% of your purchased storage space.

Example

A backup chain contains seven backups by default. There are 11 backups shown in the following figure. The 1st backup to the 7th belong to one backup chain and the 8th to the 11th belong to another.

Figure 9-3 Backup example



If there is 1,000 MB of backup space and the logical space is 1,000 MB each time, the physical space for the 1st backup is 1,000 MB. If the incremental data size is 100 MB each time, the physical space for the 2nd backup to 7th is 100 MB.

A backup chain contains seven backups by default. The physical space for the 8th backup is 1,000 MB because it represents a new backup chain.

Billed space includes the space of the two chains in the example.

Suppose that after the 11th backup was created, and the 1st, 2nd and 3rd backups expired and were automatically deleted. The size of each space is calculated as follows:

- Total logical space size = Total logical space size – Logical size of the expired backup (1,000 MB x 11 - 3,000 MB = 8,000 MB in this example)
- Physical space: size of data backed up to OBS. In this example, physical space includes the sum of physical space on the two backup links: 1,000 MB + (100 MB x 6) +1,000 MB+ (100 MB X 3) =2,900 MB
- Total billed space = Min (Total size of logical space, Total size of physical space) – free space, so the total billed space in this example = Min (8,000 MB, 2,900 MB) – 1,000 MB = 1,900 MB

9.4 Creating an Automated Backup

9.4.1 Configuring a Same-Region Backup Policy

Scenarios

When you create a TaurusDB instance, the automated backup policy is enabled by default. 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 DB instance level. If a DB instance is faulty or data is damaged, you can still restore it using backups to ensure data reliability. Backing up data affects the database read and write performance, so you are advised to set the automated backup time window to off-peak hours.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

You do not need to configure incremental backup policies because the system automatically performs an incremental backup every 5 minutes. The generated incremental backups can be used to restore the database and table data to a specified point in time.

Constraints

- The automated backup policy is enabled by default and cannot be disabled.
- Rebooting instances is not allowed during the creation of a full backup. Exercise caution when selecting a backup time window.
- When starting a full backup task, 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.

Backup Clearing

To ensure data integrity, even after the retention period expires, the most recent full backup will be retained, for example, if **Backup Cycle** was set to **Monday** and **Tuesday** and **Retention Period** was set to **2**:

- The full backup generated on Monday will be automatically deleted on Thursday because:

  The backup generated on Monday expires on Wednesday, but it was the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday.
- The full backup generated on Tuesday will be automatically deleted on the following Wednesday because:

  The backup generated on Tuesday will expire on Thursday, but as it is the last backup, it will be retained until a new backup expires. The next backup will be generated on the following Monday and will expire on the following Wednesday, so the full backup generated on Tuesday will not be automatically deleted until the following Wednesday.

Viewing or Modifying a Same-Region Backup Policy

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

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

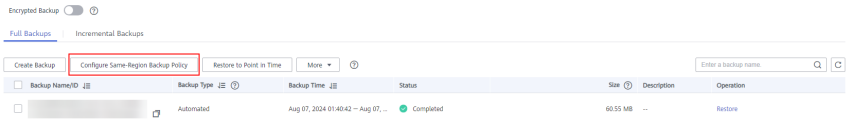
Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

Step 6 On the displayed **Full Backups** tab, click **Configure Same-Region Backup Policy**.

Figure 9-4 Configuring a same-region backup policy



Step 7 In the displayed dialog box, view the current backup policy. To modify the backup policy, adjust the parameter values by referring to **Table 9-4**.

Figure 9-5 Modifying backup policies

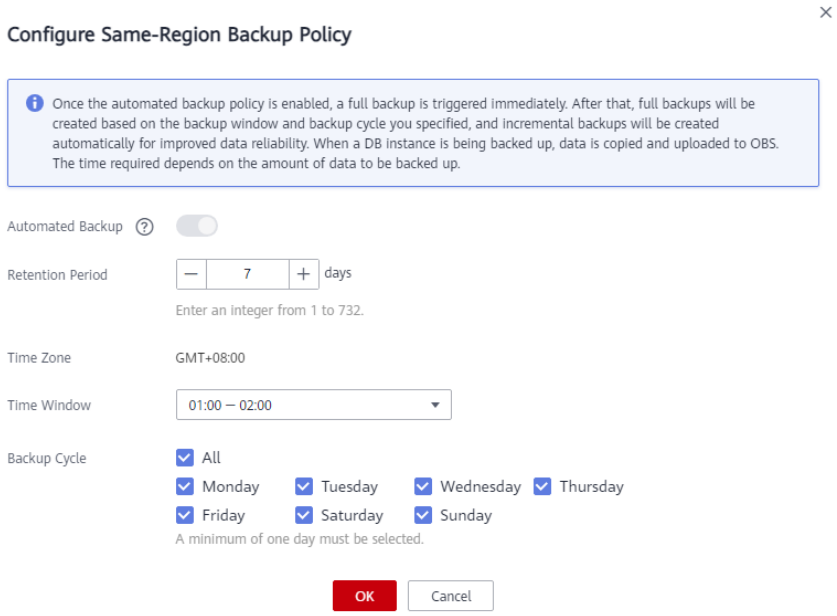


Table 9-4 Parameter description

Parameter	Description
Retention Period	Number of days that your automated backups can be retained. The retention period is from 1 to 732 days and the default value is **7**. • Extending the retention period improves data reliability. You can configure the retention period if needed. • If you shorten the retention period, the new backup policy takes effect for existing backups. Any backups (including full and automated backups) that have expired will be automatically deleted. Manual backups will not be automatically deleted but you can delete them manually. **NOTE** To extend the retention period to up to 3,660 days, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Parameter	Description
Time Zone	The backup time is in UTC format. The backup time segment changes with the time zone during the switch between the DST and standard time.
Time Window	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00.
Backup Cycle	By default, each day of the week is selected. You can change the backup cycle and must select at least one day of the week.

Step 8 Click **OK**.

Step 9 On the **Full Backups** tab, click **Configure Same-Region Backup Policy** again to view the modified backup policy.

----End

FAQs

Why Did My Automated Backup Fail?

APIs

- **Configuring a Same-Region Backup Policy**
- **Querying an Automated Backup Policy**

9.4.2 Configuring a Cross-Region Backup Policy

Scenarios

TaurusDB can store backups in a different region from the DB instance for disaster recovery. If a DB instance in a region is faulty, you can use the backups in another region to restore data to a new DB instance.

After you enable cross-region backup, the backups are automatically stored in the region you specify.

Precautions

- To apply for the permissions to configure cross-region backup policies, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- Cross-region backup can be enabled for up to 150 DB instances in a single region under a tenant. It is recommended that the data volume of a single DB instance be at most 2 TB. If the data volume is too large, the synchronization progress may be delayed.

Supported Regions

Table 9-5 Supported regions

Instance Region	Backup Region
CN North-Beijing4	CN East-Shanghai1, CN North-Ulanqab1, CN Southwest-Guiyang1, and CN South-Guangzhou
CN East-Shanghai1	CN North-Beijing4, CN North-Ulanqab1, CN Southwest-Guiyang1, and CN South-Guangzhou
CN North-Ulanqab1	CN North-Beijing4, CN East-Shanghai1, CN Southwest-Guiyang1, and CN South-Guangzhou
CN Southwest-Guiyang1	CN North-Beijing4, CN East-Shanghai1, CN North-Ulanqab1, and CN South-Guangzhou
CN South-Guangzhou	CN North-Beijing4, CN East-Shanghai1, CN North-Ulanqab1, and CN Southwest-Guiyang1

Billing

Table 9-6 Billing

Specification Code	Pricing
gaussdb.mysql.crossreg.backup.space	• Regions in China (excluding Hong Kong): $0.000157 USD/GB/hour • Hong Kong and regions outside China: $0.00022 USD/GB/hour

Enabling or Modifying a Cross-Region Backup Policy

Step 1 Log in to the management console.

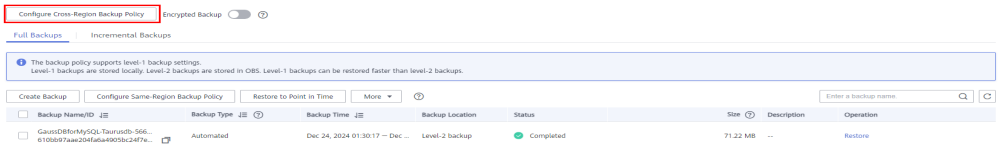
Step 2 Click  in the upper left corner and select a region and project.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Configure Cross-Region Backup Policy**.

Figure 9-6 Configuring a cross-region backup policy



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

Figure 9-7 Configuring a backup policy

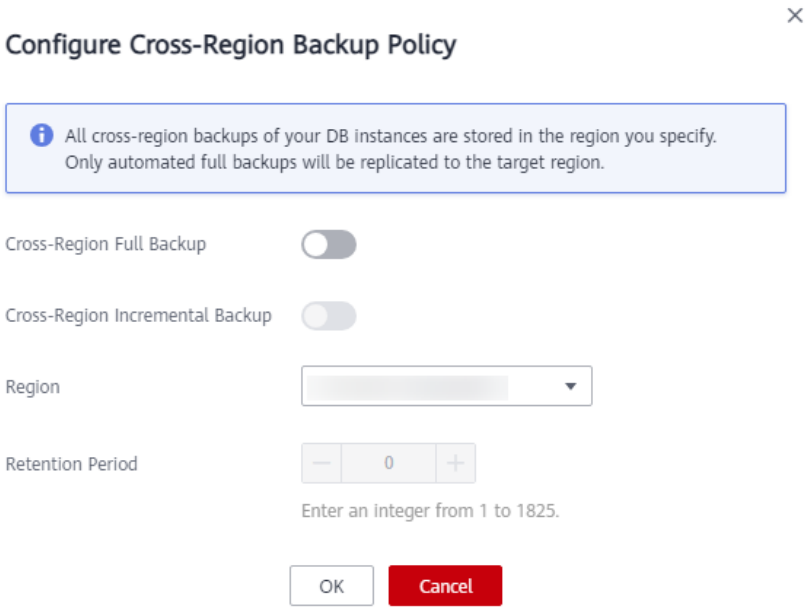


Table 9-7 Parameter description

Parameter	Description
Cross-Region Full Backup	If you enable this option, automated full backups of the DB instance are stored in the region you specify.
Cross-Region Incremental Backup	If you enable this option, incremental backups of the DB instance are stored in the region you specify. • To enable cross-region incremental backup, enable cross-region full backup first. • After cross-region incremental backup is enabled, you can restore an instance to a specified point in time only after the next automated full backup replication is complete. The specified point in time must be later than the time when the automated full backup is complete.
Region	Select the region for storing backups.
Retention Period	Cross-region backup files can be retained from 1 to 1,825 days.

Step 7 Click **OK**.

Step 8 On the **Cross-Region Backups** tab of the **Backups** page, manage cross-region backups.

Figure 9-8 Cross-region backups

- By default, all instances with cross-region backups are displayed.

  To modify the cross-region backup policy, click **Set Cross-Region Backup** in the **Operation** column.

  To view generated cross-region backup files, click **View Cross-Region Backup** in the **Operation** column. If a DB instance fails, you can use the cross-region backups to restore data to a new DB instance.

  Full or incremental backups can be resorted to a new DB instance. Select the backup you want to restore and click **Restore** in the **Operation** column.

  Figure 9-9 Full backups

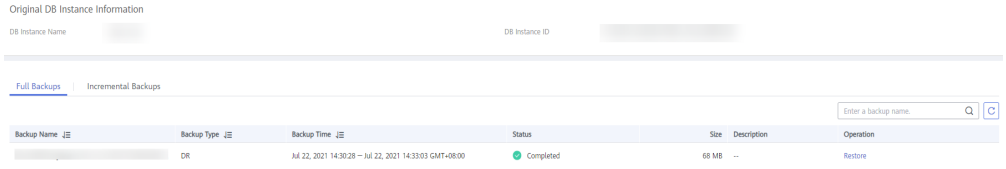
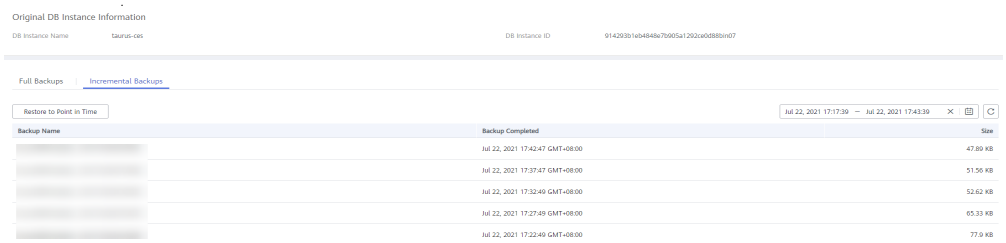
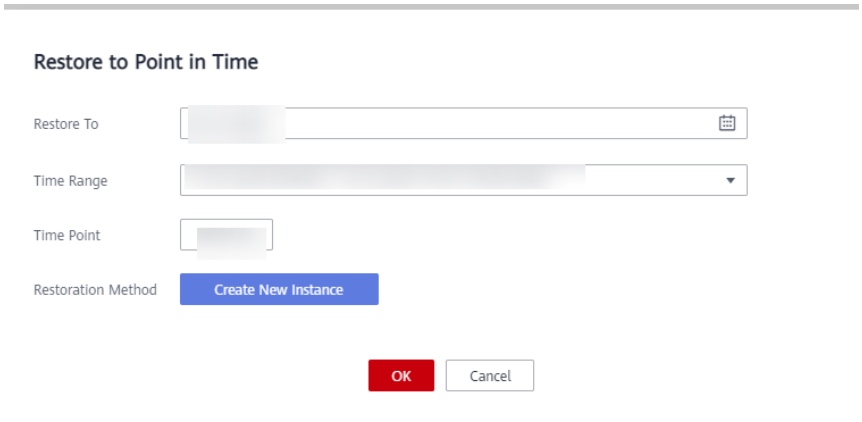
  

  Figure 9-10 Incremental backups

  Incremental backups can be restored to a point in time. You need to select a time range, select or enter a time point within the acceptable range.

  Figure 9-11 Restoring an incremental backup to a point in time

- To view all cross-region backups, click **View All Backups**.

  To restore a backup, locate the backup and click **Restore** in the **Operation**. For details, see **Restoring Data Across Regions**.

- To return to the instance list, click **View Instances**.

----End

Disabling a Cross-Region Backup Policy

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

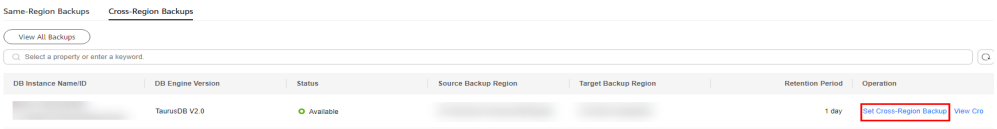
Step 4 In the upper left corner of the page, select the region where the original DB instance is located.

Step 5 Disable the cross-region backup policy using either of the following methods.

Method 1:

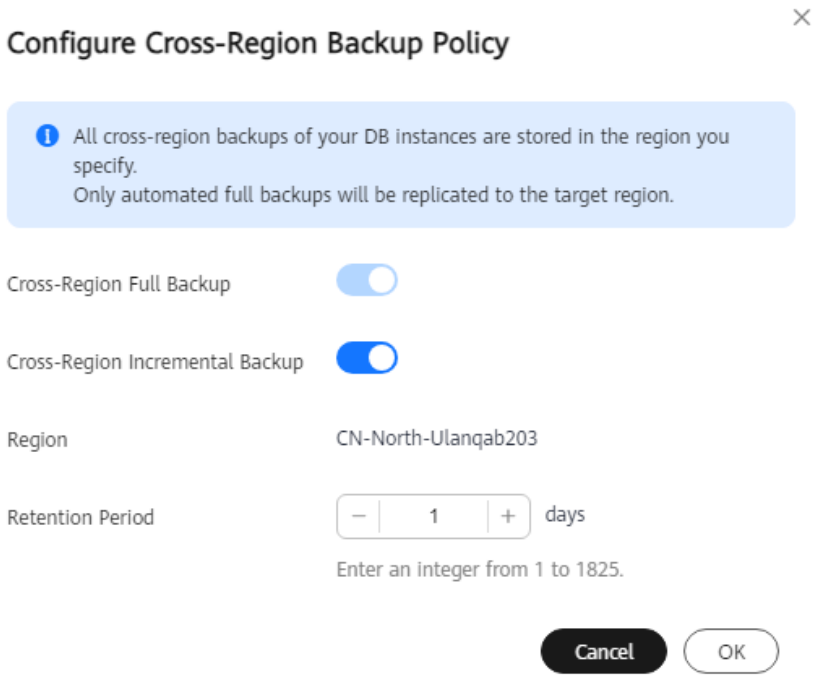
1. Choose **Backups** in the navigation pane and click the **Cross-Region Backups** tab.
2. Locate the target DB instance and click **Set Cross-Region Backup** in the **Operation** column.

Figure 9-12 Setting cross-region backup



3. In the displayed dialog box, click next to **Cross-Region Incremental Backup** and **Cross-Region Full Backup**.

Figure 9-13 Disabling cross-region backup

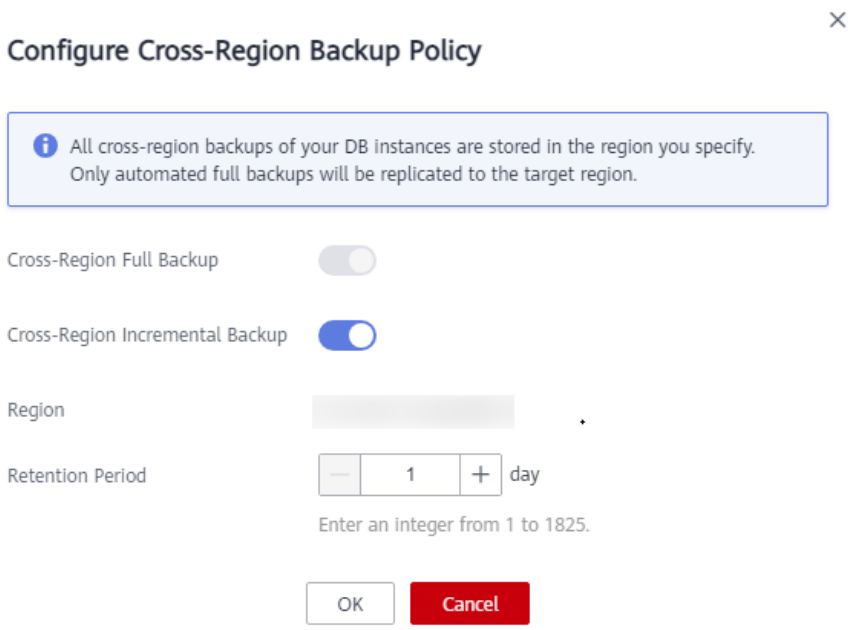


4. Click **OK**.

Method 2:

1. On the **Instances** page, click the instance name.
2. In the navigation pane, choose **Backups**.
3. Click **Configure Cross-Region Backup Policy**.
4. In the displayed dialog box, click next to **Cross-Region Incremental Backup** and **Cross-Region Full Backup**.

Figure 9-14 Disabling cross-region backup



5. Click **OK**.

----End

9.5 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

- You can create manual backups only when your account balance is no less than $0 USD.
- The backup efficiency is in direct proportion to the instance data volume.
- The system verifies the connection to the DB instance when starting a full backup task. If the backup lock failed to be obtained from the DB instance, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
- When an account is deleted, both automated and manual backups are deleted.
- The time required for creating a manual backup depends on the data volume of the DB instance.
- When a DB instance is being backed up, data is copied and uploaded to OBS. The time required depends on the amount of data to be backed up.

Backup Clearing

When a DB instance is deleted, its automated backups are also deleted, but its manual backups are retained until **you manually delete them**.

Creating a Manual Backup

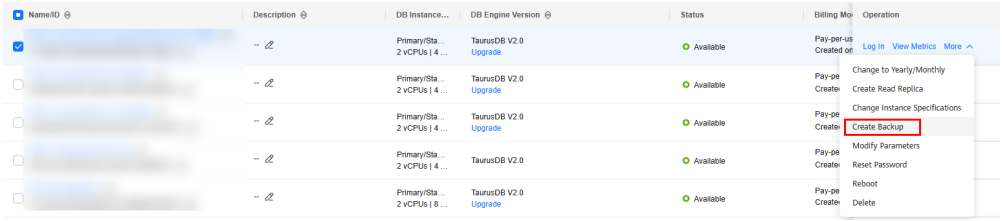
Step 1 **Log in to the management console.**

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **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.

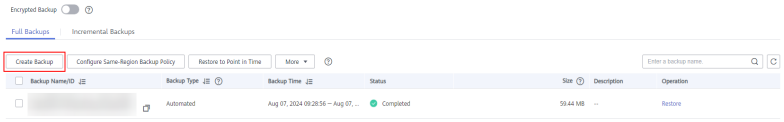
Figure 9-15 Creating a backup



You can also create a backup in either of the following ways:

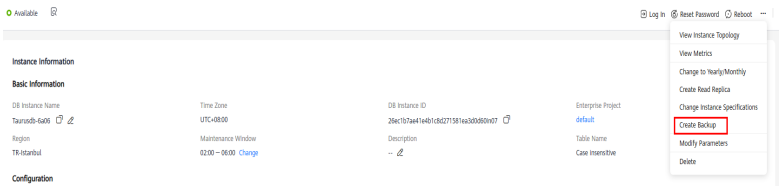
- On the **Instances** page, click the instance name. Choose **Backups** in the navigation pane and click **Create Backup**.

  Figure 9-16 Creating a backup

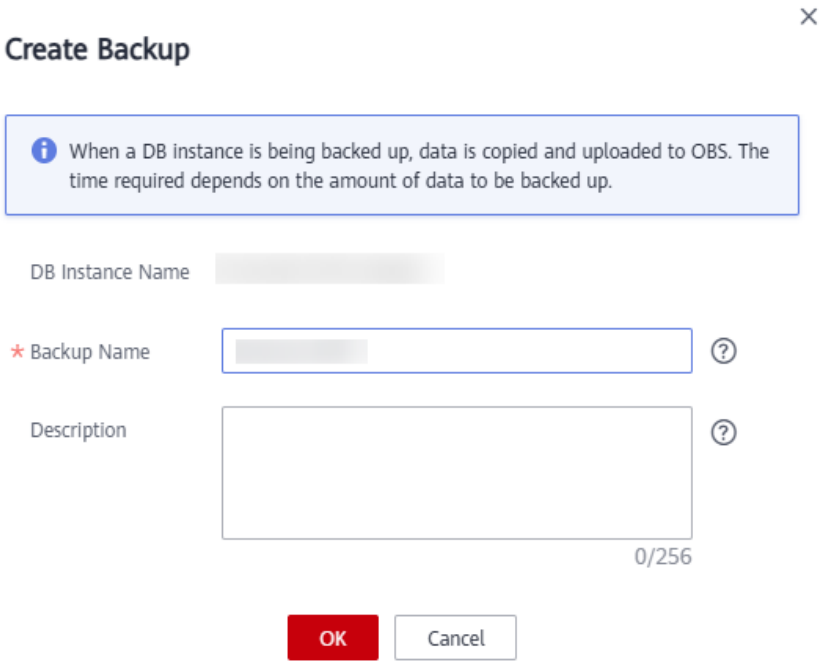
- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click and choose **Create Backup**.

  Figure 9-17 Creating a backup

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

Figure 9-18 Creating a backup



- The backup name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
- The backup description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters: !<''='>&

Step 6 View the created backup on the **Backups** page.

----End

Deleting a Manual Backup

Step 1 On the **Backups** page, locate the backup you want to delete and click **Delete** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name. On the **Backups** page, locate the backup you want to delete and click **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored or created

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

----End

APIs

- Creating a Manual Backup
- Querying Backups

9.6 Enabling or Disabling Encrypted Backup

Scenarios

TaurusDB can encrypt backups. After encrypted backup is enabled, a key is required, which is generated and managed by **Data Encryption Workshop (DEW)**.

Precautions

- Only the backups generated after encrypted backup is enabled will be encrypted.
- After encrypted backup is disabled, new backup files will not be encrypted for storage. Backup files created before encrypted backup is disabled will not be decrypted.
- Currently, only the SM4 and AES_256 key algorithms are supported. After encrypted backup is enabled, the key algorithm cannot be changed.
- The key cannot be disabled, deleted, or frozen while in use, or the encrypted backups cannot be used for restoration.
- Encrypted backups can be directly used to restore data on the management console. You do not need to manually decrypt backups.
- Once encrypted backup is enabled for your DB instance, data cannot be restored to an existing DB instance, even if encrypted backup is disabled later.
- Cross-region backup and encrypted backup cannot be both enabled.
- When encrypted backup is enabled for a DB instance, only the key of the corresponding enterprise project can be selected. To view keys in an enterprise project, see **Viewing a Key**.

Enabling Encrypted Backup

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

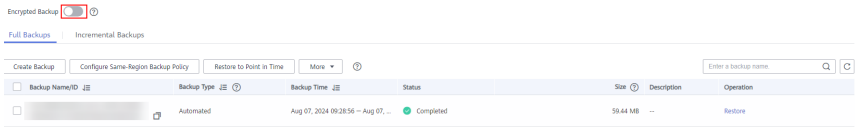
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 Choose **Backups** in the navigation pane and click next to **Encrypted Backup**.

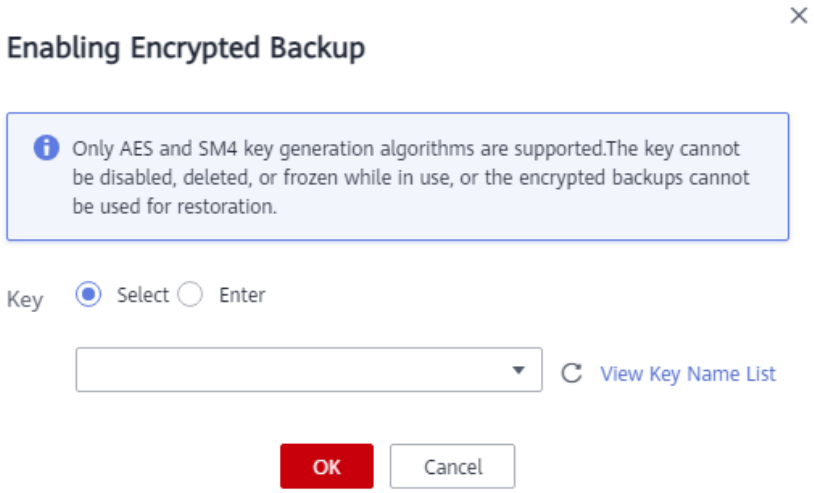
Figure 9-19 Enabling encrypted backup



Step 6 In the displayed dialog box, select a key or enter a key ID and click **OK**.

Only SM4 and AES_256 key algorithms are supported.

Figure 9-20 Selecting a key



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

Step 8 Refresh the page and check whether encrypted backup is enabled.

----End

Disabling Encrypted Backup

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

Step 2 Choose **Backups** in the navigation pane and click next to **Encrypted Backup**.

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

----End

APIs

- **Enabling or Disabling Encrypted Backup**
- **Checking Whether Encrypted Backup Is Enabled**

9.7 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, status, size, and description.

Constraints

Automated and manual backups cannot be downloaded. Backups of TaurusDB are designed based on Huawei Cloud decoupled storage and compute. Data is backed up based on the minimum storage unit of the Huawei-developed distributed storage pool. After the backup data is downloaded, it cannot be parsed and restored using an open-source tool.

Procedure

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

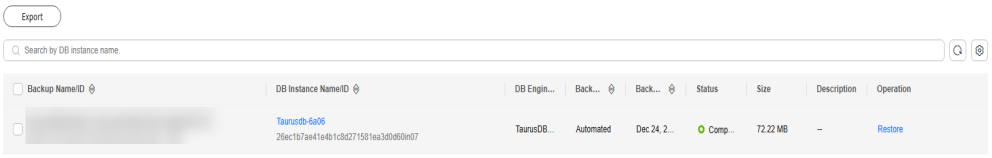
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

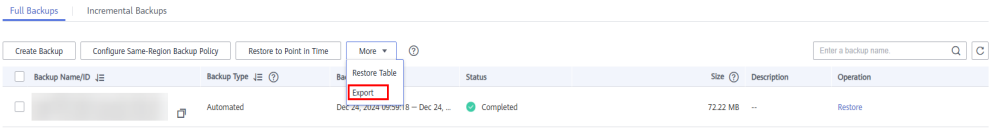
Step 5 On the **Same-Region Backups** tab, select the backups to be exported and click **Export**.

Figure 9-21 Exporting backup information



Alternatively, on the **Instances** page, click the instance name. In the navigation pane, choose **Backups**. On the **Full Backups** tab, select the backups to be exported and choose **More** > **Export**.

Figure 9-22 Exporting backup information



- Currently, only the backup information displayed on the current page can be exported.
- The backup information is exported to an Excel file.

Step 6 View the exported backup information.

----End

10 Data Restorations

10.1 Restoration Schemes

If your instance is damaged or mistakenly deleted, you can restore it using the following methods.

Restoring or Migrating Data to TaurusDB

- You can restore data using backups. For details, see **Restoring a DB Instance from Backups**.
- You can migrate data using DRS, mysqldump, or DAS. For details, see **Data Migration Schemes**.

Restoring Deleted or Modified Data

The following table describes how to restore deleted tables, deleted databases, deleted instances, deleted or modified columns, rows, and data in tables, and overwritten tables.

Table 10-1 Restoration schemes

Scenario	Restoration Solution	Restoration Scope	Restore To	Operation Guide
Restoring a deleted instance	If the deleted instance is in the recycle bin, rebuild it by referring to **Rebuilding a DB Instance in the Recycle Bin**.	All databases and tables	The original instance	**Rebuilding a DB Instance**

Scenario	Restoration Solution	Restoration Scope	Restore To	Operation Guide
	If **a manual backup has been created** before the instance was deleted, restore the instance on the **Backups** page.	All databases and tables	• A new instance • An existing instance • The original instance	**Restoring a DB Instance from Backups**
Restoring a deleted table	Use the database table restoration method to restore the table.	• All databases and tables • Some databases and tables	• A new instance • An existing instance • The original instance	**Restoring Tables to a Point in Time**
Restoring a deleted database	Use the database table restoration method to restore the database.	• All databases and tables • Some databases and tables	• A new instance • An existing instance • The original instance	**Restoring Tables to a Point in Time**
Restoring deleted or modified columns, rows, and data in tables, and overwritten tables	If more than 100,000 data records are deleted or modified, use the database table restoration method to restore data in the table.	• All databases and tables • Some databases and tables	• A new instance • An existing instance • The original instance	**Restoring Tables to a Point in Time**

10.2 Restoring a DB Instance from Backups

Scenarios

You can use an automated or manual backup to restore data to the point in time when the backup was created. The restoration is at the instance level.

A full backup will be downloaded from OBS for restoration. The time required depends on the amount of data to be restored.

Prerequisites

There is an automated or manual backup available for the DB instance. If you want to restore a deleted DB instance, you need to ensure that there is a manual backup because automated backups are deleted along with the DB instance.

Constraints

- Data can be restored to a new, the original, or an existing DB instance.
- Keep your account balance above zero so that backups can be restored to a new DB instance. You will be billed for the new DB instance.
- Data on the original DB instance will be overwritten and the original DB instance will be unavailable during the restoration.
- Restoring to an existing DB instance will overwrite its data and root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.
- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.
- Once encrypted backup is enabled for your DB instance, data cannot be restored to an existing DB instance, even if encrypted backup is disabled later.
- Ensure that the storage space of the selected DB instance is at least that of the original DB instance. Otherwise, data will not be restored.

Restoring to a New DB Instance

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

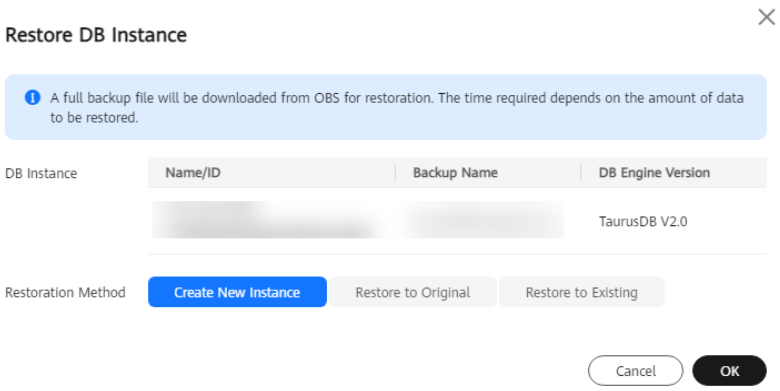
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Backups** page, locate the backup you want to restore and click **Restore** in the **Operation** column.

Step 5 Set **Restoration Method** to **Create New Instance** and click **OK**.

Figure 10-1 Restoring to a new DB instance



Step 6 On the displayed page, set required parameters and click **Next**.

- The region and DB engine version of the new DB instance are the same as those of the original DB instance and cannot be changed.
- The default database port is **3306**.
- Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

Step 7 View the restoration results.

A new DB instance is created, where data is restored based on the point in time when the backup was created. When the instance status changes from **Creating** to **Available**, the restoration is complete.

The new DB instance is independent from the original one. If you want to offload the read load from the primary node, create one or more read replicas for the new DB instance.

After the restoration, a full backup will be automatically triggered.

----End

Restoring to the Original DB Instance

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

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

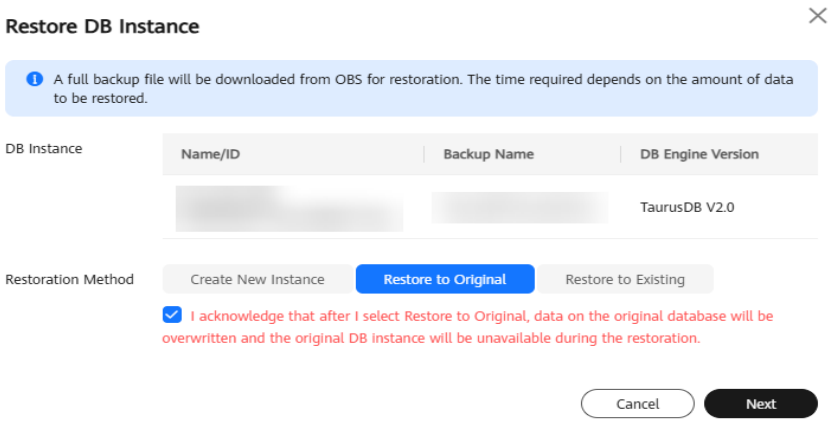
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Backups** page, locate the backup you want to restore and click **Restore** in the **Operation** column.

Step 5 Select where you want to restore the backup to.

Step 6 Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.

Figure 10-2 Restoring to the original DB instance



Step 7 Confirm the task details and click **OK**.

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

Step 8 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete.

After the restoration, a full backup will be automatically triggered.

----End

Restoring to an Existing DB Instance

Step 1 Log in to the management console.

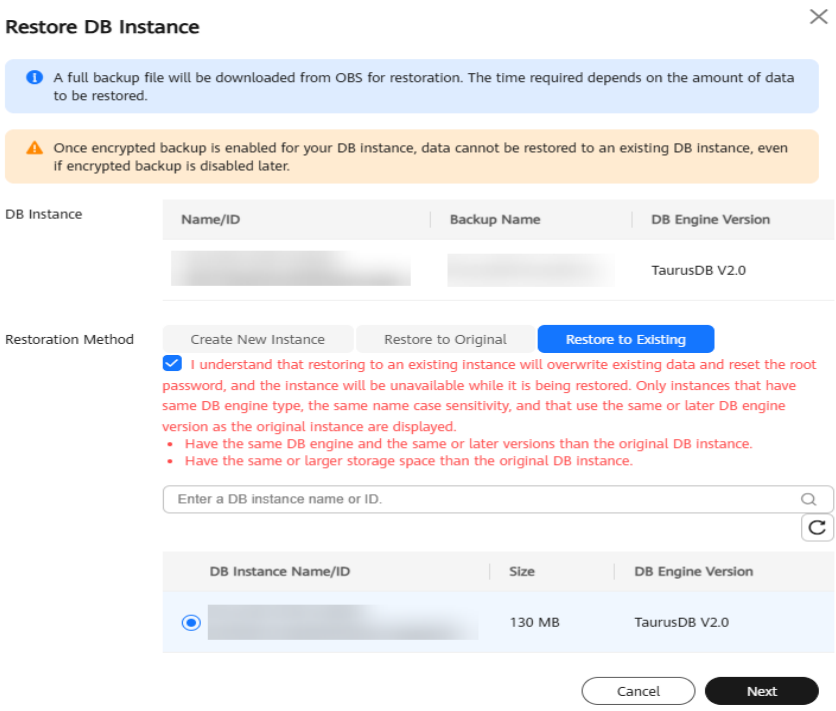
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Backups** page, locate the backup you want to restore and click **Restore** in the **Operation** column.

Step 5 Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, and click **Next**.

Figure 10-3 Restoring to an existing DB instance



Step 6 Confirm the task details and click **OK**.

- Restoring to an existing DB instance will overwrite its data and root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.
- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.

Step 7 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. If the existing DB instance contains read replicas, the read replica status is the same as the existing DB instance status.

After the restoration, a full backup will be automatically triggered.

----End

APIs

- **Restoring Data to the Original Instance or an Existing Instance**
- **Querying the Restoration Time Range**

10.3 Restoring a DB Instance to a Point in Time

Scenarios

You can restore instance data to a specified point in time. Data can be restored to a new, the original, or an existing DB instance.

The most recent full backup will be downloaded from OBS for restoration. After the restoration is complete, incremental backups will be replayed to the specified point in time. The time required depends on the amount of data to be restored.

Precautions

- Data can be restored to a new, the original, or an existing DB instance.
- Keep your account balance above zero so that backups can be restored to a new DB instance. You will be billed for the new DB instance.
- Do not run the **reset master** command on DB instances within their lifecycle. Otherwise, an exception may occur when restoring a DB instance to a specified point in time.
- Data on the original DB instance will be overwritten and the original DB instance will be unavailable during the restoration.
- Restoring to an existing DB instance will overwrite its data and root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.
- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.
- Once encrypted backup is enabled for your DB instance, data cannot be restored to an existing DB instance, even if encrypted backup is disabled later.

- Ensure that the storage space of the selected DB instance is at least that of the original DB instance. Otherwise, data will not be restored.

Restoring to a New DB Instance

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

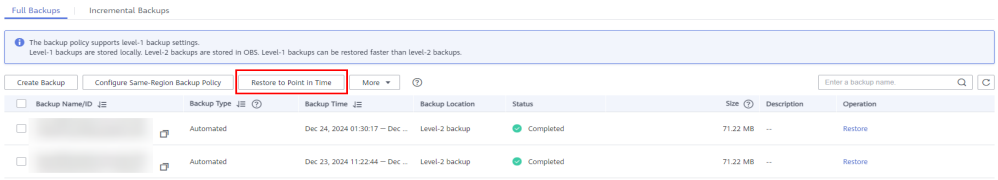
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

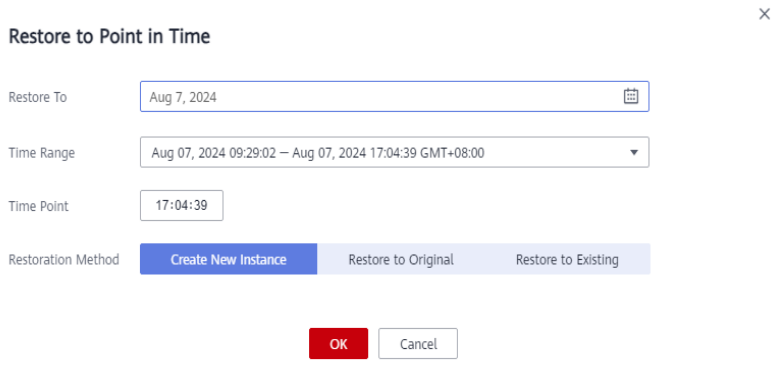
Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Restore to Point in Time**.

Figure 10-4 Restoring to a point in time



Step 6 Select the restoration date and time range, enter a time point within the selected time range, and set **Restoration Method** to **Create New Instance**. Then, click **OK**.

Figure 10-5 Restoring to a new DB instance



Step 7 On the displayed page, set required parameters and click **Next**.

- The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.
- The default database port is **3306**.
- To synchronize database parameters of the original DB instance, select **Original DB instance parameter template**.

> **CAUTION**
>
> If the original DB instance is deleted, the database parameters of the original DB instance cannot be synchronized using backups.
>
> When you synchronize the database parameters of the original DB instance, the following parameters cannot be synchronized and you need to **manually modify them** after the DB instance is restored.
>
> **innodb_write_io_threads**, **innodb_read_io_threads**, **max_connections**, **innodb_log_buffer_size**, **innodb_parallel_read_threads**, **temptable_max_ram**, **threadpool_size**, **innodb_buffer_pool_size**, and **innodb_page_cleaners**

- Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

Step 8 View the restoration results.

A new DB instance is created, where data is restored based on the point in time when the backup was created. When the instance status changes from **Creating** to **Available**, the restoration is complete. The new DB instance is independent from the original one. If you want to offload the read load from the primary node, create one or more read replicas for the new DB instance.

After the restoration, a full backup will be automatically triggered.

----End

Restoring to the Original DB Instance

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

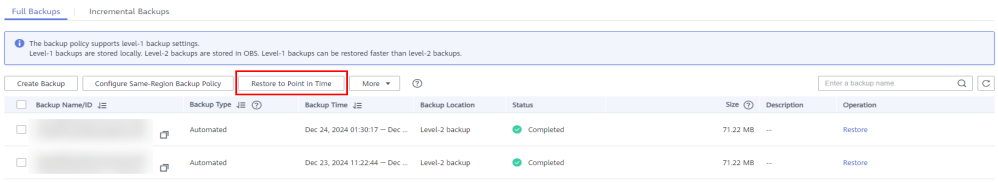
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

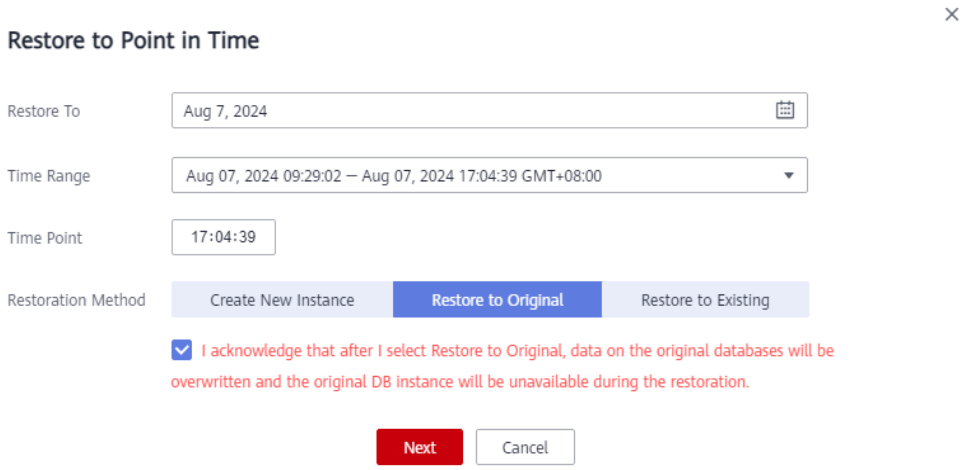
Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Restore to Point in Time**.

Figure 10-6 Restoring to a point in time



Step 6 Select the restoration date and time range, enter a time point within the selected time range, set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.

Figure 10-7 Restoring to the original DB instance



Step 7 Confirm the task details and click **OK**.

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

Step 8 View the restoration results.

When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

Restoring to an Existing DB Instance

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

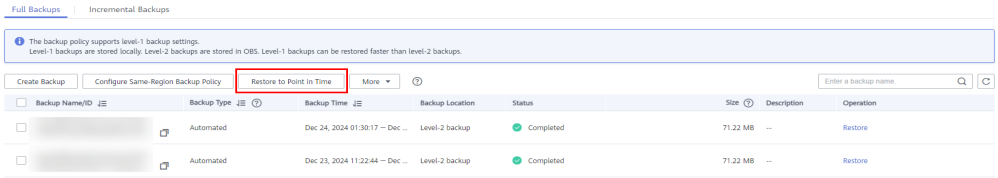
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

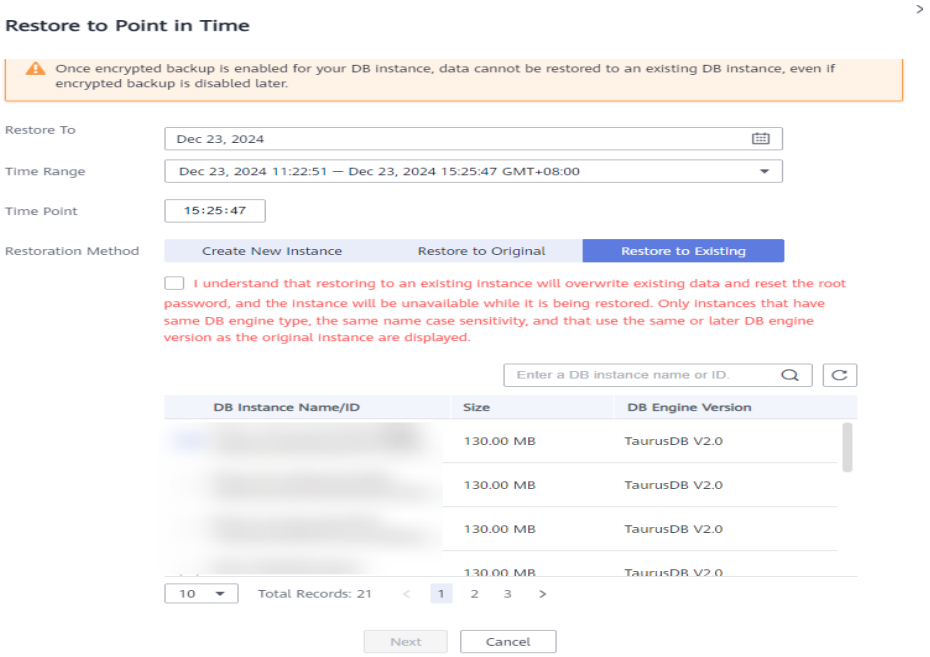
Figure 10-8 Restoring to a point in time



Step 6 Select the restoration date and time range, and enter a time point within the selected time range.

Step 7 Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, and click **Next**.

Figure 10-9 Restoring to an existing DB instance



Step 8 Confirm the task details and click **OK**.

- Restoring to an existing DB instance will overwrite its data and root password. The existing DB instance is unavailable during the restoration. DB instances will not be displayed unless they have the same DB engine type, version, and table name case sensitivity as the original DB instance.
- The restored DB instance contains the data and account information of the original DB instance, but does not contain the parameter settings of the original DB instance.
- If the original password of the existing DB instance cannot be used to connect to the database after the restoration, you can reset the password.

Step 9 View the restoration results.

When the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

10.4 Restoring Tables to a Point in Time

Scenarios

To ensure data integrity and reduce impact on the original instance performance, the system restores the full and incremental data at the selected point in time to a temporary instance, exports the tables to be restored, and then restores the tables to the original instance. The time required depends on the amount of data to be backed up and restored on the instance. Restoring tables will not overwrite data in the instance.

Constraints

- Tables that have triggers cannot be restored.
- To prevent restoration failures and impact on original data, table-level restoration removes foreign key constraints.
- If the tables to be restored are not found at the selected point in time, the restoration will fail.
- The DB instance cannot be rebooted or deleted, and the instance specifications cannot be modified.
- The number of tables to be restored must be no more than 20,000.
- If the number of tables to be restored exceeds 2,000, you are advised to restore the DB instance to a point in time. For details, see **Restoring a DB Instance to a Point in Time**.

Restoring Tables to a Point in Time

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

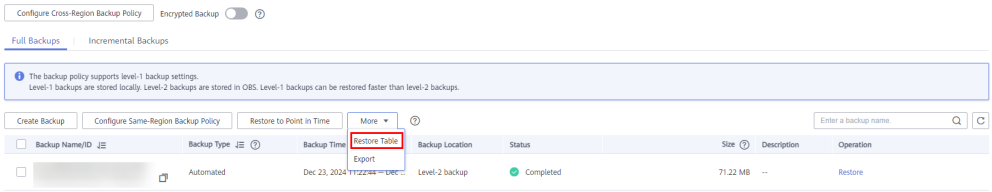
Step 2 Click in the upper left corner and select a region and project.

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.

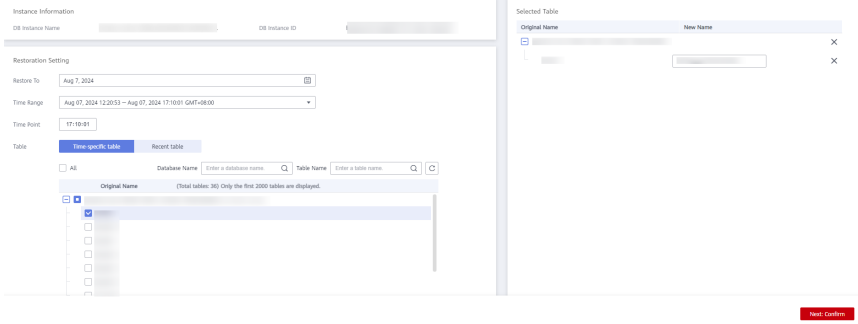
Step 5 In the navigation pane, choose **Backups**. On the **Full Backups** tab, choose **More** > **Restore Table** above the backup list.

Figure 10-10 Restoring tables to a specified point in time



Step 6 On the displayed page, set the restoration date, time range, time point, and tables to be restored.

Figure 10-11 Setting required parameters



- To facilitate your operations, you can search for the tables and databases to be restored.
- After the restoration is complete, new tables with timestamps as suffixes are generated in the instance. You can rename the new tables. The new table name must be unique. It can contain up to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.
- **Time-specific table**: The tables to be restored are read from the latest full backup before the selected time point. **Recent table**: The tables to be restored are read from the selected time point.
- Tables created after the latest full backup are not displayed in the time-specific table list. You can select **Recent table** to view the latest table details.
- If the tables to be restored are not found or are deleted by mistake, you need to log in to the databases and create tables with the same names. Then, the tables to be restored will be displayed when you select **Recent table**.
- Only specified tables are restored. Ensure that all tables to be restored are selected.

Step 7 Click **Next: Confirm**. On the displayed page, confirm the information about the tables to be restored and click **Restore Now**.

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

Step 8 On the **Instances** page, view the instance status, which is **Restoring**. During the restoration process, services are not interrupted.

You can also view the progress and result of restoring tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the tables as required.

----End

10.5 Restoring Data Across Regions

Scenarios

TaurusDB can store backups in a different region from your DB instance. If your DB instance is faulty, you can use a backup to restore data to a new DB instance in the region where the backup is stored.

Prerequisites

A cross-region backup has been created. For details, see **Configuring a Cross-Region Backup Policy**.

Restoring a Full Backup Across Regions

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

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

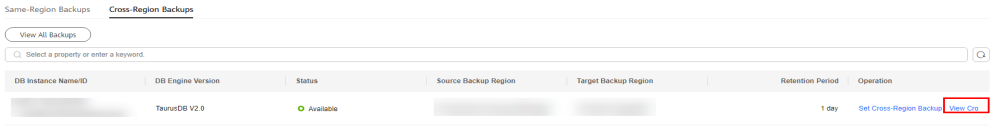
Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 In the upper left corner of the page, select the region where the backup is located.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click the **Cross-Region Backups** tab.

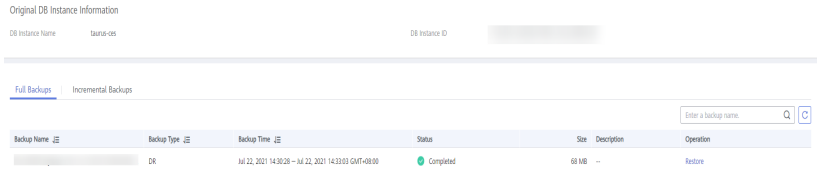
Step 6 Locate the target DB instance and click **View Cross-Region Backup** in the **Operation** column.

Figure 10-12 Viewing cross-region backups



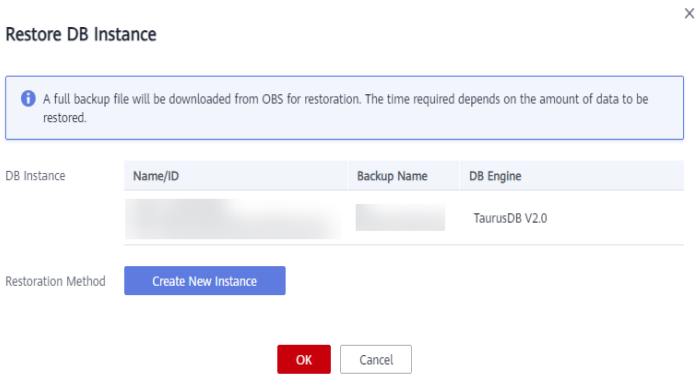
Step 7 On the **Full Backups** page, locate the backup you want to restore and click **Restore** in the **Operation** column.

Figure 10-13 Restoring a full backup



Step 8 In the displayed dialog box, confirm instance details and click **OK**.

Figure 10-14 Restoring a full backup to a new DB instance



Step 9 On the displayed page, set the parameters of the new DB instance and click **Next**.

- The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.
- The default database port is **3306**.

- Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

----End

Restoring an Incremental Backup Across Regions

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

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

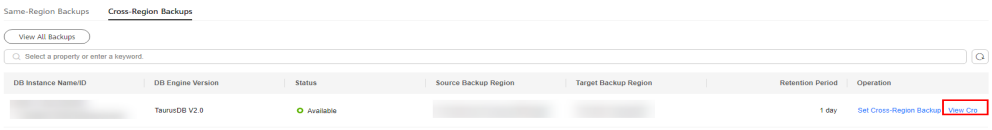
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 In the upper left corner of the page, select the region where the backup is located.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click the **Cross-Region Backups** tab.

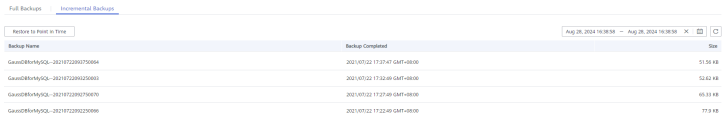
Step 6 Locate the target DB instance and click **View Cross-Region Backup** in the **Operation** column.

Figure 10-15 Viewing cross-region backups



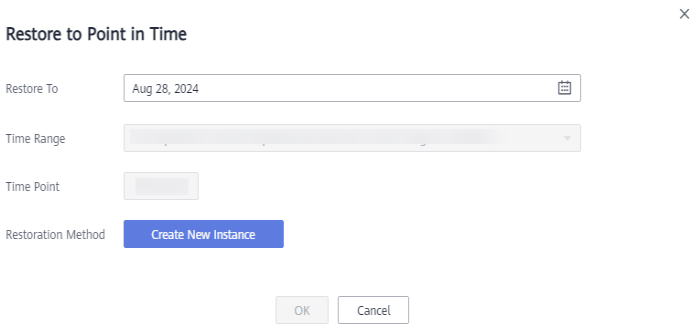
Step 7 On the **Incremental Backups** page, click **Restore to Point in Time**.

Figure 10-16 Restoring an incremental backup



Step 8 Select the date and time range, and select or enter a time point within the time range.

Figure 10-17 Restoring an incremental backup to a point in time



Step 9 Click **OK**.

Step 10 On the displayed page, set the parameters of the new DB instance and click **Next**.

- The region, DB engine, and DB engine version are the same as those of the original DB instance and cannot be changed.
- The default database port is **3306**.
- Other settings are the same as those of the original DB instance by default but can be modified. For details, see **Buying a DB Instance**.

----End

11 Serverless Instances

11.1 What Is a Serverless Instance?

Context

The stability and reliability of databases are crucial for enterprise-grade IT systems. If a database is not stable, the entire system cannot run properly. To ensure smooth database operation during peak hours, users typically configure various parameters and redundant resources (such as compute, memory, and storage).

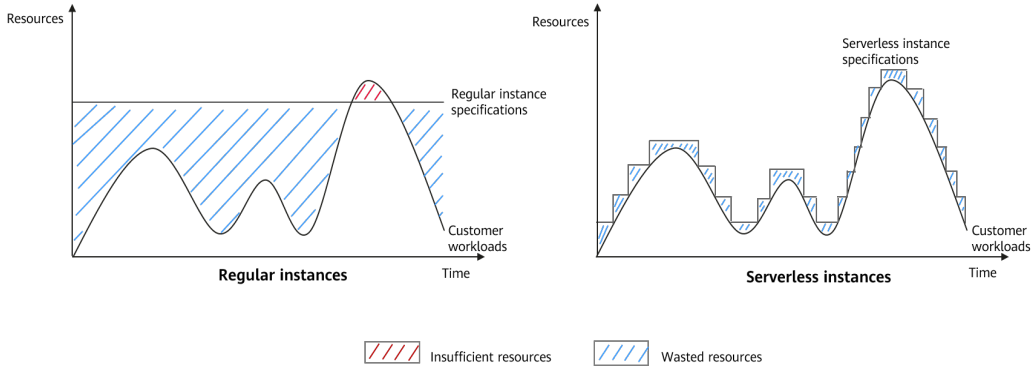
However, during off-peak hours, those redundant resources are often left idle, resulting in wasted costs. Even with those configurations, there is still a risk of temporary resource shortages in the face of unexpected surges in workloads, which can compromise the overall system.

Apart from the typical enterprise users, there are also many users who occasionally use small-scale databases only for development and testing, applet development, and school laboratory teaching. Those users often have minimal specification requirements but demand workload continuity. Constantly creating or deleting pay-per-use instances is not feasible, and buying low-spec yearly/monthly instances results in a significant waste of money when there are no workloads to process.

To address those concerns, TaurusDB has introduced serverless instances. These instances can dynamically adjust resources based on workloads and are billed on a pay-per-use basis, helping customers speed up data processing at lower costs. Additionally, serverless instances make it easier for small- and medium-sized enterprises to use cloud databases.

The following figure shows the resource usage and specification changes of regular and serverless instances during significant workload fluctuations.

Figure 11-1 Resource usage and specification changes of regular and serverless instances



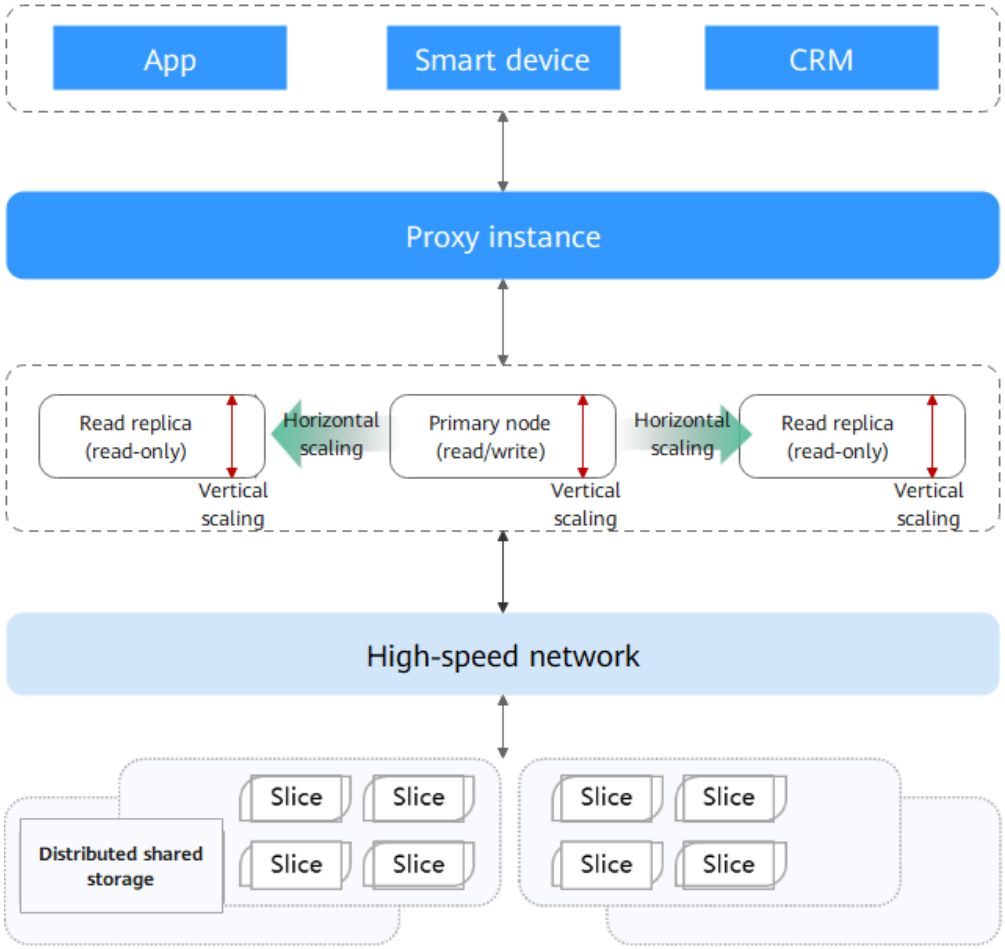
As shown in the figure, regular and serverless instances perform differently during significant workload fluctuations.

- Regular instances: Resources are wasted during off-peak hours and insufficient during peak hours, which will affect workloads.
- Serverless instances: The specifications are adjusted based on workload demands to achieve minimal resource wastes. Even during peak hours, workload demands can still be met, ensuring workload continuity and improving system stability.

How a Serverless Instance Works

TaurusDB serverless instances use a write once read many (WORM) architecture and shared storage. They provide the ability to dynamically scale with system workloads. Each instance node can vertically scale CPUs and memory in seconds and horizontally scale read replicas. It means that compute can quickly and independently adapt to the peaks and troughs, achieving high cost-effectiveness.

Figure 11-2 Serverless architecture



- Both the primary node and read replicas are serverless. They use distributed shared storage and can be scaled based on workload changes.
- The billing unit is TaurusDB Capacity Unit (TCU). 1 TCU is approximately equal to 1 CPU and 2 GB of memory. When the primary node or a read replica is scaled, its TCUs increase or decrease accordingly.
- When creating a serverless instance, you can specify a TCU range, instead of configuring specific specifications. Then the instance can be scaled based on the CPU usage and memory usage.

  Vertical scaling: The node performance (CPU and memory specifications) changes.

  Cloud Eye monitors the CPU usage and memory usage of serverless instances. If any of the following conditions is met, a scale-up is automatically triggered:

  - The CPU usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up.
  - The memory usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up.
  - The CPU usage is greater than 60% for 20 seconds and it has been at least 10 seconds since the last scale-up.

  If the following condition is met, a scale-down is automatically triggered:

The CPU usage is less than 30% for 15 seconds, it has been at least 15 seconds since the last scale-down, and the memory usage is 80% or less.

Horizontal scaling: The number of read replicas changes.

If the compute of all read replicas has already been scaled up as much as possible but the CPU or memory usage still meets a compute scale-up condition, read replicas will be added.

If the compute of all read replicas has already been scaled down as much as possible but the CPU or memory usage still meets a compute scale-down condition, read replicas will be removed.

Billing

For details, see **Serverless Billing**.

Advantages

- Lower cost: TaurusDB serverless instances do not depend on other infrastructure or related services. They can be used right out of the box and provide stable and efficient data access services. You are only billed for the resources you use.
- Larger storage space: The storage space of a serverless instance can reach up to 32,000 GB. It can scale up if the data volume of the instance increases, avoiding impacts on workloads due to insufficient storage resources.
- Auto scaling of compute resources: Compute resources required for read and write operations can flexibly scale, greatly reducing O&M costs and system risks.
- Fully managed and O&M-free experience: All O&M tasks, such as specification scaling, storage autoscaling, monitoring and alarms, and intelligent O&M, are completed by Huawei Cloud professional teams, providing you with a truly O&M-free experience. You will not even notice, and your workloads will not be affected.

Use Cases

- Databases are infrequently used, such as for enterprise testing and individual developers.
- There are intermittent scheduled tasks to be executed, such as data statistics and archiving, school teaching, and R&D tasks.
- There are unpredictable fluctuations in workloads, such as check-in and edge computing.
- An O&M-free experience or fully managed database is required.
- Database costs need to be reduced during off-peak hours.

11.2 Changing the Compute Range

After **buying a serverless instance**, you can change its compute range. When certain trigger conditions are met, instance compute is automatically changed.

Trigger Conditions for Compute Changes

Cloud Eye monitors the CPU usage and memory usage of serverless instances. If any condition listed in **Table 11-1** is met, a compute change will be triggered.

Table 11-1 Trigger conditions for compute changes

Change Type	Trigger Condition
Compute scale-up	If any of the following conditions is met, a compute scale-up is automatically triggered: • The CPU usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up. • The memory usage is greater than 80% for 5 seconds and it has been at least 5 seconds since the last scale-up. • The CPU usage is greater than 60% for 20 seconds and it has been at least 10 seconds since the last scale-up.
Compute scale-down	The CPU usage is less than 30% for 15 seconds, it has been at least 15 seconds since the last scale-down, and the memory usage is 80% or less.

Constraints

- As data grows, there may be some cache or memory fragments that cannot be released, leading to high memory usage. If you want to reduce compute to the minimum, you can reboot the instance.
- If resources are insufficient when you change the compute change, nodes with the desired specifications will be created on a physical machine that has enough resources. If resources on the primary node are insufficient, a primary/standby failover will be performed.

Changing the Compute Range

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

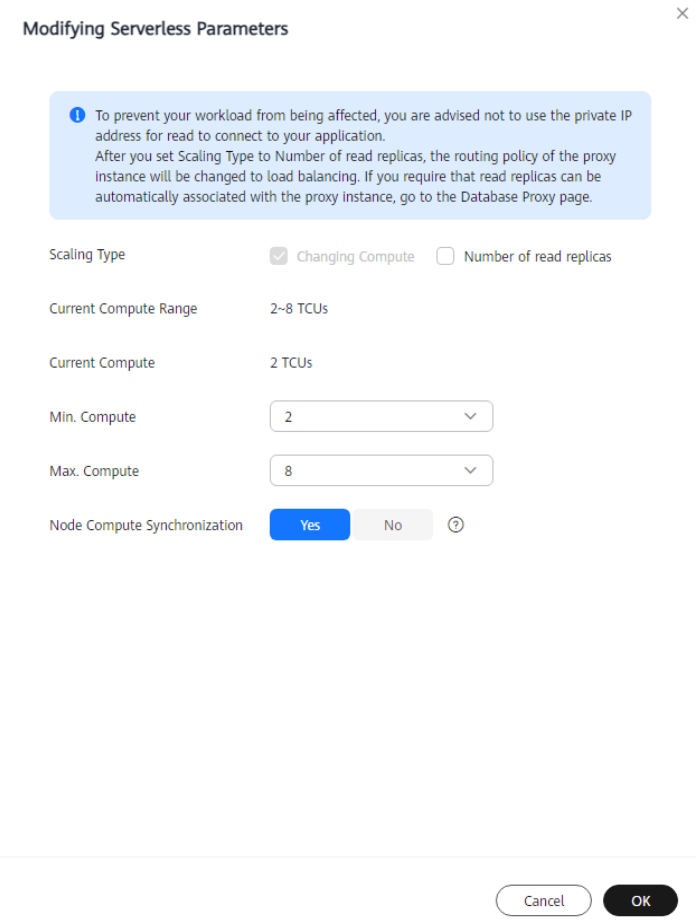
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Serverless Information** area, click **Change** under **Compute Range**.

Figure 11-3 Changing serverless information

Step 6 In the displayed dialog box, set **Min. Compute** and **Max. Compute**, and click **OK**.

Figure 11-4 Changing the compute range



Step 7 Check the new compute range in the **Serverless Information** area.

----End

11.3 Changing the Maximum and Minimum Numbers of Read Replicas

After **buying a serverless instance**, you can change the maximum and minimum numbers of read replicas. When certain trigger conditions are met, the number of read replicas of a serverless instance is automatically changed.

Conditions for Changing the Number of Read Replicas

If the compute has already been scaled up as much as possible but the CPU or memory usage still meets a compute scale-up condition, read replicas will be added.

If the compute has already been scaled down as much as possible but the CPU or memory usage still meets a compute scale-down condition, read replicas will be removed.

Constraints

- If database proxy is not enabled for an instance, the number of read replicas cannot be adjusted.
- To adjust the number of read replicas, there must be at least one proxy instance and new nodes can automatically be associated with the proxy instance. To associate new read replicas with a proxy instance, go to the **Database Proxy** page.
- To prevent your workloads from being affected, you are advised not to use the private IP address for read to connect to your application.
- After you set **Scaling Type** to **Number of read replicas**, the routing policy of the proxy instance will be changed to load balancing.
- Manually created read replicas are affected by the configured auto scaling policy. For example, if you set the minimum number of read replicas to 1 and manually create four read replicas, when the CPU or memory usage meets the scale-down conditions, the manually created read replicas will be removed until there is only one read replica.
- If serverless read replicas are added to an instance with fixed specifications, workloads are determined based on the serverless read replicas instead of the original TaurusDB read replicas.

Changing the Maximum and Minimum Numbers of Read Replicas

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

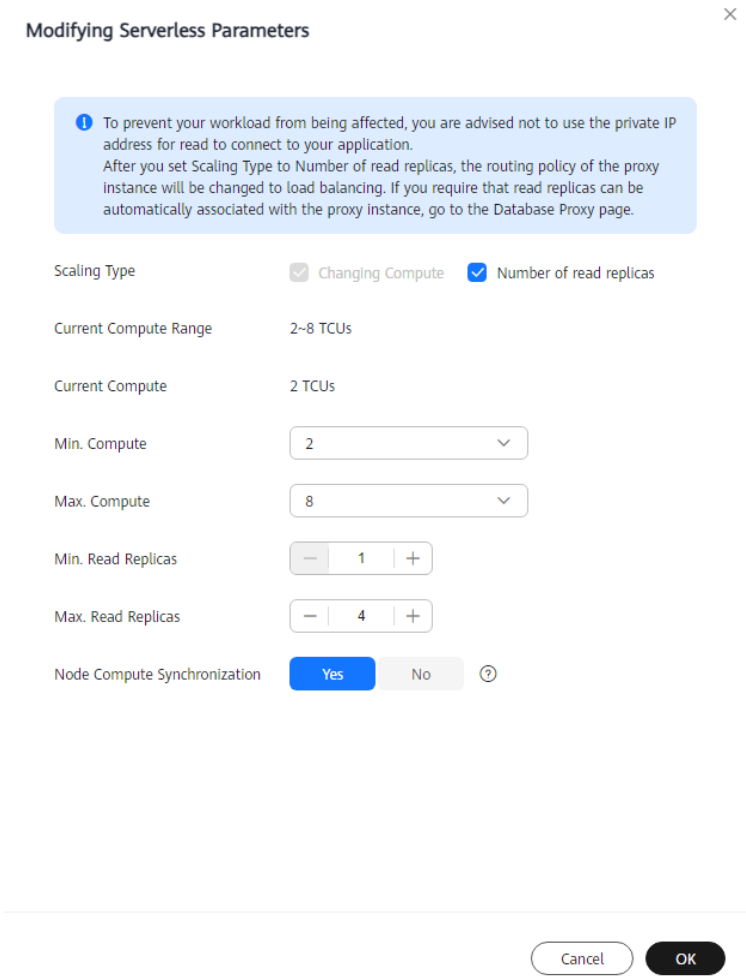
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Serverless Information** area, click **Change** under **Compute Range**.

Figure 11-5 Changing serverless information

Step 6 In the displayed dialog box, set **Scaling Type** to **Number of read replicas**, change the maximum and minimum numbers of read replicas, and click **OK**.

Figure 11-6 Changing the maximum and minimum numbers of read replicas



Step 7 Check the new maximum and minimum numbers of read replicas in the **Serverless Information** area.

----End

11.4 Adding Serverless Read Replicas to an Instance with Fixed Specifications

You can add serverless read replicas to a pay-per-use or yearly/monthly instance.

Constraints

- To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- The first time you create serverless read replicas, you need to initialize a scaling policy. If you need to modify the scaling policy later, click the instance name to enter the **Basic Information** page, click **Change** in the **Serverless Information** area, and modify serverless parameters.

- Adding serverless read replicas is mutually exclusive with the auto scaling function. If the serverless function has been enabled for an instance with fixed specifications, the auto scaling function cannot be enabled and vice versa.

Billing

After serverless read replicas are added, their compute is billed by hour. For details, see **Product Pricing Details**.

Adding Serverless Read Replicas

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **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, select the serverless billing mode and set other parameters.

Figure 11-7 Creating serverless read replicas

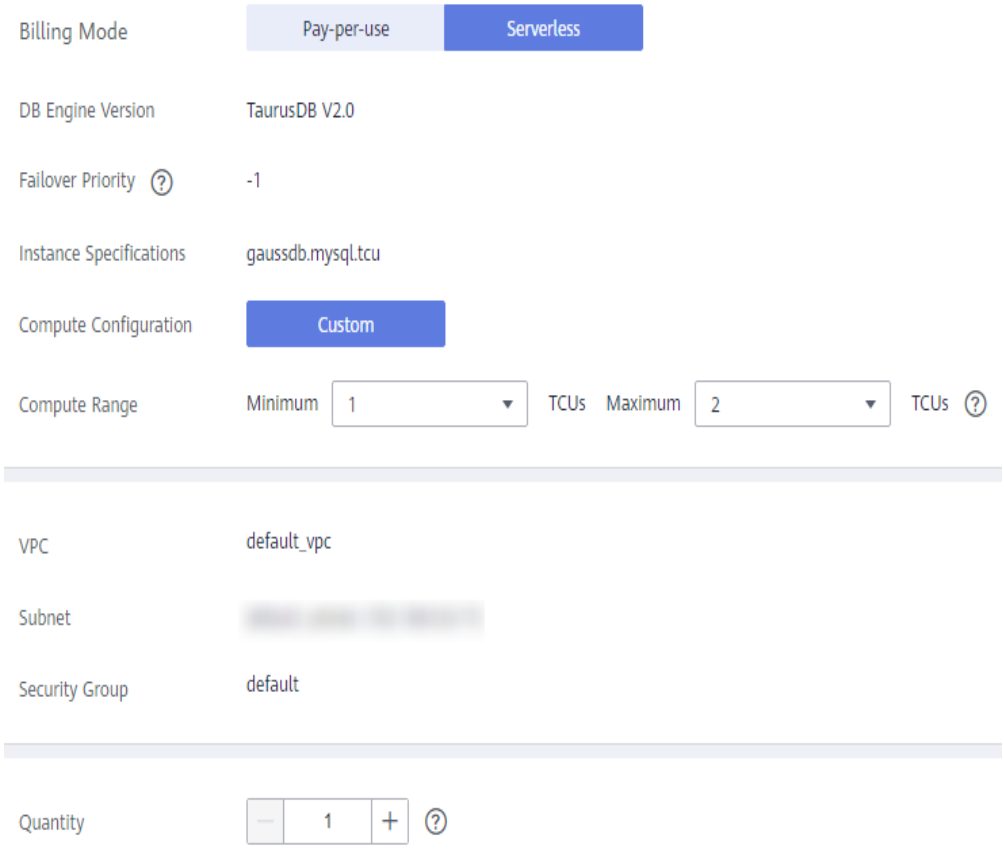


Table 11-2 Parameter description

Parameter	Description
Failover Priority	The failover priority of a serverless read replica is fixed to **-1**. During a failover, the serverless read replica will not be promoted to primary. The failover priority for serverless read replicas cannot be changed.
Compute Range	• **Minimum TCUs**: Set the minimum compute. 1 TCU is approximately equal to 1 CPU and 2 GB of memory. The initial specifications of a new serverless read replica are the minimum compute. • **Maximum TCUs**: Set the maximum compute.
Quantity	Up to seven serverless read replicas can be created for each instance.

Step 6 Click **Next**.

Step 7 Check the read replica settings.

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

Step 8 View or manage the created read replicas in the **Node List** area on the **Basic Information** page.

- If you want to modify the scaling policy, see **Changing the Maximum and Minimum Numbers of Read Replicas**.

  NOTE

  - To adjust the number of read replicas, there must be at least one proxy instance and new nodes can automatically be associated with the proxy instance, and there are serverless read replicas. If you require that read replicas can be automatically associated with the proxy instance, go to the **Database Proxy** page.
  - Once you enable the function to adjust the number of serverless read replicas for an instance with fixed specifications, the routing policy of proxy instances will remain unchanged. However, if the routing policy is weighted, an inappropriate weight configuration may render the function invalid. To prevent your workloads from being affected, you are advised to configure the same weight for serverless read replicas or use the load balancing routing policy when adjusting the number of serverless read replicas.
  - Each instance supports a maximum of 15 read replicas. The total number of serverless read replicas and existing non-serverless read replicas in an instance cannot exceed 15, and the number of serverless read replicas cannot exceed 7.

    Examples:

    If an instance has 13 non-serverless read replicas, the maximum number of serverless read replicas can be set to 2 instead of 7.

    If an instance has 5 non-serverless read replicas, the maximum number of serverless read replicas can be set to 7 instead of 10.

- If you want to delete a serverless read replica, see **Deleting a Read Replica**.

----End

12 Multi-primary Instances (OBT)

12.1 What Is a Multi-primary Instance?

A multi-primary instance can contain 2 to 63 read/write nodes, with no read replicas, enabling write many read many.

In a multi-primary instance, read/write nodes manage the metadata of the instance in a unified manner by sharing metadata. You can access the entire TaurusDB instance through a proxy address. The proxy instance automatically forwards your SQL commands to the correct read/write nodes.

Compared with a single-primary instance, a multi-primary instance allows for concurrent writes to different databases or tables on different nodes. Data can be concurrently written to up to 63 nodes, greatly improving the concurrent read/write capability.

12.2 Buying and Connecting to a Multi-primary Instance

Scenarios

This section describes how to buy a multi-primary instance on the TaurusDB console and connect to it through a proxy instance.

Billing

Multi-primary instances only support pay-per-use billing. After buying a multi-primary instance, you will be billed for resources you actually use. For billing details, see **Pay-per-Use Billing**.

Constraints

- To use multi-primary instances, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

- Multi-primary instances only support pay-per-use billing.
- The kernel version of a multi-primary instance must be:

  2.0.63.250300, 2.0.60.241201, 2.0.60.241200, 2.0.57.240922, 2.0.57.240920, 2.0.57.240905, or 2.0.57.240900

  For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**

Prerequisites

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

Step 1: Buy a Multi-primary Instance

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

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

- Basic configuration

  Figure 12-1 Basic configuration

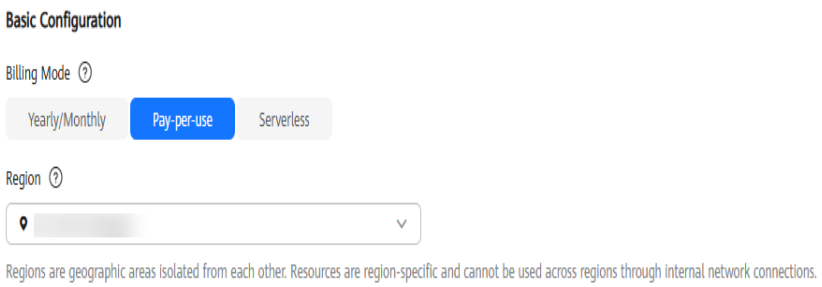


  Table 12-1 Basic configuration

Parameter	Description
Billing Mode	Select **Pay-per-use**.
Region	Region where an instance is deployed. **NOTE** You cannot change the region of an instance once it is purchased.

- Resource selection

Figure 12-2 Resource selection

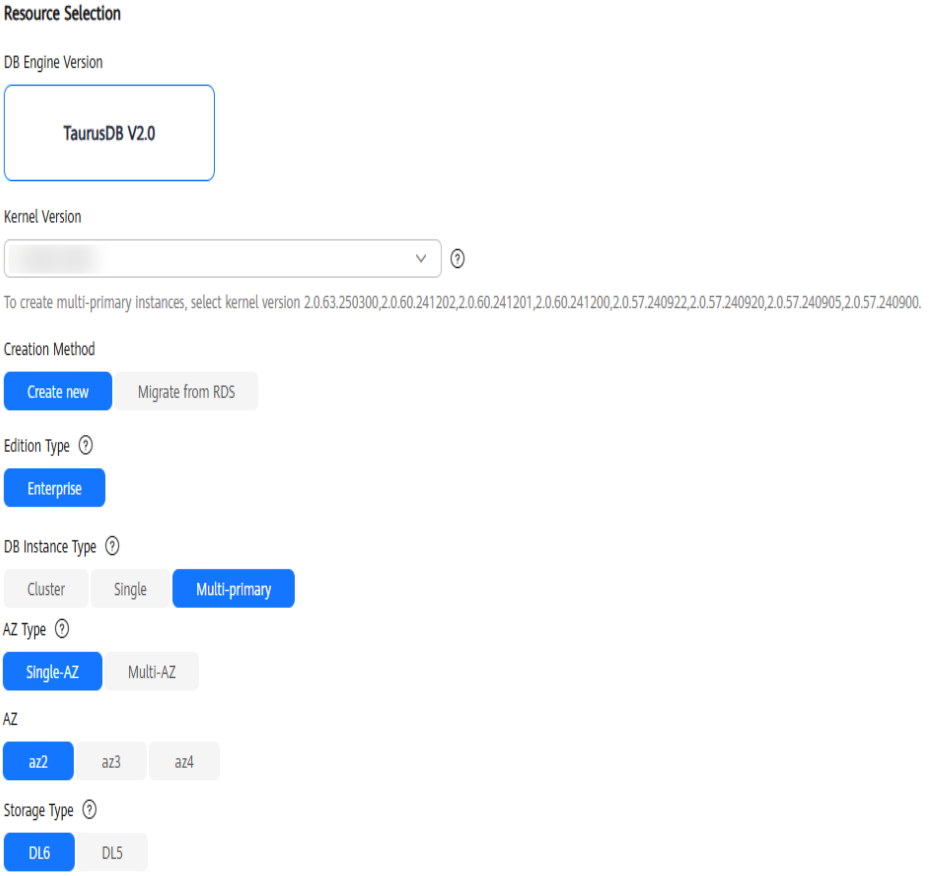


Table 12-2 Resource selection

Parameter	Description
DB Engine Version	Select **TaurusDB V2.0**.
Kernel Version	DB kernel version. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** To specify the kernel version when buying an instance, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
Creation Method	Select **Create new**.

Parameter	Description
Edition Type	**Enterprise**: Enterprise Edition is an enterprise-grade cloud-native database with high scalability and performance. It is fully compatible with open-source MySQL 8.0. It decouples compute from storage and uses Huawei-developed Data Function Virtualization (DFV), which scales to up to 128 TB per instance. A failover can be complete within seconds. High-value capabilities such as read/write splitting, operator pushdown, a serverless framework, and HTAP are also supported. It provides the high availability and superior performance of a commercial database at the price of an open-source database.
DB Instance Type	Select **Multi-primary**. A multi-primary instance can contain 2 to 63 primary nodes, with no read replicas. Such an instance can process multiple reads and writes, delivering excellent read/write performance at high concurrency.
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. – **Single-AZ**: The primary node and read replicas are deployed in the same AZ. – **Multi-AZ**: The primary node and read replicas are deployed in different AZs to achieve higher availability and reliability. It is suitable for workloads that require cross-AZ DR or are insensitive to cross-AZ latency.

Parameter	Description
Storage Type	– **DL6** The original shared storage. The default storage type of TaurusDB instances created before July 2024 is shared storage, while that of TaurusDB instances created in July 2024 and beyond is **DL6**. DL6-based instances achieve zero RPO with a 3-AZ deployment and deliver better performance and higher peak throughput. They are suitable for core application systems that are sensitive to performance and have demanding requirements on storage I/O during peak hours, such as those in finance, e-commerce, government, and gaming. – **DL5** A new type of storage. With Huawei Cloud's hardware and network infrastructure technologies, DL5-based instances maintain the same high availability (zero RPO in the 3-AZ deployment) as DL6-based instances. Although the peak performance of DL5-based instances may be a bit less than what you get with DL6-based instances, the cost per unit of capacity is a lot less. DL5-based instances are suitable for CPU-intensive sub-core business systems, or application modules that need to minimize costs. For more information about storage types, see **Storage Types**.

- Instance options

Figure 12-3 Specifications and storage

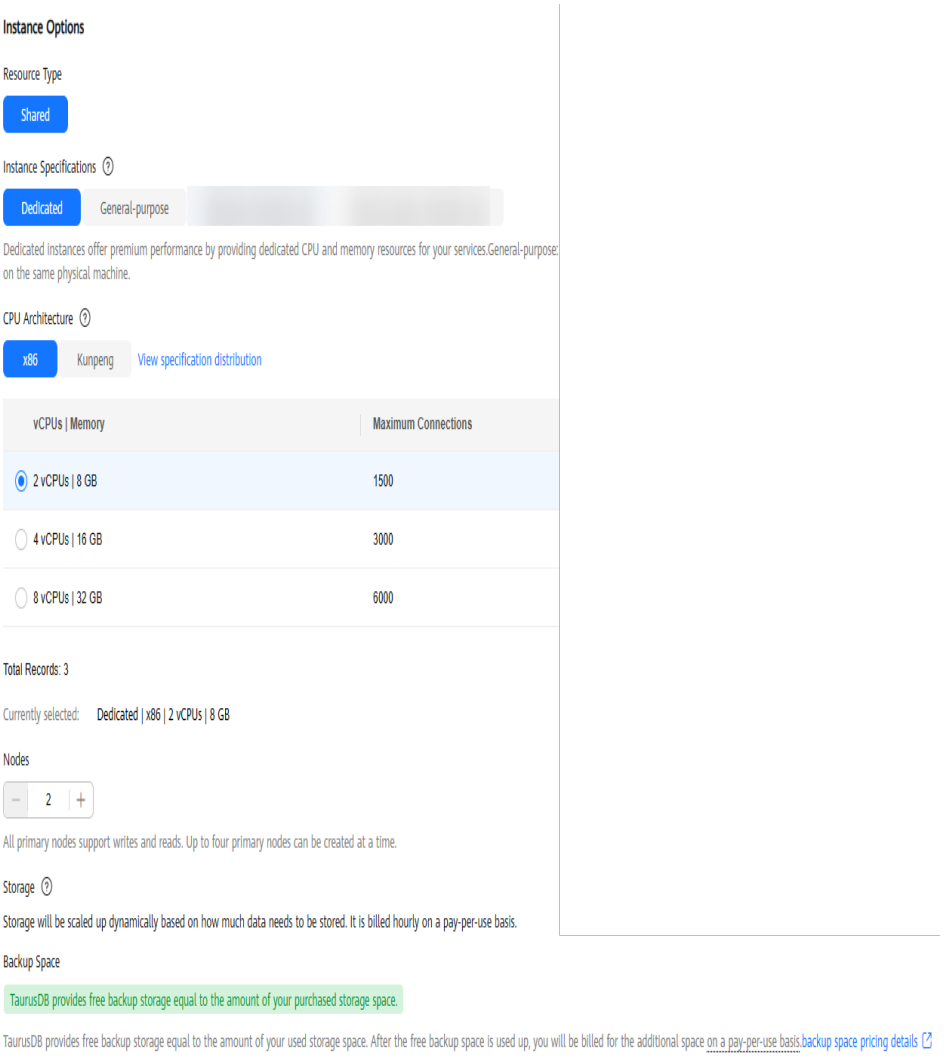


Table 12-3 Specifications and storage

Parameter	Description
Resource Type	Select **Shared**.
Instance Specifications	TaurusDB is a cloud-native database that uses the shared storage. To ensure workload stability in high read/write pressure, the system controls the read/write peaks of DB instances based on instance specifications. For details about how to select instance specifications, see **Performance White Paper**.

Parameter	Description
CPU Architecture	Select **x86** or **Kunpeng**. – **x86**: x86 instances use Intel® Xeon® Scalable processors and feature robust and stable computing performance. When working on high-performance networks, the instances provide the additional performance and stability that enterprise-class applications demand. – **Kunpeng**: Kunpeng instances use Kunpeng 920 processors and 25GE high-speed intelligent NICs for powerful compute and high-performance networks, making them an excellent choice for enterprises needing cost-effective, secure, and reliable cloud services.
Nodes	This parameter is mandatory for multi-primary instances. Each multi-primary instance requires at least two primary nodes. You can create up to 63 primary nodes at a time. All primary nodes are both readable and writable. You can also add read/write nodes after an instance is created. For details, see **Adding Read/Write Nodes to a Multi-primary Instance**.
Storage	It contains the system overhead required for inodes, reserved blocks, and database operations. Storage of a pay-per-use instance will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.
Backup Space	TaurusDB provides free backup space equal to the amount of your used storage. After the free backup space is used up, you will be billed for the additional space on a pay-per-use basis.

Figure 12-4 Network

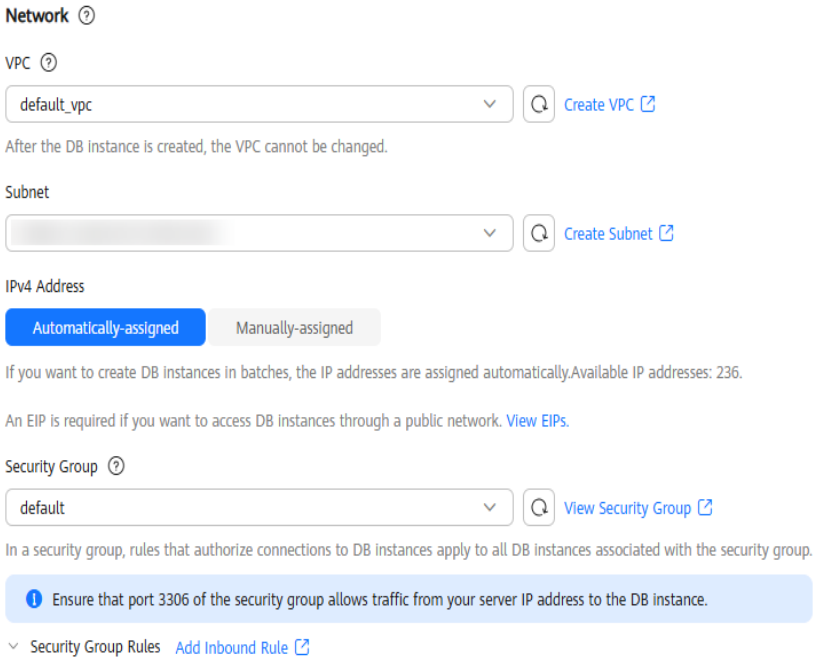


Table 12-4 Network

Parameter	Description
VPC	A dedicated virtual network where your instance is located. It isolates networks for different workloads to enhance security. TaurusDB allocates a default VPC (**default_vpc**) for your instance. You can also use an existing, new, or shared VPC. After an instance is created, the VPC cannot be changed. – To use an existing VPC, select an existing VPC under the current account from the drop-down list. – To use a new VPC, create a VPC, and then select the VPC from the drop-down list. For details about how to create a VPC, see **Creating a VPC and Subnet** in *Virtual Private Cloud User Guide*. – To use a shared VPC, select a VPC that another account shares with the current account from the drop-down list. With Resource Access Manager (RAM), you can share subnets in a VPC with one or more accounts, so you can easily configure and manage multiple accounts' resources at low costs. For more information about VPC subnet sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within an AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets where DB instances are located and cannot be disabled.
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 DB instances. If no security group is available or has been created, TaurusDB allocates a security group to your DB instance by default. To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. If the port and protocol are not enabled for the selected security group, click **Add Inbound Rule** as prompted and complete the configuration in the displayed dialog box. For details, see **Configuring Security Group Rules**.

Figure 12-5 Setting an administrator password

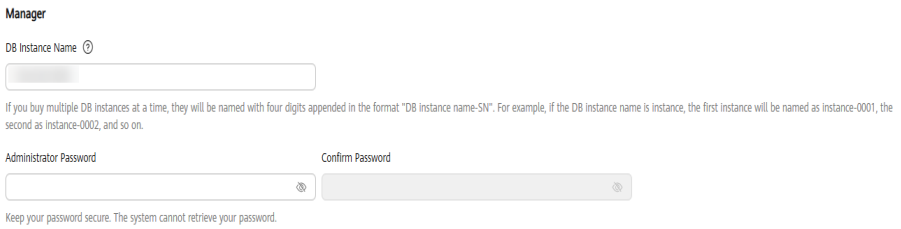


Table 12-5 Database configuration

Parameter	Description
DB Instance Name	The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. – If you create multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on. – The names for instances created in batches must consist of 4 to 59 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

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

- Advanced settings and required quantity

Figure 12-6 Advanced settings and required quantity

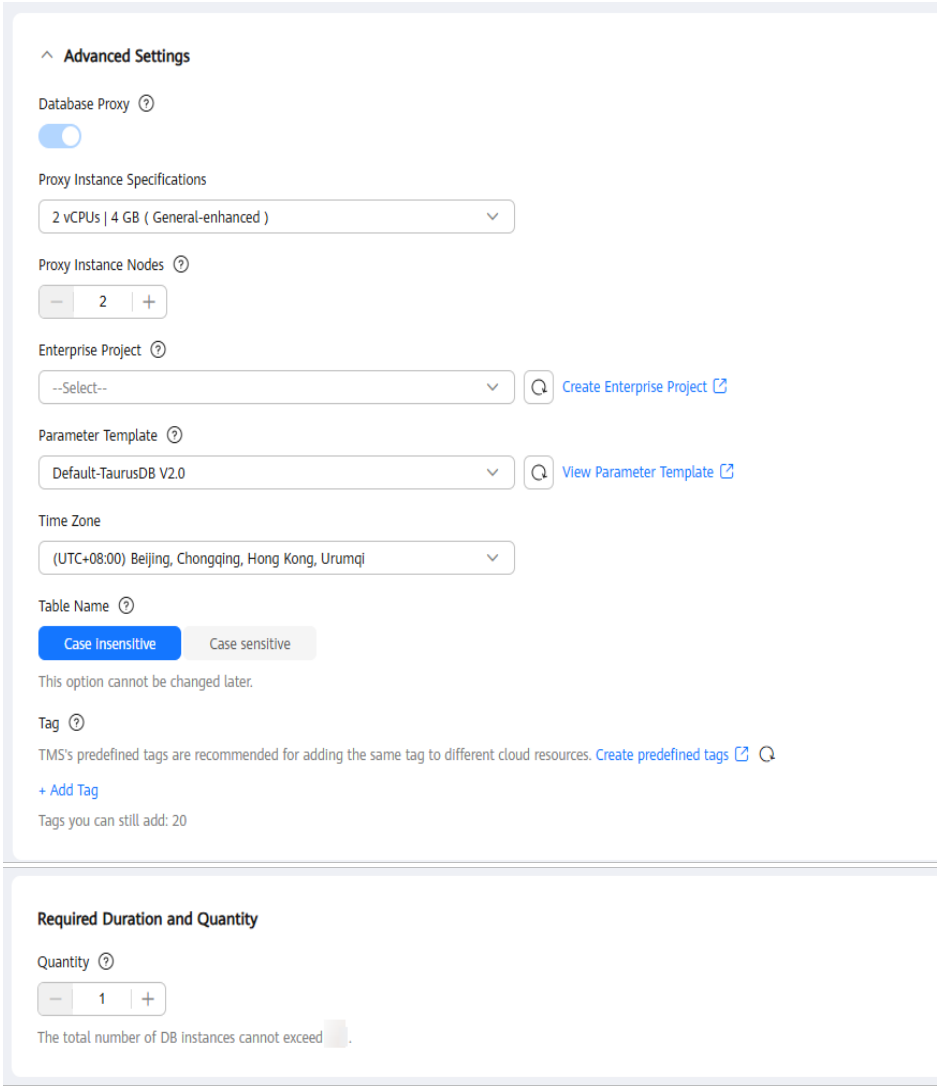


Table 12-6 Advanced settings

Parameter	Description
Database Proxy	You must enable **Database Proxy** for multi-primary instances. Then you can use proxy addresses to connect to your databases.
Proxy Instance Specifications	You can select the proxy instance specifications as needed.
Proxy Instance Nodes	You can create 2 to 16 proxy instance nodes.

Parameter	Description
Enterprise Project	Only available for enterprise users. If you want to use this function, contact customer service. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. In the drop-down list, you can select the default parameter template, the high-performance parameter template, or a custom parameter template in the current region as required. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** For more information about parameter templates, see **Parameter Management**. For more information about the high-performance parameter template, see **Introducing the High-Performance Parameter Template**.
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 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.
Tag	Tags a DB instance. This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.
Quantity	You can buy DB instances in batches. The default value is **1**. The value ranges from 1 to 10.

Step 3 Confirm the specifications of the pay-per-use instance.

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

Step 4 On the **instances** page, check and manage your multi-primary instance.

- 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 and cannot be changed after the instance is created.

----End

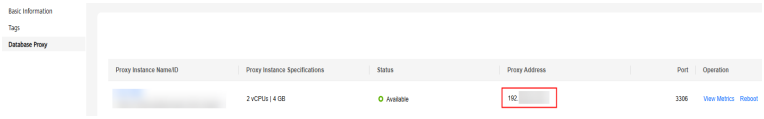
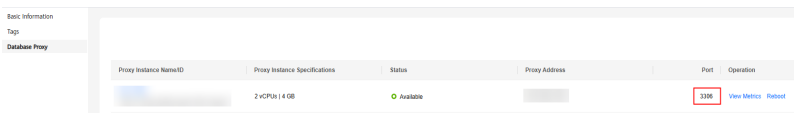
Step 2: Connect to the Multi-primary Instance Through a Proxy Instance

Step 1 Log in to an ECS.

Step 2 Connect to the instance through a proxy address.

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

Table 12-7 Parameter description

Parameter	Description
<hostIP>	Proxy address. Click the multi-primary instance name, choose **Database Proxy** in the navigation pane, and view the proxy address in the proxy instance list. **Figure 12-7** Viewing a proxy address 
<port>	Database port. By default, the value is **3306**. Click the multi-primary instance name, choose **Database Proxy** in the navigation pane, and view the database port in the proxy instance list. **Figure 12-8** Viewing a database port 

Parameter	Description
<userName>	The username of the database administrator account. The default username is **root**.

Enter the password of the database account as prompted.

```
Enter password:
```

----End

12.3 Adding Read/Write Nodes to a Multi-primary Instance

A multi-primary instance can contain 2 to 63 read/write nodes. With those nodes, it enables write many read many to deliver excellent read/write performance at high concurrency.

You can add read/write nodes after a multi-primary instance is created.

Billing

Added nodes are billed on a pay-per-use basis. Your billing will be adjusted according to the new number of nodes.

The following prices are for reference only. The actual prices are displayed on the console.

Suppose you purchased a TaurusDB multi-primary instance (instance specifications: dedicated, 2 vCPUs | 8 GB, 2 nodes; storage: DL6) in CN-Hong Kong on April 1, 2025. The instance price was $0.52 USD per hour.

You added two read/write nodes on April 15, 2025. Then you were billed for four nodes. The instance price was $1.04 USD per hour.

Procedure

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

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

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

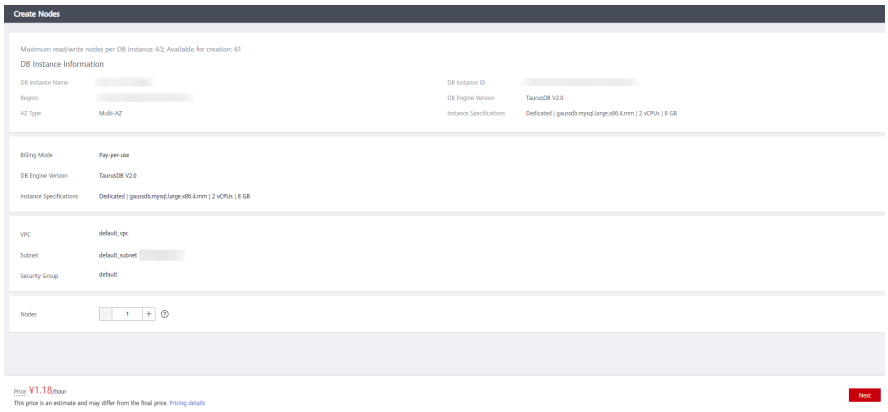
Step 4 On the **Instances** page, locate a multi-primary instance and click **Create Nodes** in the **Operation** column.

Alternatively, click the multi-primary instance name. On the **Basic Information** page, click in the upper right corner of the page and click **Create Nodes**.

Step 5 On the displayed page, set the number of read/write nodes.

Each multi-primary instance can contain up to 63 read/write nodes.

Figure 12-9 Creating nodes



Step 6 Click **Next**.

Step 7 Confirm the node settings.

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

Step 8 Check that the new read/write nodes are displayed in the **Node List** area of the **Basic Information** page. You can also change node names and reboot or delete those nodes.

----End

12.4 Deleting a Read/Write Node of a Multi-primary Instance

Scenarios

You can delete a read/write node of a multi-primary instance on the **Basic Information** page to release resources as required.

Constraints

- A deleted read/write node cannot be recovered. Exercise caution when performing this operation.
- You can only delete a read/write node when the DB instance has more than two read/write nodes. At least two read/write node must be retained in the instance.
- If another operation is being performed on a DB instance, the read/write nodes of the instance cannot be manually deleted.

Procedure

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

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

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

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

Step 5 In the **Node List** area in the lower part of the page, locate a read/write node and click **Delete** in the **Operation** column.

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

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

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

Step 8 In the **Node List** area of the **Basic Information** page, check that the node is deleted.

----End

13 Read Replicas

13.1 Introducing Read Replicas

What Are Read Replicas?

In read-intensive scenarios, a primary node may be unable to handle the read pressure and service performance may be affected. To offload read pressure from the primary node, you can create one or more read replicas. These read replicas can process a large number of read requests and increase application throughput. To do this, connection addresses need to be scheduled separately for the primary node and each read replica on your applications so that all read requests can be sent to read replicas and write requests to the primary node.

Billing Standards

Read replicas are billed as well. The billing mode is the same as that of the primary node.

Functions

- You do not need to maintain accounts and databases of read replicas. They are synchronized from the primary node.
- The system can monitor the performance of read replicas.

Constraints

- You can create a maximum of 15 read replicas for a yearly/monthly or pay-per-use instance, and seven read replicas for a serverless instance.
- Read replicas do not support restoration from backups.
- Data cannot be migrated to read replicas.
- You cannot create or delete databases on read replicas.
- You cannot create database accounts for read replicas.
- There may be a latency between the read replicas and the primary node. The latency of the full-text index is significant due to its special mechanism. For

latency-sensitive application workloads, you are advised to send queries to the primary node.

13.2 Adding Read Replicas to a DB Instance

Scenarios

Read replicas of a DB instance are used to enhance instance capabilities and reduce the read pressure on the primary node. After a DB instance is created, you can add read replicas.

There are synchronous and asynchronous read replicas.

- Synchronous read replicas: Their failover priority is 1 and specifications are the same as those of the primary node. To avoid failover failures caused by inconsistent specifications between the primary node and read replicas, a DB instance must have a synchronous read replica, and a multi-AZ DB instance must have a synchronous read replica in a different AZ from the primary node.
- Asynchronous read replicas: Their failover priority is not 1 and specifications are different from those of the primary node.

For more information about read replicas, see **Introducing Read Replicas**.

Deployment Relationships Between the Primary Node and Read Replicas

- If you select single-AZ deployment, read replicas are deployed in the same AZ as the primary node.
- If you select multi-AZ deployment, read replicas are evenly deployed in different AZs to ensure high reliability.

Constraints

- Each yearly/monthly or pay-per-use DB instance has a maximum of 15 read replicas.
- Each serverless DB instance has a maximum of 7 read replicas.
- If all synchronous read replicas are unavailable during a failover, an asynchronous read replica will be promoted to primary.

Billing

Table 13-1 Billing for new nodes

Billing Mode of New Nodes	Impact on Price
Yearly/ Monthly	You will be billed for the new nodes based on the time remaining in the requested period of your instance. You need to pay the price difference. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month TaurusDB cluster instance (instance specifications: dedicated, 2 vCPUs | 8 GB, 2 nodes; storage: DL6, 10 GB) in CN-Hong Kong on April 1, 2025. The instance price was $296 USD per month. On April 15, 2025, you added a read replica (specifications: dedicated, 2 vCPUs | 8 GB). The instance price was $436 USD per month. **Price difference = Price for the new instance configuration x Remaining period – Price for the original instance configuration x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Price difference = $436 USD x 0.5 – $296 USD x 0.5 = $70 USD
Pay-per-use	New nodes are billed by hour. For details, see **Product Pricing Details**.

Procedure

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

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

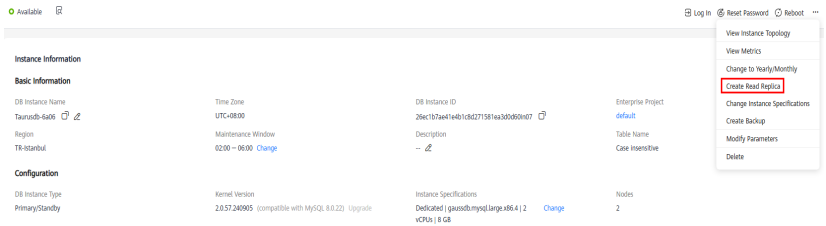
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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.

You can also enter the **Create Read Replica** page in either of the following ways:

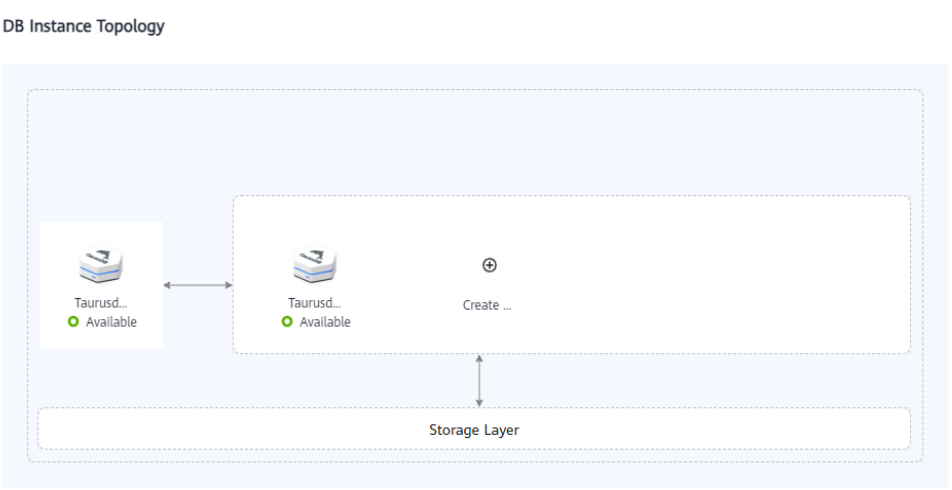
- Click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click and choose **Create Read Replica**.

Figure 13-1 Creating read replicas on the **Basic Information** page



- Click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click ••• and choose **View Instance Topology**. In the instance topology, click **Create Read Replica**.

Figure 13-2 Creating read replicas in the instance topology



Step 5 On the displayed page, set required parameters.

Table 13-2 Parameter description

Parameter	Description
Billing Mode	• Pay-per-use DB instance: Pay-per-use and serverless read replicas can be added. • Yearly/Monthly DB instance: Yearly/Monthly, pay-per-use, and serverless read replicas can be added. • Serverless DB instance: Only serverless read replicas can be added.

Parameter	Description
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. • Serverless DB instance: The failover priority for the primary node can only be **1**, while that for a newly added read replica can be 1 to 15. • Yearly/Monthly DB instance: When a pay-per-use or serverless read replica is added, the failover priority is **-1** by default and cannot be changed. • Pay-per-use DB instance: When a serverless read replica is added, the failover priority is **-1** by default and cannot be changed.
AZ	TaurusDB multi-AZ instances allow you to select an AZ when creating a read replica. Serverless DB instances do not allow you to specify AZs for read replicas. • If no AZs are specified, the created read replicas are evenly distributed in each AZ. • If too many nodes are created in a specified AZ, the read replicas may fail to be created due to insufficient resources. **NOTE** To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
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	A DB instance can contain up to 15 read replicas.

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

Step 7 For a pay-per-use instance, click **Next**.

Step 8 Check the read replica settings.

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

Step 9 View the new read replicas in the **Node List** area of the **Basic Information** page. You can also promote a read replica to primary or delete a read replica.

----End

APIs

- **Creating a Read Replica**
- **Deleting or Unsubscribing from a Read Replica**

13.3 Promoting a Read Replica to Primary

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.

Constraints

- A read replica whose failover priority is -**1** cannot be promoted to the primary node.
- Services may be intermittently interrupted for several seconds or minutes when a read replica is promoted to the primary node.
- Promoting a read replica to primary will switch over the private IP addresses for read of the primary node and read replica. To ensure workloads are not interrupted, connect to your DB instance using the private IP address from the **Network Information** area in the **Basic Information** page or the proxy address from the **Database Proxy** page. For details about the differences between the two addresses, see **Description of Each IP Address**.
- To ensure workload continuity, you first **enable Application Lossless and Transparent (ALT)** and then promote a read replica to primary.

Manual Switchover

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

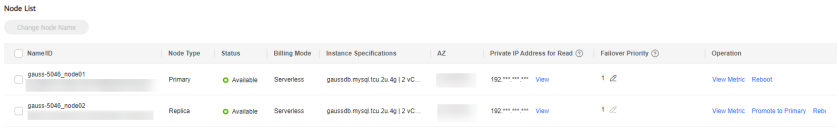
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **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.

Figure 13-3 Promoting a read replica to the new primary node



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

- During a manual switchover, there may be a brief disconnection lasting about 30 seconds. Ensure that your applications support automatic reconnection.
- During a manual switchover, the DB instance status is **Promoting to primary** and this process takes several seconds or minutes.

Step 7 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**.

----End

Automatic Failover

TaurusDB uses an active-active HA architecture that automatically fails over to a read replica selected by the system.

Each read replica has a failover priority that determines which read replica is promoted if the primary node fails.

- Priorities range from 1 for the highest priority to 16 for the lowest priority.
- If two or more read replicas share the same priority, they have a same probability of being promoted to the new primary node.

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.

13.4 Deleting a Read Replica

Scenarios

You can delete read replicas billed on a pay-per-use or serverless basis on the **Basic Information** page.

Constraints

- Deleted read replicas cannot be recovered. Exercise caution when performing this operation.
- You can only delete a read replica when the DB instance has two or more read replicas.
- If another operation is being performed on a DB instance, the read replicas of the instance cannot be manually deleted.
- For multi-AZ deployment, make sure that the primary node and remaining read replicas are located in different AZs after a read replica is deleted.

If a primary node and a read replica are deployed in AZ1 and the other read replica is deployed in AZ2, the read replica in AZ2 cannot be deleted.

- Before deleting the last serverless read replica, ensure that the function for adjusting the number of serverless read replicas has been disabled.

Billing

Deleted pay-per-use or serverless read replicas are no longer billed.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **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 deleted and choose **More** > **Delete** in the **Operation** column.

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

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

Step 7 In the displayed dialog box, click **Yes**. Refresh the **Instances** page later to confirm that the deletion has completed.

----End

APIs

- **Creating a Read Replica**
- **Deleting or Unsubscribing from a Read Replica**

13.5 Unsubscribing from a Read Replica

Scenarios

You can unsubscribe a read replica of a yearly/monthly instance.

Constraints

- You can only unsubscribe a read replica when the DB instance has two or more read replicas.
- Only isolated read replicas can be unsubscribed.
- If a read replica of a DB instance is being isolated, you cannot perform the following operations for the instance:

  - Creating read replicas
  - Scaling up storage space
  - Changing instance specifications
  - Rebooting the instance
  - Resetting the password
  - Upgrading the patch
  - Changing the private IP address
  - Changing the database port
  - Enabling or disabling SSL
  - Binding an EIP
  - Operations related to proxy instances
- The following operations cannot be performed on other read replicas of the instance:
  - Changing a failover priority
  - Promoting a read replica to primary
  - Isolating a read replica

Billing

For unsubscription fees, see **Unsubscription Rules**.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **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 a read replica and click **Isolate** in the **Operation** column.

- When a read replica is isolated, you can only unsubscribe or release it.
- When the workloads are heavy, you can release the isolated read replica if necessary.

Step 6 After the read replica status changes to **Isolated**, choose **More** > **Unsubscribe** in the **Operation** column.

- It takes about 1 minute to isolate a read replica.
- When a read replica is isolated, read operations and database synchronization cannot be performed.

Step 7 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For unsubscription details, see **Unsubscription Rules**.

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

> **CAUTION**
>
> After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.

Step 9 View the unsubscription result. After the order is successfully unsubscribed, the unsubscribed read replica of the instance will be deleted.

----End

APIs

- **Creating a Read Replica**
- **Deleting or Unsubscribing from a Read Replica**

13.6 Changing the Private IP Address for Read of a Read Replica

Scenarios

If the IP address of a read replica conflicts with other IP addresses, you can change it on the DB instance's **Basic Information** page to reduce application changes.

Constraints

If a DB instance has a read-only proxy instance, and the proxy instance contains only the read replica whose private IP address for read is to be changed and read weight is greater than 0, the private IP address for read of the read replica cannot be changed.

Changing a Private IP Address for Read

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**.

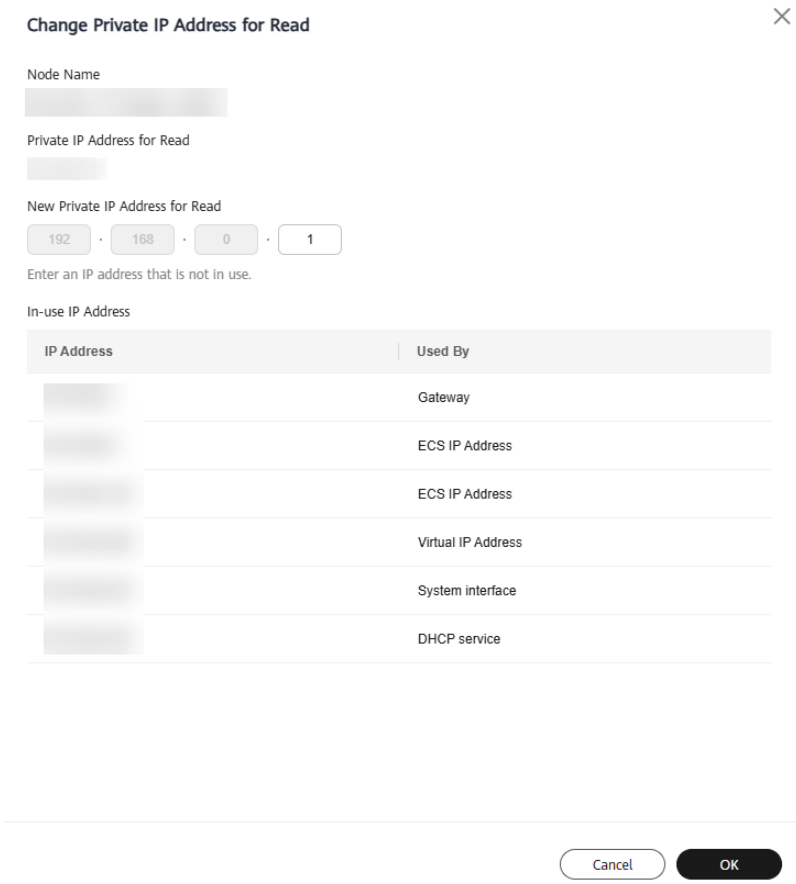
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, locate a read replica and click **Change** in the **Private IP Address for Read** column.

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

An in-use IP address cannot be used as the new private IP address for read.

Figure 13-4 Changing a private IP address for read



----End

14 Database Proxy (Read/Write Splitting)

14.1 What Is Database Proxy?

Database Proxy is a network proxy service that sits between TaurusDB and applications. It is used to handle all requests from the applications to access TaurusDB instances.

Read/write splitting means that read and write requests are automatically forwarded through database proxy addresses. After creating a TaurusDB instance, you can **create a proxy instance**. With the proxy address, write requests are automatically forwarded to the primary node and read requests are forwarded to each node based on the routing policy of the proxy instance, offloading the read load from the primary node.

Proxy instances are free.

Proxy Address

After buying a proxy instance, you can view the proxy address on the **Database Proxy** page. The proxy instance sends write requests to the primary node and read requests to read replicas through this address.

Proxy Mode

There are read/write and read-only (TP) proxy modes. The read/write attribute processing logic varies depending on the proxy mode. For details, see **Table 14-1**.

- **Read/Write**: All write requests are routed only to the primary node, and all read requests are routed to the selected nodes based on the read weights or active connections.
- **Read-only (TP)**: 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.

Table 14-1 Read/Write attribute processing logic

Proxy Mode	Routing Policy	Weight of Primary Node	Normal Case	All Read Replicas Are Faulty
Read-only	Weighted Load balancing	Not configurable	The primary node does not process read-only requests. Proxy address: readable but not writable	The primary node does not process read-only requests. Proxy address: connection error
Read/Write	Load balancing	Assigned by system	Primary node: readable and writable Proxy address: readable and writable	Primary node: readable and writable Proxy address: readable and writable
	Weighted	> 0	Primary node: readable and writable Proxy address: readable and writable	Primary node: readable and writable Proxy address: readable and writable
		= 0	Primary node: not readable but writable Proxy address: readable and writable	Primary node: readable and writable Proxy address: readable and writable

Transaction Splitting

With transaction splitting enabled for a proxy instance, the proxy instance can route read requests prior to write operations in a transaction to read replicas, reducing the load on the primary node.

For more information about transaction splitting, see **Enabling Transaction Splitting for a Proxy Instance**.

Connection Pool

Proxy instances provide session-level connection pools, which help reduce the database load caused by frequent establishment of short connections.

For more information about connection pools, see **Enabling the Connection Pool for a Proxy Instance**.

Routing Policy

Proxy instances support weighted and load balancing routing policies.

- **Weighted**: Read requests are assigned to nodes based on the weights you specify.
- **Load balancing**: Read requests are assigned to nodes with fewer active connections. In the load balancing policy, you do not need to configure the weights of nodes.

For more information about routing policies, see **Changing the Routing Policy of a Proxy Instance**.

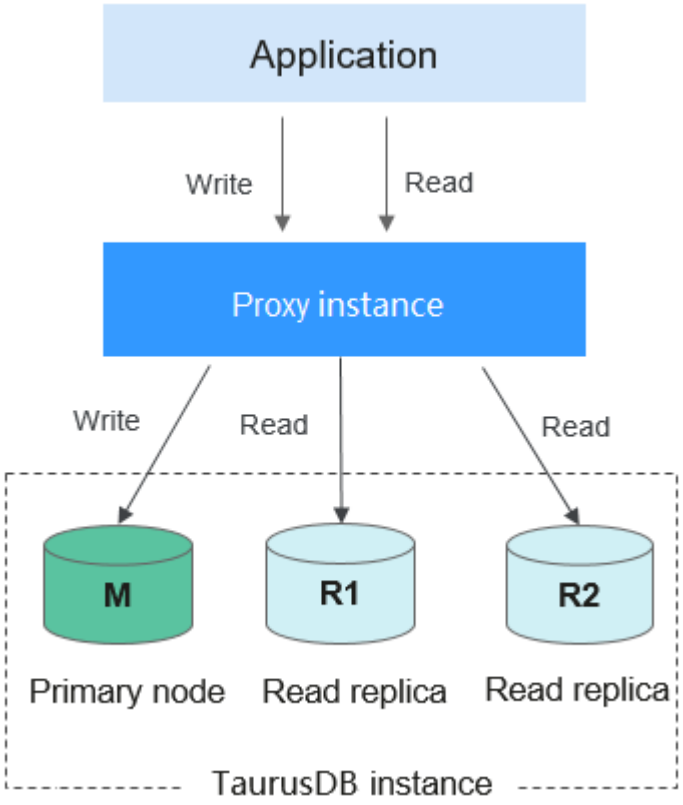
How Read/Write Splitting Works

You can create one or more proxy instances for your TaurusDB instance to enable read/write splitting.

Single Proxy Instance

If your TaurusDB instance has only one proxy instance, applications connect to the proxy instance through the proxy address. Write requests are forwarded to the primary node and read requests to the primary node or read replicas based on the routing policy you specify.

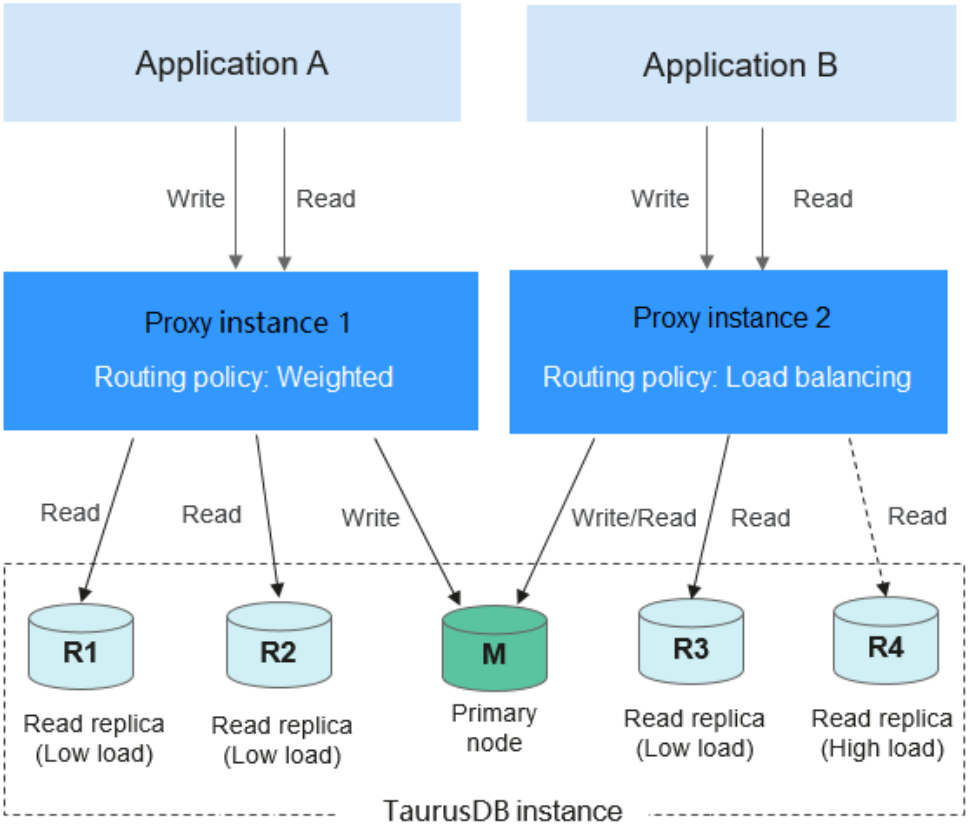
Figure 14-1 Read/write splitting with only one proxy instance



Multiple Proxy Instances

To isolate workloads from one another, you can create up to four proxy instances for a TaurusDB instance. Different applications can connect to different proxy instances as required. The associated read replicas of the proxy instances process read requests from different applications for workload isolation.

Figure 14-2 Read/write splitting with multiple proxy instances



Read/Write Splitting Advantages

- Compared with manual read/write splitting in applications, the read/write splitting using proxy addresses features flexible scale-out and low maintenance costs.
- Read requests are distributed to your read replicas based on weights to balance your database traffic and improve resource utilization.
- A proxy instance routes read requests of an application only to its associated read replicas to isolate workloads.
- By default, proxy instances provide overload protection to prevent server OOM (out of memory) due to heavy pressure when you perform operations on large result sets. This function is enabled by default and does not need to be configured separately. The pressure caused by the slow kernel depends on flow control.

Write Requests Sent Only to the Primary Node

- INSERT, UPDATE, and DELETE

- All DDL operations (such as table/database creation, table/database deletion, table structure change, and permission change)
- All requests in transactions (But if transaction splitting is enabled, some read requests in transactions may be sent to read replicas. For details, see **Enabling Transaction Splitting for a Proxy Instance**.)
- User-defined functions
- Stored procedures
- EXECUTE statements
- Multi-statement requests
- Requests that use temporary tables
- All changes to user variables
- KILL in SQL statements (not command KILL)

Read Requests Sent Only to the Primary Node

- If query statements are in transactions, the transaction requests are routed to the primary node. If **SET AUTOCOMMIT=0** is added before a query statement, the transaction requests are routed to the primary node.
- If all read replicas are abnormal or the read weights allocated to the read replicas are 0, requests will be routed to the primary node. You can set read weights for the primary node and read replicas after read/write splitting is enabled.
- When running SQL statements:
  - If multi-statements (for example, **insert xxx;select xxx**) are executed, all subsequent requests will be routed to the primary node. To restore read/write splitting, disconnect your application from your instance and then connect it back again.
  - Read operations with locks (for example, **SELECT for UPDATE**) will be routed to the primary node.
  - When **/*FORCE_MASTER*/** is used, requests will be routed to the primary node.
  - If **the HANDLER statement** is executed, all subsequent requests will be routed to the primary node. To restore read/write splitting, disconnect your application from your instance and then connect it back again.
- SELECT last_insert_id()
- All queries of user variables

Requests Sent Either to the Primary Node or a Read Replica

- SELECT not in a transaction
- The COM_STMT_EXECUTE command

Requests Always Sent to All Nodes

- Changes to all system variables
- The USE command

Precautions

Table 14-2 Precautions for proxy instances

Category	Precaution
Version constraints	• If the kernel version of your TaurusDB instance is one of the following, proxy instances cannot be created: – From 2.0.26.2 to 2.0.28.3 – 2.0.29.1 • If the kernel version of your TaurusDB instance is earlier than 2.0.42.230601, only one proxy instance can be created. • If the kernel version of your TaurusDB instance is 2.0.42.230601 or later, up to four proxy instances can be created. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
Unsupported functions	• Proxy instances do not support compression protocols. • Proxy instances do not support the READ-UNCOMMITTED transaction isolation level. • Proxy instances do not support reads from and writes to any column containing more than 16 MB of data in a table. • Database proxies do not support the SQL mode parameter **PAD_CHAR_TO_FULL_LENGTH**.

Category	Precaution
Usage constraints	• To create a proxy instance, the TaurusDB instance must have at least 8 vCPUs. • Read/write splitting can be enabled only when at least one read replica is created. • After read/write splitting is enabled, the database port and private IP address of your TaurusDB instance cannot be changed. • If **multi-statements** are executed, all subsequent requests will be routed to the primary node. To restore the read/write splitting function, disconnect the connection from your applications and establish a connection again. • When a proxy address is used, all transaction requests are routed to the primary node (you can use transaction splitting to route read requests prior to write operations in a transaction to read replicas). The non-transaction read consistency is not ensured. To ensure read consistency, encapsulate the read requests into a transaction. • When a proxy address is used, you can run **show processlist** command on the proxy instance or TaurusDB instance. If **show processlist** is executed on a proxy instance, only the services delivered through proxy nodes are displayed. • If a proxy node is abnormal, running **show processlist** or **Kill** on the proxy instance may take a long time, but services are not affected. • After a proxy node is deleted, services on the deleted proxy node may be displayed when **show processlist** is executed on the proxy instance. • If **Kill** is executed on the proxy instance, error information such as timeout may be displayed occasionally. You can run **show processlist** again to check whether the services are killed successfully. • If a proxy node is abnormal, there may be frame freezing for 2 seconds when you run **show processlist** on the proxy instance. The results will still be returned. • When a proxy instance is used, the size of a concatenate SQL statement cannot exceed 100 MB to prevent statement parsing from consuming too many resources.
HTAP analysis	• Consistency levels and connection pools are not supported. • Only the weighted routing policy is supported. • Only the read/write proxy mode is supported.

14.2 Creating a Proxy Instance for Read/Write Splitting

After creating a TaurusDB instance, you can create a proxy instance. With the proxy address, write requests are automatically forwarded to the primary node, and read requests are forwarded to each node based on the routing policy of the proxy instance, offloading the read pressure from the primary node.

This section describes how to create a proxy instance for read/write splitting.

Step 1: Create a Proxy Instance

Step 2: Perform User Authentication

Step 3: Check Security Group Rules

Step 4: Use the Proxy Address to Connect to Your TaurusDB Instance

Step 5: Verify Read/Write Splitting

Constraints

Before creating a proxy instance, you need to ensure that:

- You have purchased a TaurusDB instance.
- You have understood the precautions. For details, see **Precautions**.

Step 1: Create a Proxy Instance

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **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 **Database Proxy**.

Step 6 Click **Create Proxy Instance**.

Step 7 On the displayed page, set required parameters and click **Next**.

Figure 14-3 Creating a proxy instance

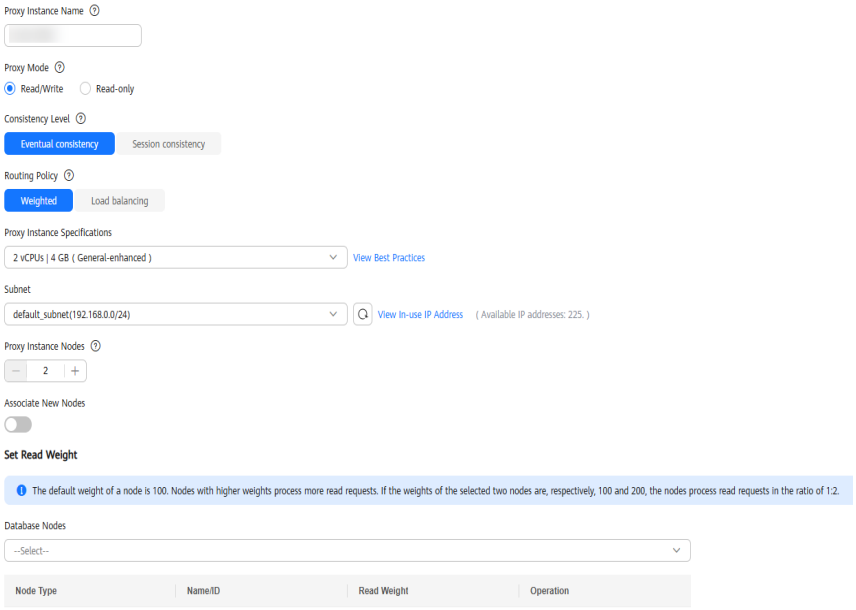


Table 14-3 Parameter description

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	You can select **Read/Write** or **Read-only** as needed. • **Read/Write**: All write requests are forwarded only to the primary node, and all read requests are forwarded to the selected nodes based on the read weights or active connections. The default read weight of a node is **100**. • **Read-only (TP)**: Write requests are not forwarded to any node. All read requests are forwarded to the selected read replicas based on the read weights or active connections. The read requests are not forwarded to the primary node, even if the primary node is selected. **NOTE** – In the read-only (TP) mode, only read requests are supported. If write requests are forwarded to any node, an error message is displayed. – DDL, DML, and temporary table operations are not supported in the read-only (TP) mode.

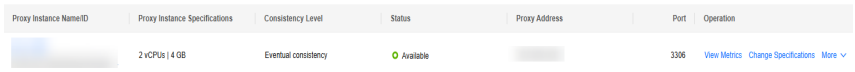
Parameter	Description
Consistency Level	The consistency level can only be configured when the kernel version of your TaurusDB instance is 2.0.28.1 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?** Value: • **Eventual consistency** After a proxy instance is created, requests for SELECT operations are routed to different nodes based on their read weights. Because there is a replication latency between the primary node and each read replica and the replication latency varies for different read replicas, the result returned by each SELECT statement may be different when you repeatedly execute a SELECT statement within a session. In this case, only eventual consistency is ensured. To offload read requests from the primary node to read replicas, you can select eventual consistency. • **Session consistency** To eliminate data inconsistencies caused by eventual consistency, session consistency is provided. Session consistency ensures the result returned by each SELECT statement in a session is the data that was updated after the last write request. To use session consistency, the kernel version of your proxy instance must be 2.7.4.0 or later.
Routing Policy	Value: • **Weighted**: Read requests are assigned to nodes based on the weights you specify. • **Load balancing**: Read requests are assigned to nodes with fewer active connections. To use load balancing, the kernel version of your proxy instance must be 2.22.07.000 or later. For more information about routing policies, see **Changing the Routing Policy of a Proxy Instance**.
Proxy Instance Specifications	You can select the proxy instance specifications as needed. • Kunpeng general computing-plus: 2 vCPUs | 4 GB, 4 vCPUs | 8 GB, and 8 vCPUs | 16 GB • General-enhanced: 2 vCPUs | 4 GB, 4 vCPUs | 8 GB, and 8 vCPUs | 16 GB

Parameter	Description
Subnet	To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. When creating a proxy instance, you can specify a subnet for the proxy instance. If the subnet where the TaurusDB instance is associated with is a secondary CIDR block, you can only select the same subnet as the TaurusDB instance for the proxy instance.
Proxy Instance Nodes	You can enter an integer from 2 to 16. The default value is **2**. Number of recommended proxy instance nodes = (Number of vCPUs of the primary node + Total number of vCPUs of all read replicas)/(4 x Number of vCPUs of the proxy instance), rounded up.
Associate New Nodes	After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the proxy instance.
New Node Weight	If **Routing Policy** is **Weighted**, you need to set read weights of the new nodes. The default weight of a node is **100**. Nodes with higher weights process more read requests.
Database Nodes	You need to select the nodes to be associated with the proxy instance for processing read requests. • If **Routing Policy** is **Load balancing**, you do not need to configure read weights for selected nodes. Read requests are forwarded to nodes with fewer active connections. • If **Routing Policy** is **Weighted**, you need to configure read weights of the primary node and read replicas. Read requests are forwarded to nodes based on the weights you specify. For example, read weights assigned to one primary node and two read replicas are 100, 200, and 200, respectively. In the read/write mode, the primary node and two read replicas process read requests in the ratio of 1:2:2. The primary node processes 20% of read requests, and each read replica processes 40% of read requests. Write requests are automatically routed to the primary node. In the read-only mode, the read weight of the primary node does not take effect, and the two read replicas process 50% of read requests, respectively.

Step 8 View the proxy instance and associated nodes.

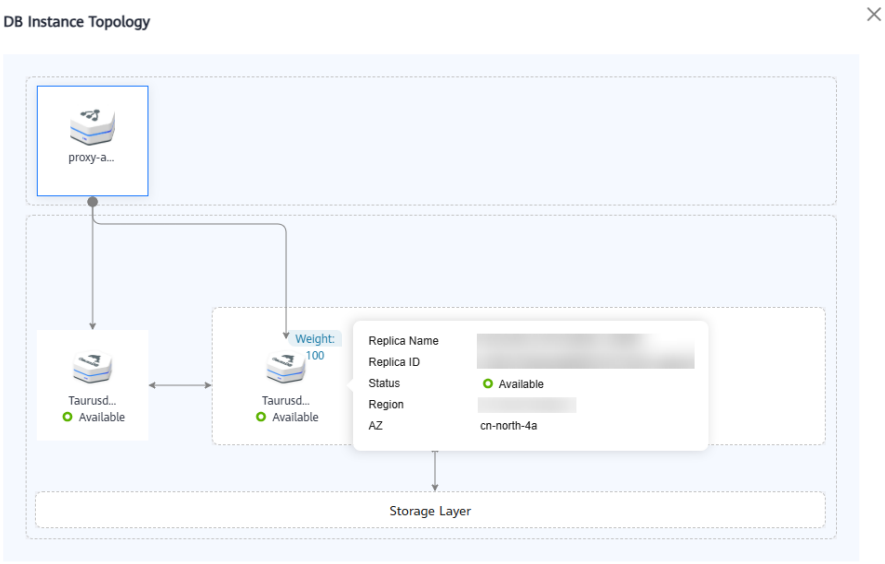
After the proxy instance creation is complete, you can view the created proxy instance on the **Database Proxy** page.

Figure 14-4 Viewing the create proxy instance



Click **Basic Information** in the navigation pane. Click ··· in the upper right corner of the page and select **View Instance Topology**. In the instance topology, you can view the database nodes associated with the proxy instance. You can move the pointer to a node name to view its details.

Figure 14-5 Viewing information about nodes associated with a proxy instance



----End

Step 2: Perform User Authentication

Before using a proxy instance to connect to your TaurusDB instance, ensure that the current database account has the permissions to access the proxy address, or the proxy instance cannot connect to your TaurusDB instance.

You can perform the following steps to check the permissions and grant the account the permissions to access the proxy address.

Step 1 Connect to the TaurusDB instance.

For details, see **Connecting to a DB Instance**.

Step 2 After the instance is connected, run the following SQL statement to check whether the host of the current database account contains a proxy address:

```
SELECT user,host FROM mysql.user;
```

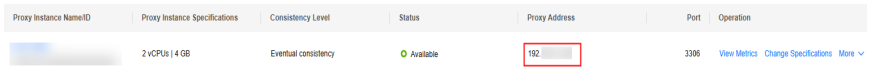


```
mysql> select user,host from mysql.user;
+---------------+-----------+
| user          | host      |
+---------------+-----------+
| app           | %         |
| rdsProxy      | %         |
| repl          | %         |
| root          | %         |
| test          | %         |
| testGTPUser   | %         |
| mysql.session | localhost |
| mysql.sys     | localhost |
| root          | localhost |
+---------------+-----------+
```

To obtain the proxy address:

Click the TaurusDB instance name. In the navigation pane, choose **Database Proxy**. In the proxy instance list, view the proxy address.

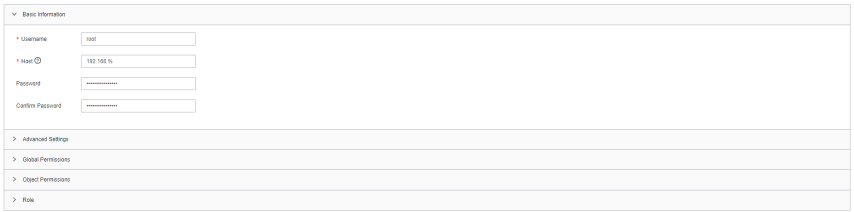
Figure 14-6 Viewing a proxy address



Step 3 If the host does not contain the CIDR block where the proxy instance is associated with, assign the remote access permissions to the host.

For example, if you want to connect to the TaurusDB instance using 192.168.0 as user **root**, set **Host** to **192.168.%** on the DAS user management page. For details, see **Editing a User**.

Figure 14-7 Configuring a host IP address



----End

Step 3: Check Security Group Rules

You need to ensure that the inbound and outbound rules allow access from the proxy address. The default port number is 3306.

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

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

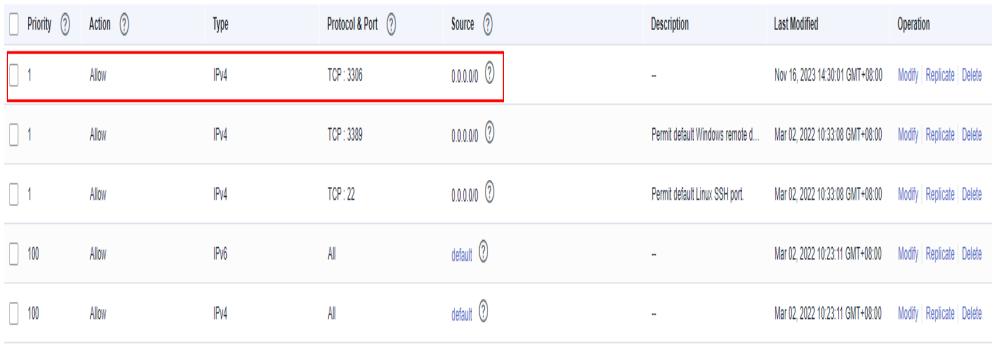
Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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 the security group name in the **Security Group** field.

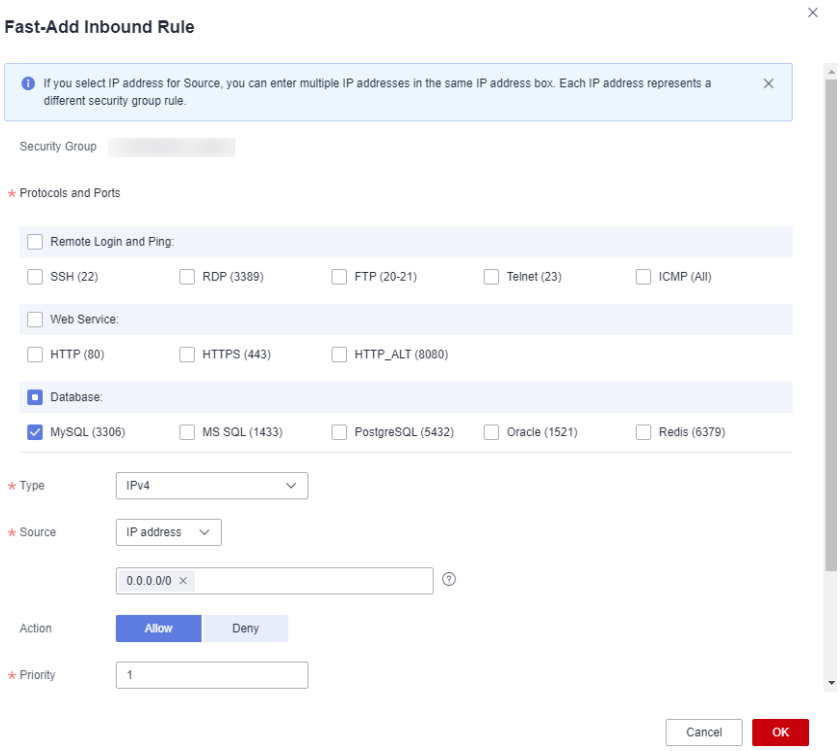
Step 6 On the **Inbound Rules** tab, check whether access through port **3306** is allowed by default.

Figure 14-8 Allowing access through port 3306



If there is no such a rule, click **Fast-Add Rule**. In the displayed dialog box, select **MySQL (3306)** and click **OK**.

Figure 14-9 Fast adding port 3306



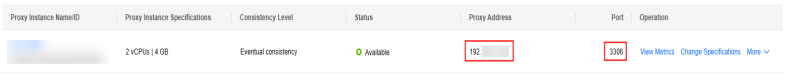
----End

Step 4: Use the Proxy Address to Connect to Your TaurusDB Instance

Step 1 View the proxy address and port on the TaurusDB console.

Click the TaurusDB instance name. In the navigation pane, choose **Database Proxy**. In the proxy instance list, view the proxy address and port.

Figure 14-10 Viewing a proxy address and port



Step 2 Log in to an ECS.

For details, see ***Elastic Cloud Server User Guide***.

Step 3 Run the following command to connect to the TaurusDB instance using the proxy address:

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

Table 14-4 Parameter description

Parameter	Description
<hostIP>	Proxy address obtained in **Step 1**.
<port>	Port obtained in **Step 1**.
<userName>	Username, that is, the TaurusDB database administrator account. The default value is **root**.
<password>	Password of the TaurusDB database administrator.

----End

Step 5: Verify Read/Write Splitting

After each read operation is complete, you can run the **show last route** command to view the routing result of the read operation.

The following is an example.

Step 1 After the TaurusDB instance is connected, perform a read operation.

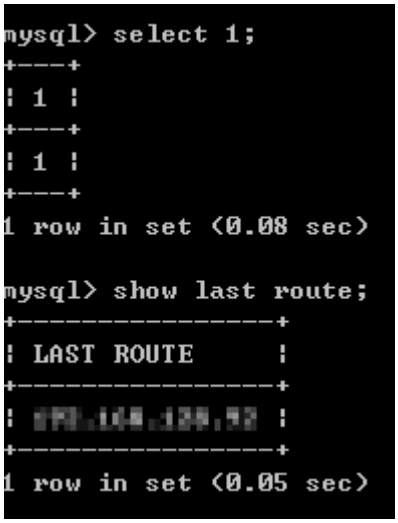
Example: **select 1;**

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


Step 2 Run the following command to view the routing result of the read operation in **Step 1**:

show last route

Figure 14-11 Viewing a query result



NOTE

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

----End

APIs

- **Creating a Proxy Instance**
- **Querying Proxy Instances**
- **Querying Proxy Instance Specifications**
- **Deleting a Proxy Instance**

14.3 Changing Configurations of a Proxy Instance

14.3.1 Changing the Consistency Level of a Proxy Instance

You can configure a consistency level when **creating a proxy instance** or change the consistency level of an existing proxy instance.

This section describes how to change the consistency level of a proxy instance.

Consistency Levels

There are several consistency levels to meet requirements in different scenarios.

- Eventual consistency (default)

  After a proxy instance is created, requests for SELECT operations are routed to different nodes based on their read weights. Because there is a replication

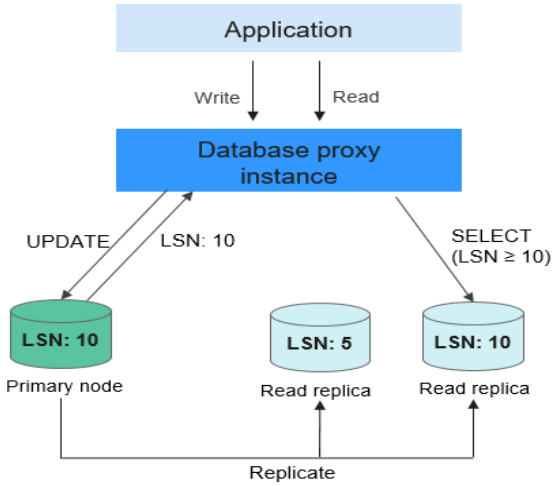
latency between the primary node and each read replica and the replication latency varies for different read replicas, the result returned by each SELECT statement may be different when you repeatedly execute a SELECT statement within a session. In this case, only eventual consistency is ensured.

- Session consistency

  To eliminate data inconsistencies caused by eventual consistency, session consistency is provided. Session consistency ensures the result returned by each SELECT statement in a session is the data that was updated after the last write request.

  Proxy instances record the log sequence number (LSN) of each node and session. When data in a session is updated, a proxy instance records the LSN of the primary node as a session LSN. When a read request arrives subsequently, the proxy instance compares the session LSN with the LSN of each node and routes the request to a node whose LSN is at least equal to the session LSN. This ensures session consistency.

Figure 14-12 Principle of session consistency



NOTE

In session consistency, if there is significant replication latency between the primary node and read replicas and the LSN of each read replica is smaller than the session LSN, requests for SELECT operations will be routed to the primary node. In this case, loads on the primary node are heavy and instance performance suffers.

Constraints

To use session consistency, the kernel versions of TaurusDB instances must be 2.0.54.1 or later, and the kernel versions of proxy instances must be 2.7.4.0 or later. For details about how to check the kernel version, see **How Can I Check the Version 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.

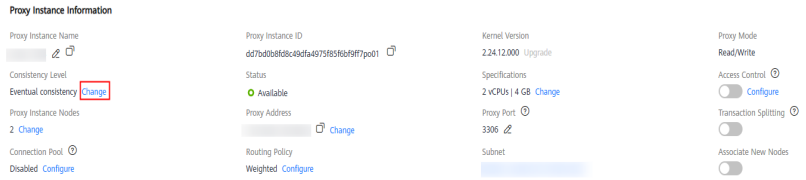
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

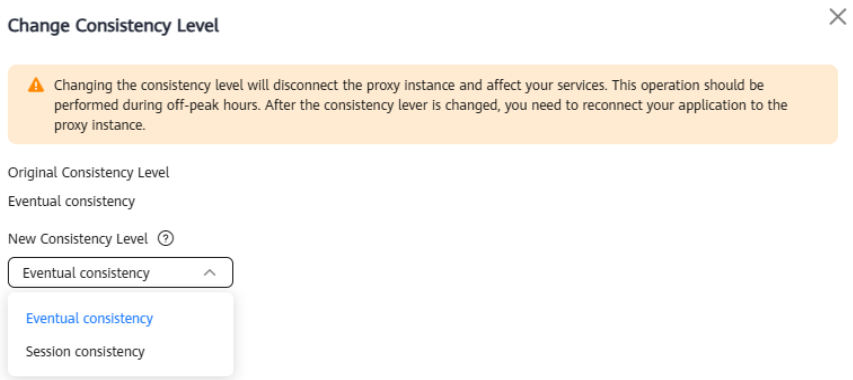
Step 6 Click the proxy instance name to go to the **Basic Information** page. In the **Proxy Instance Information** area, click **Change** under **Consistency Level**.

Figure 14-13 Changing a consistency level



Step 7 Click the drop-down arrow and select the required consistency level.

Figure 14-14 Changing a consistency level



NOTE

After the consistency level is changed, you need to manually reboot the proxy instance or re-establish the connection to the proxy instance on the management console.

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

Step 8 Click **OK**. The proxy instance status changes to **Changing consistency level**.

Step 9 After several minutes, check that the proxy instance status becomes **Available** and the consistency level is updated.

----End

APIs

Changing Session Consistency of a Proxy Instance

14.3.2 Enabling the Connection Pool for a Proxy Instance

A session-level connection pool helps reduce the database load caused by frequent establishment of short connections.

Connection Pool is disabled by default. You can enable a session-level connection pool.

A session-level connection pool is suitable for short connections. When your client disconnects from your database, the system checks whether the connection is idle. If it is, the system places the connection in the connection pool of a proxy instance and retains the connection for a short period of time. When your client re-initiates a connection, any available connection in the connection pool is used, reducing the overhead of establishing a new connection to the database. If no connections are available in the connection pool, a new connection will be established.

Constraints

- To use a connection pool, the kernel versions of proxy instances must be 2.22.07.000 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- When any of the following operations is performed, the connection is locked until the connection ends. That is, the connection will not be placed in the connection pool for other users to use.
  - Running the PREPARE statement
  - Creating a temporary table
  - Modifying user variables
  - Inserting or querying big data (for example, more than 16 MB)
  - Running the LOCK TABLE statement
  - Executing a multi-statement query (concatenated SQL statements with semicolons, for example, SELECT 1;SELECT 2)
  - Calling a stored procedure

Configuring a Connection Pool

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

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

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

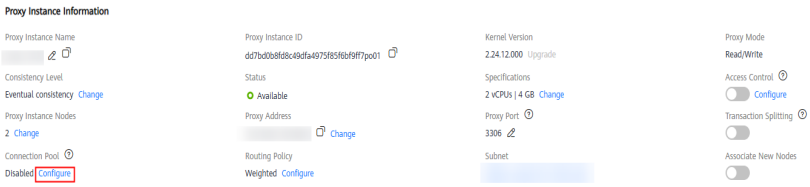
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

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

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click **Configure** under **Connection Pool**.

Figure 14-15 Configuring a connection pool



Step 8 In the displayed dialog box, set **Connection Pool** to **Session level** and click **OK**. This process takes several seconds.

Step 9 Check that the connection pool is successfully configured.

----End

APIs

- **Changing the Connection Pool Type of a Proxy Instance**
- **Querying Proxy Instances**
- **Querying Proxy Instance Specifications**

14.3.3 Enabling Transaction Splitting for a Proxy Instance

In most cases, a proxy instance sends all requests in transactions to the primary node to ensure transaction correctness. However, in some frameworks, all requests are encapsulated into transactions that are not automatically committed using **set autocommit=0**. This causes heavy load on the primary node.

With transaction splitting enabled for a proxy instance, the proxy instance can route read requests prior to write operations in a transaction to read replicas, reducing the load on the primary node.

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

Constraints

- The kernel versions of proxy instances must be 2.3.9.5 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- Transaction isolation levels of TaurusDB instances must be READ UNCOMMITTED or READ COMMITTED. The default isolation level is REPEATABLE READ.
- Proxy instances must be in the read/write mode.
- After transaction splitting is enabled, the transaction isolation level can only be changed to READ UNCOMMITTED or READ COMMITTED. To change the isolation level to a higher level, disable transaction splitting first.

Procedure

Step 1 Log in to the management console.

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

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

Step 6 Click the name of a proxy instance.

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

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

NOTE

- To disable transaction splitting, click  .
- Transaction splitting takes effect only for new connections established after this function is enabled or disabled.

Step 9 On the **Basic Information** page of the proxy instance, check that transaction splitting is enabled.

----End

APIs

- Enabling or Disabling Transaction Splitting for a Proxy Instance
- Querying Proxy Instances
- Querying Proxy Instance Specifications
- Deleting a Proxy Instance

14.3.4 Changing the Routing Policy of a Proxy Instance

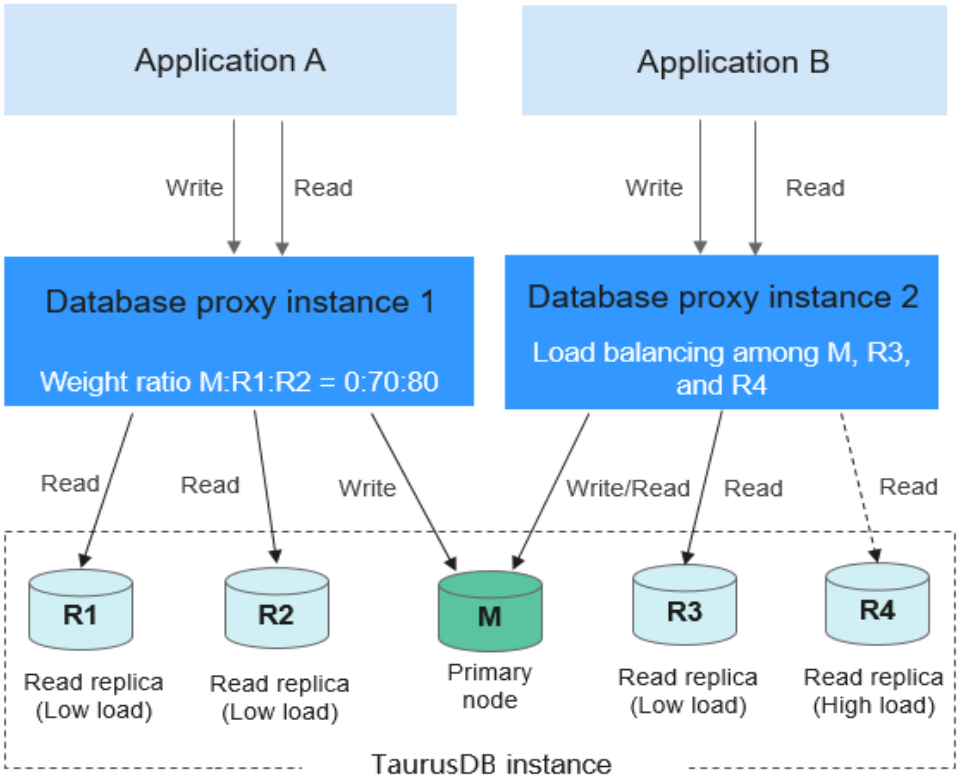
You can configure the routing policy when creating a proxy instance. The default routing policy is weighted. You can also change the routing policy of an existing instance.

Working Principles of the Routing Policy

There are weighted and load balancing routing policies.

- **Weighted**: Read requests are assigned to nodes based on the weights you specify.
- **Load balancing**: Read requests are assigned to nodes with fewer active connections. In the load balancing policy, you do not need to configure the weights of nodes.

Figure 14-16 Working principles of the routing policy



Example:

As shown in the preceding figure, a TaurusDB instance contains one primary node and four read replicas.

For the database proxy instance 1, the routing policy is weighted and the selected nodes include the primary node, read replica R1, and read replica R2, with their read weight ratio of 0:70:80. The write requests of the Application A are automatically forwarded to the primary node through the proxy instance, and the read requests are routed to read replicas R1 and R2 in the ratio of 7:8.

For the database proxy instance 2, the routing policy is load balancing and the selected nodes include the primary node, read replica R3, and read replica R4. The proxy instance determines the node to which the read requests are forwarded based on the number of real-time active connections.

When there are many active connections in read replica R4, the proxy instance forwards most read requests to read replica R3 and the primary node to offload the pressure of read replica R4.

Constraints

To use the load balancing policy, the kernel versions of proxy instances must be 2.22.07.000 or later. To upgrade a kernel version, see **Upgrading the Kernel Version of a Proxy Instance**. For details about how to check the kernel version, see **How Can I Check the Version 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 **Databases** > **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 **Database Proxy**.

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click **Configure** under **Routing Policy**.

Step 8 In the displayed dialog box, configure required parameters.

Figure 14-17 Changing the routing policy of a proxy instance

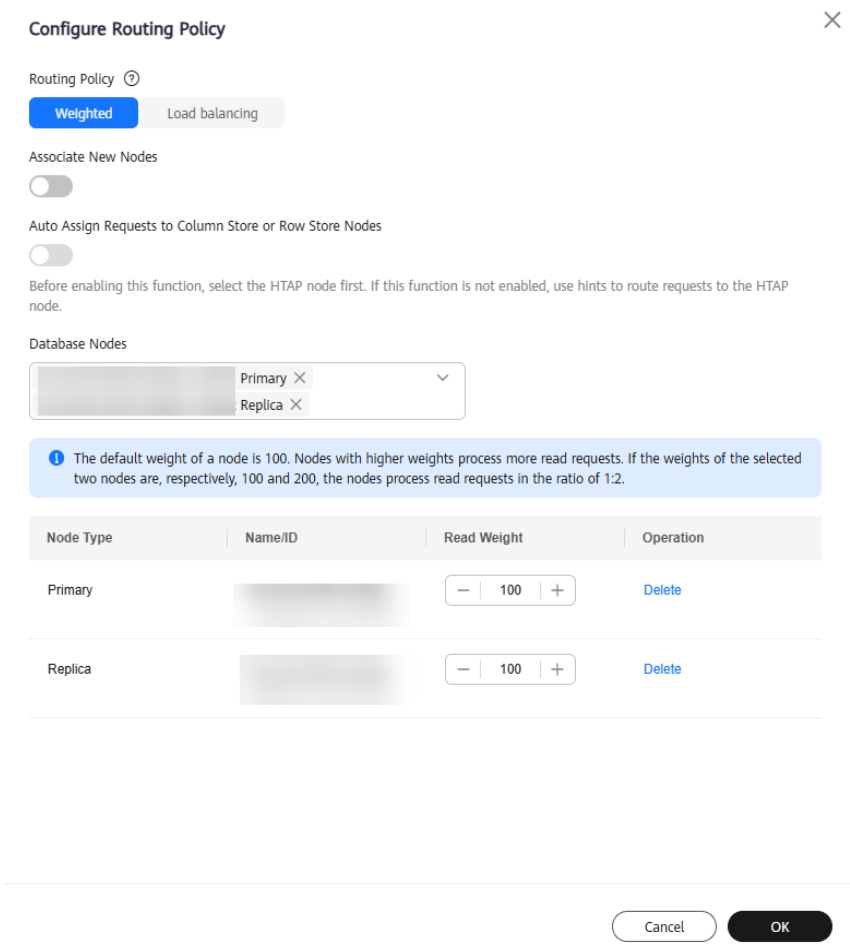


Table 14-5 Parameter description

Parameter	Description
Routing Policy	• **Weighted**: Read requests are assigned to nodes based on the weights you specify. • **Load balancing**: Read requests are assigned to nodes with fewer active connections. In the load balancing policy, you do not need to configure the weights of nodes.
Associate New Nodes	After this function is enabled, new read replicas will be automatically associated with the current proxy instance. If **Routing Policy** is **Weighted**, you need to configure read weights for the new nodes. The default weight of a node is 100. Nodes with higher weights process more read requests.
Database Nodes	The proxy mode of a proxy instance determines which nodes read requests are assigned to. • Read-only mode: All read requests are assigned to the selected, but not to the primary node. • Read/write mode: All read requests are assigned to the selected nodes (including the primary node and read replicas) based on the routing policy.

Step 9 Click **OK**. The proxy instance status changes to **Configure routing policy**.

Step 10 After several minutes, check that the proxy instance status becomes **Available** and the routing policy is updated.

----End

APIs

Changing the Routing Policy of a Proxy Instance

14.3.5 Changing Read Weights of Nodes

After a proxy instance is created, you can change the read weights of its associated nodes. Read requests are forwarded to each node based on the read weights you specify, enabling read/write splitting and reducing the load of the primary node.

Constraints

- **This function is only available for proxy instances that use the weighted routing policy.**
- You can configure read weights for both the primary node and read replicas.
- The default read weight of the primary node is 0. The higher read weight the primary node is assigned, the more read requests it can process.
- When the read weights of all nodes are 0, services are not affected. In this case, the primary node processes all read and write requests by default.

- The weight of a read replica ranges from 0 to 1000.
- After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the current proxy instance. The default read weight of any new node is 100.
- After a read replica is deleted, its weight is automatically removed while the weights of other read replicas remain unchanged.

Procedure

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

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

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

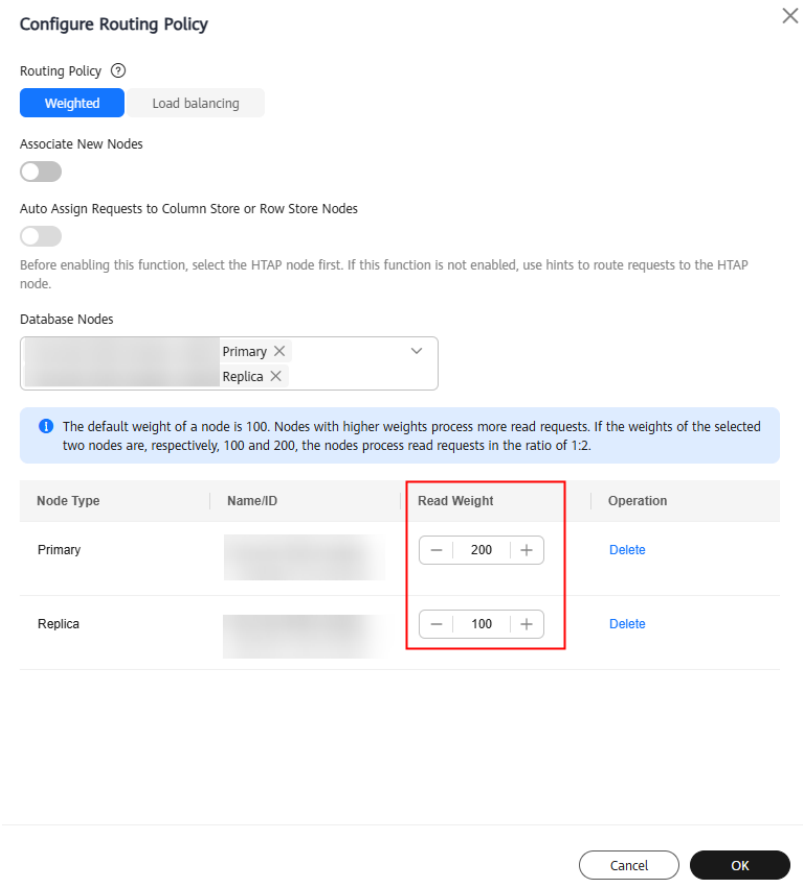
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

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

Step 6 On the **Basic Information** page, click **Configure** under **Routing Policy**.

Step 7 In the displayed dialog box, select the nodes that you want to associate with the current proxy instance or deselect the nodes that you want to remove from the current proxy instance in the **Database Nodes** area, and configure read weights in the **Read Weight** column.

Figure 14-18 Configuring read weights

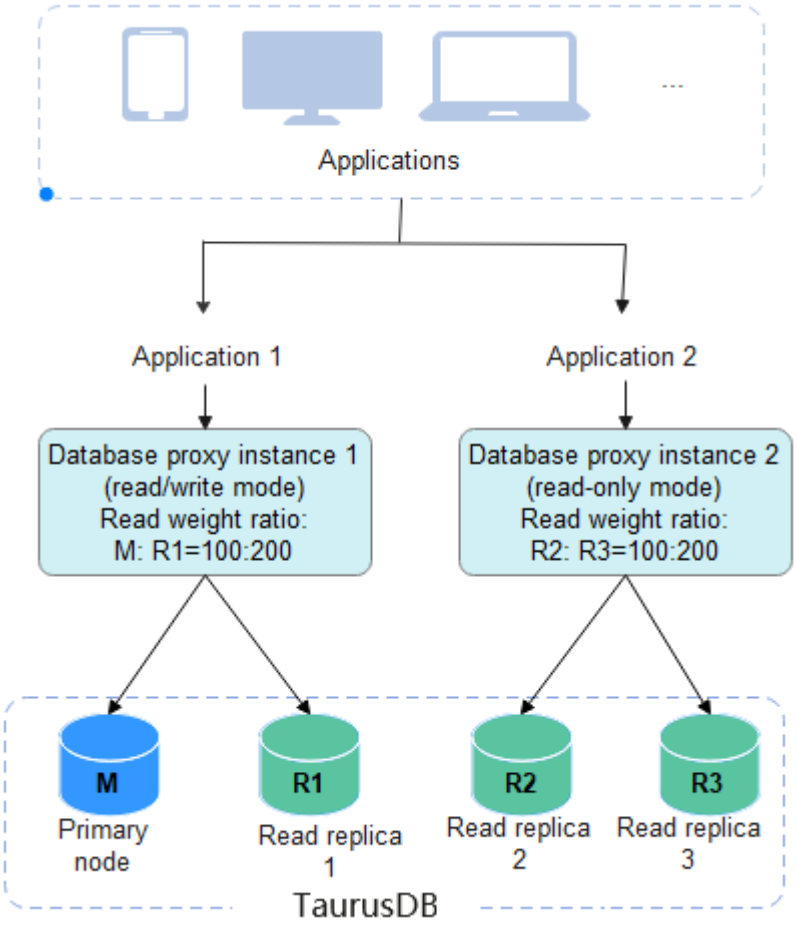


Example:

As shown in **Figure 14-19**, one TaurusDB instance has one primary node and three read replicas. Two proxy instances have been created and they both use the weighted routing policy.

- Proxy instance 1 is in the read/write mode. The primary node and read replica 1 are associated with proxy instance 1 and assigned with a read weight of 100 and 200, respectively. They process read requests in the ratio of 1:2, that is, the primary node processes 1/3 read requests and read replica 1 processes 2/3 read requests. Write requests are automatically routed to the primary node.
- Proxy instance 2 is in the read-only mode. Read replica 2 and read replica 3 are associated with proxy instance 2 and assigned with a read weight of 100 and 200, respectively. Read replica 2 and read replica 3 process read requests in the ratio of 1:2, that is, read replica 2 processes 1/3 read requests, and read replica 3 processes 2/3 read requests.

Figure 14-19 Read/Write splitting in multi-proxy scenarios (weighted routing policy)



Step 8 Click **OK**. The proxy instance status changes to **Configure routing policy**.

Step 9 After several minutes, check that the proxy instance status becomes **Available** and the read weights are updated.

----End

APIs

Assigning Read Weights

14.3.6 Changing the Multi-statement Processing Mode of a Proxy Instance

When you enable **multi-statement execution** for a proxy instance, you can set multi-statement processing mode to **Strict**, **Loose**, or **Parse**.

- **Strict** (default)

  If a request containing multiple statements is routed to the primary node, the subsequent requests are all routed to the primary node. Read/write splitting

can be restored only after you disconnect the current connection and reconnect it.

Your proxy instance will not parse these statements, so the performance is better. It is suitable for short connections.

- **Loose**

  If a request containing multiple statements is routed to the primary node, the subsequent requests of the current connection can still be routed to the primary node or read replicas.

  Your proxy instance will not parse these statements, so the performance is better. It is good for when multiple statements contain only DML SQL statements and do not contain operations like setting session variables, creating temporary tables, creating stored procedures, or executing uncommitted transactions.

- **Parse**

  A read-only request containing multiple statements is routed based on weights. A read/write request containing multiple statements is routed to the primary node, and your proxy instance parses these statements and determines whether to split subsequent read and write requests received over the current connection based on the operations in the SQL statements (**Parse-based mode description**).

  Parsing statements affects the proxy instance performance. The degree of the impact depends on the length and complexity of statements. It is recommended that the statements be less than 100 MB.

Constraints

- To set the multi-statement processing mode on the management console, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- The changed multi-statement processing mode applies to your proxy instance immediately. You do not need to reboot the proxy instance. If a read/write splitting connection fails due to a multi-statement execution, changing the multi-statement processing mode will not restore the connection. You will need to reconnect the connection manually.
- Parse-based mode description:

  If multi-statements contain the operations listed here, all subsequent requests are routed to the primary node. To restore read/write splitting, you need to disconnect the connection and then re-establish it.

  - Creating temporary tables
  - Creating stored procedures
  - Executing uncommitted transactions (for example, **begin** is executed but **commit** or **rollback** is not executed)
  - Executing complex or special syntax. These statements will not be parsed.

Procedure

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

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

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

Step 6 Click a proxy instance name to go to the **Basic Information** page.

Step 7 In the navigation pane, choose **Parameter Modifications**.

Step 8 Configure the parameter **multiStatementType** as required.

Figure 14-20 Configuring the parameter **multiStatementType**

You can set this parameter to **Strict**, **Loose**, or **Parse**.

Step 9 Click **Save** to save your change. In the displayed dialog box, click **Yes**.

----End

14.3.7 Enabling Automatic Association of New Nodes with a Proxy Instance

After **Associate New Nodes** is enabled, new read replicas will be automatically associated with the current proxy instance.

This section describes how to enable or disable **Associate New Nodes** for an existing proxy instance. To enable this function during the proxy instance creation, see **Creating a Proxy Instance for Read/Write Splitting**.

Procedure

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

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

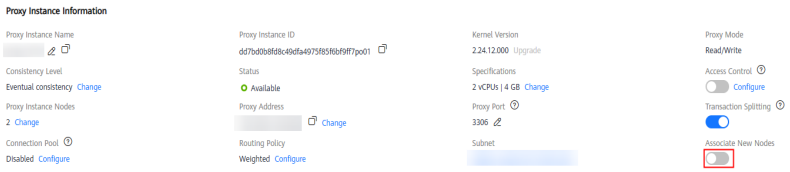
Step 3 Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.

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

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

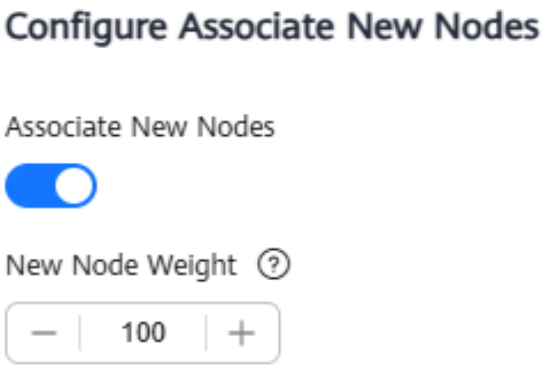
Step 6 In the **Proxy Instance Information** area, click under **Associate New Nodes**.

Figure 14-21 Going to the Basic Information page



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

Figure 14-22 Enabling automatic association of new nodes with a proxy instance



When the routing policy is weighted, you need to configure weights for the new nodes as required. The default read weight of any new node is 100. Nodes with higher weights process more read requests.

Step 8 Click **OK**.

To disable the function, click  .

----End

14.3.8 Enabling Access Control for a Proxy Instance

If load balancing is enabled for a proxy instance, the security group associated with the proxy instance does not apply. You need to use access control to limit access from specific IP addresses.

Constraints

If access control is not displayed on the management console, the security group associated with the proxy instance is used.

Enabling Access Control

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

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

Step 3 Click in the upper left corner of the page, choose **Databases** > **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 **Database Proxy** and click the name of a proxy instance to go to the **Basic Information** page.

Step 6 Click under **Access Control**.

Step 7 Click **Configure**. In the displayed dialog box, configure required parameters.

Figure 14-23 Configuring access control

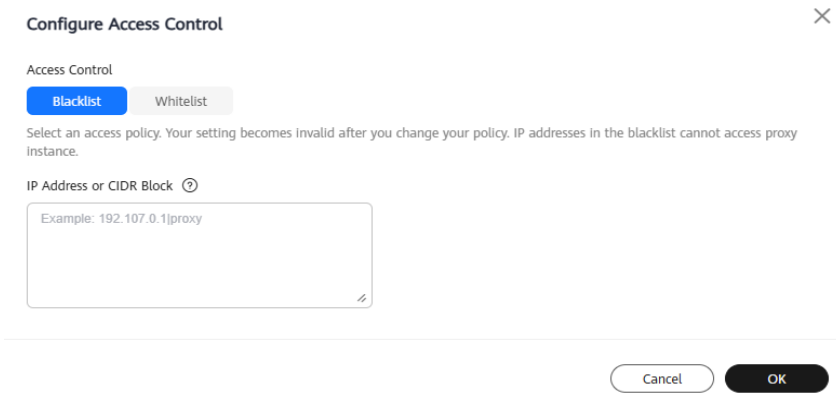


Table 14-6 Parameter description

Parameter	Description
Access Control	The blacklist and whitelist cannot be configured at the same time. If you switch between lists, your previously entered settings will be lost. IP addresses or CIDR blocks in the whitelist can access the proxy instance, while those in the blacklist cannot.
IP Address or CIDR Block	You need to enter IP addresses or CIDR blocks that meet the following requirements: • Each line contains an IP address or a CIDR block and ends with a line break. • Each IP address or CIDR block can include a description separated by a vertical bar symbol (|), for example, **192.168.10.10|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.

Step 8 On the **Basic Information** page of the proxy instance, check that access control is enabled. Click **Configure** to view the new access control setting.

----End

Disabling Access Control

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

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

Step 3 Click  in the upper left corner of the page, choose **Databases** > **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 **Database Proxy** and click the name of a proxy instance to go to the **Basic Information** page.

Step 6 Click  under **Access Control**.

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

Step 8 Refresh the page later and check that the access control is disabled.

----End

14.3.9 Enabling Binlog Pull for a Proxy Instance (OBT)

Scenarios

You can enable binlog pull for a proxy instance and then use the proxy address to pull binlogs from the primary node or read replicas.

Constraints

- To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- This function is only available when the kernel version of the proxy instance is 2.24.12.000 or later and that of the TaurusDB instance is 2.0.54.240600 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- After this function is enabled for a read-only proxy instance, binlogs can only be pulled from read replicas. So, you must enable binlog pull for read replicas of the TaurusDB instance first.
- If you want to pull binlogs from a different node (such as from a read replica instead of the primary node), you must disconnect the current connection and reconnect to the desired node.
- Pulling binlogs from a proxy instance consumes its resources, which increases pressure on the proxy instance and even causes performance issues when binlogs are pulled concurrently.

Enabling Binlog Pull for the Primary Node

Step 1 Log in to the management console.

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

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

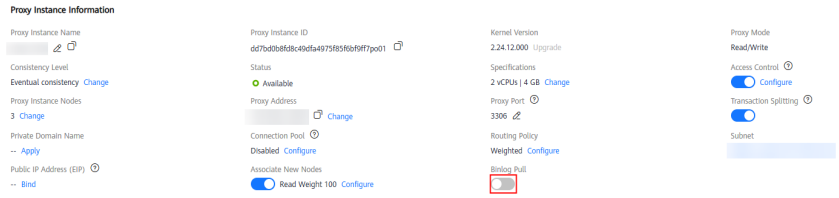
Step 4 On the **Instances** page, click the instance name.

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

Step 6 Click the proxy instance name to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click on the right of **Binlog Pull**.

Figure 14-24 Enabling binlog pull



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

----End

Enabling Binlog Pull for Read Replicas

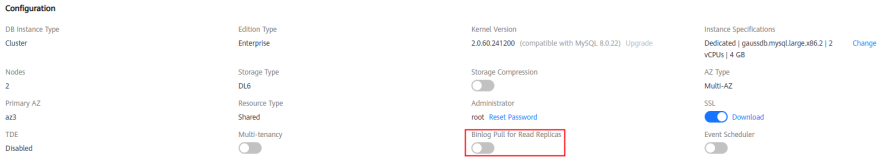
Step 1 Log in to the management console.

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

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

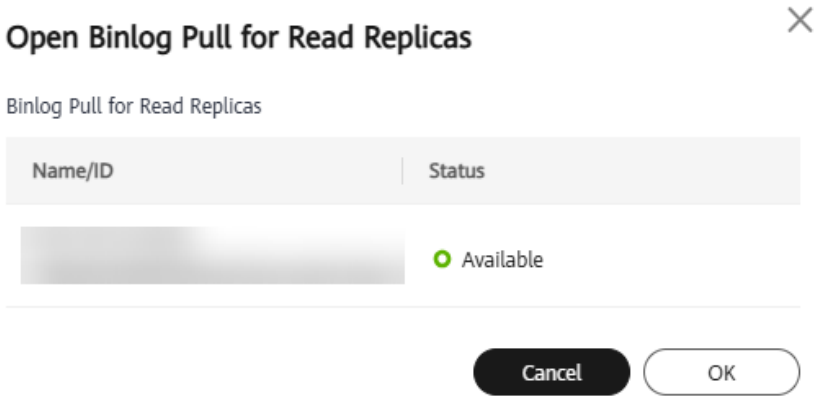
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **Configuration** area, click under **Binlog Pull for Read Replicas**.



Step 6 In the displayed dialog box, click **OK**. The DB instance status changes from **Binlog pull is being configured for the read replicas** to **Available**. This process takes several seconds.

Figure 14-25 Enabling binlog pull for read replicas

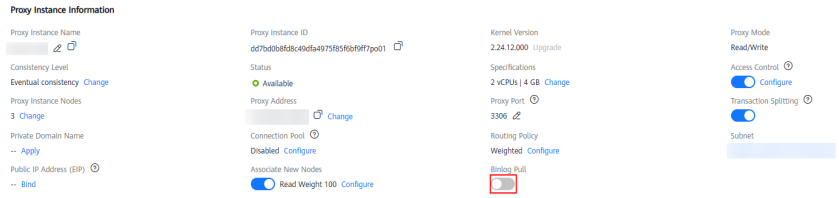


Step 7 In the navigation pane, choose **Database Proxy**.

Step 8 Click the proxy instance name to go to the **Basic Information** page.

Step 9 In the **Proxy Instance Information** area, click  on the right of **Binlog Pull**.

Figure 14-26 Enabling binlog pull



Step 10 In the displayed dialog box, click **OK**. The proxy instance status changes from **Changing binlog pull of the proxy instance** to **Available**. This process takes several seconds.

Step 11 Check whether **Binlog Pull for Primary Node** is enabled. If it is, click  to disable it. Then, binlogs will only be pulled from read replicas.

Figure 14-27 Checking whether **Binlog Pull for Primary Node** is enabled

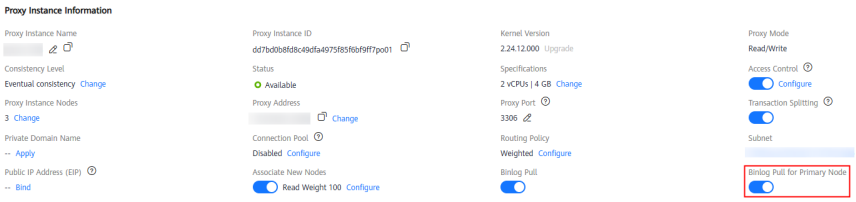
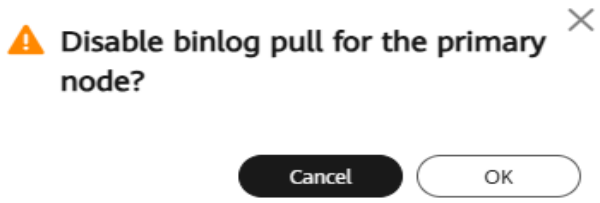


Figure 14-28 Disabling binlog pull for the primary node



----End

14.3.10 Changing the Specifications of a Proxy Instance

If the proxy instance specifications cannot meet your workload requirements, you can manually upgrade them.

Constraints

- The proxy instance specifications can be changed only when your TaurusDB instance, primary node, and read replicas are all normal.
- A proxy instance cannot be deleted when its CPU and memory specifications are being changed.

Procedure

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

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

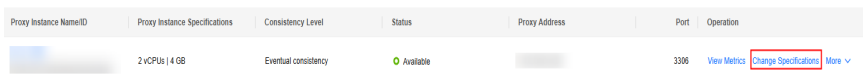
Step 3 Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.

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

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

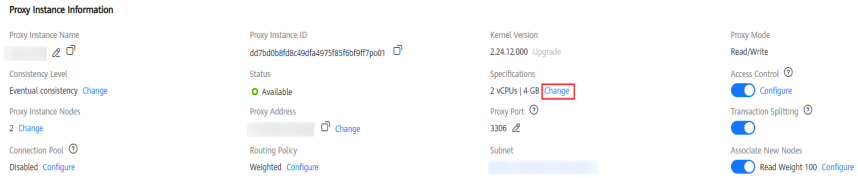
Step 6 On the **Database Proxy** page, locate the desired proxy instance and choose **More** > **Change Specifications** in the **Operation** column.

Figure 14-29 Changing proxy instance specifications (1)



Alternatively, click the proxy instance name. In the **Proxy Instance Information** area, click **Change** under **Specifications**.

Figure 14-30 Changing proxy instance specifications (2)



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

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

----End

APIs

- Changing the Specifications of a Proxy Instance
- Querying Proxy Instances
- Querying Proxy Instance Specifications

14.3.11 Changing the Number of Nodes for a Proxy Instance

Scenarios

You can change the number of proxy instance 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.
- The number of proxy nodes ranges from 2 to 16.

Procedure

Step 1 Log in to the management console.

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

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

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

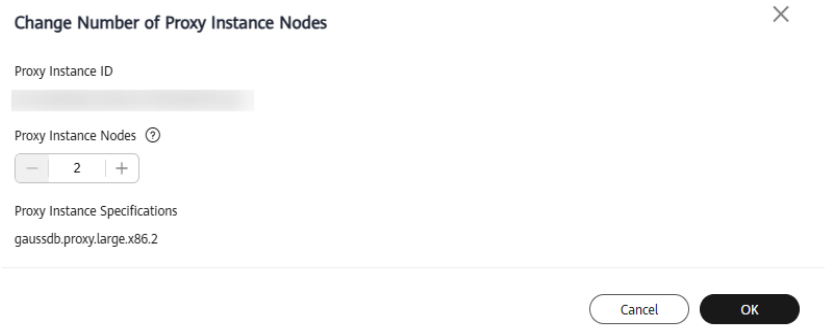
Step 5 In the navigation pane, choose **Database Proxy**. Click the name of a proxy instance.

Step 6 In the **Proxy Instance Information** area, click **Change** under **Proxy Instance Nodes**.

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

Number of recommended proxy instance nodes = (Number of vCPUs of the primary node + Total number of vCPUs of all read replicas)/(4 x Number of vCPUs of the proxy instance), rounded up.

Figure 14-31 Changing the number of proxy nodes



Step 8 After several minutes, check that the proxy instance status changes from **Adding nodes** to **Available**.

----End

APIs

- Adding Proxy Nodes
- Querying Proxy Instances
- Deleting Proxy Nodes

14.3.12 Applying for a Private Domain Name for a Proxy Instance (OBT)

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

Constraints

To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Applying for a Private Domain Name for a Proxy Instance

Step 1 Log in to the management console.

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

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

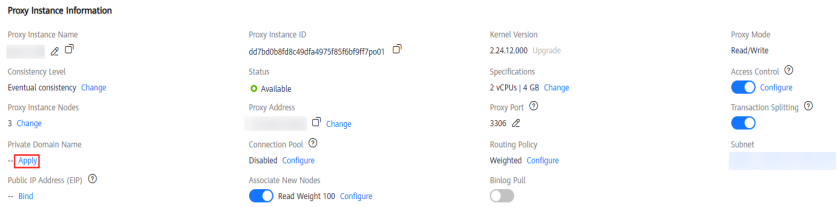
Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

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

Step 6 Click the name of a proxy instance.

Step 7 In the **Proxy Instance Information** area of the **Basic Information** page, click **Apply** under **Private Domain Name**.

Figure 14-32 Applying for a private domain name



Step 8 In the **Private Domain Name** field, view the generated private domain name.

----End

Changing the Private Domain Name of a Proxy Instance

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

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

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

Step 6 In the **Proxy Instance Information** area of the **Basic Information** page, click **Change** under **Private Domain Name**.

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

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

----End

Deleting the Private Domain Name of a Proxy Instance

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

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

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

Step 6 In the **Proxy Instance Information** area of the **Basic Information** page, click **Delete** under **Private Domain Name**.

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

----End

14.3.13 Changing the Port of a Proxy Instance

Scenarios

You can change the port for a proxy instance.

Constraints

- Changing the port of a proxy instance will interrupt the database connection. You are advised to change the port during off-peak hours.
- Changing the port of a proxy instance does not reboot the proxy instance.
- Only the port of the current proxy instance will be changed.

Procedure

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

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

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

Step 6 Click the name of a proxy instance.

Step 7 On the **Basic Information** page, click under **Proxy Port**.

Proxy port range: 1025 to 65534 (except for 1033, 5342, 5343, 5344, 5345, 12017, 20000, 20201, 20202, 33060, 33062, and 33071, which are reserved by the system)

Step 8 Click to submit the change.

----End

APIs

Changing the Port of a Proxy Instance

14.3.14 Changing the Proxy Address of a Proxy Instance

Scenarios

You can change the proxy address of a proxy instance.

Constraints

- Changing a proxy address will interrupt database connections and services. Perform the operation during off-peak hours or when services are stopped.
- The new proxy address is not in use and must be associated with the same subnet as your 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.

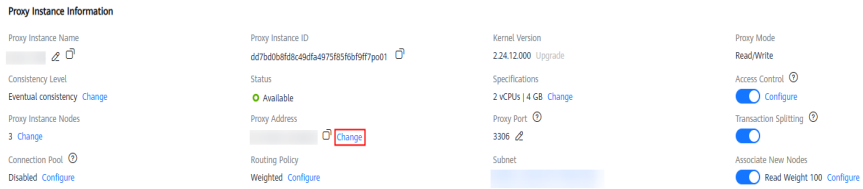
Step 3 Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.

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

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

Step 6 Click the name of a proxy instance. In the **Proxy Instance Information** area, click **Change** under **Proxy Address**.

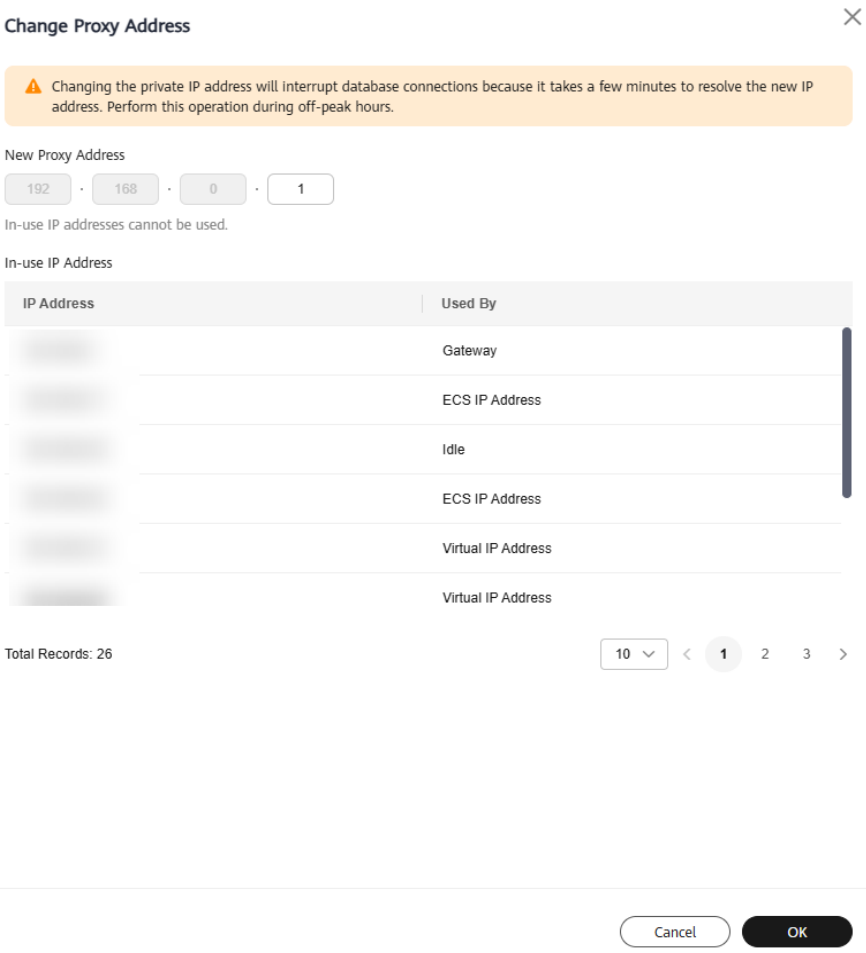
Figure 14-33 Changing a proxy address



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

In-use IP addresses cannot be used.

Figure 14-34 Changing a proxy address



Step 8 Check the task progress on the **Task Center** page and refresh the **Basic Information** page of the proxy instance later. If the proxy instance status changes from **Changing proxy address** to **Available**, the proxy address is changed successfully. You can view the new proxy address on the page.

----End

14.3.15 Modifying Parameters of a Proxy Instance

Scenarios

You can change parameters for a proxy instance.

Procedure

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

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

Step 3 Click in the upper left corner of the page, choose **Databases** > **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 Modifications**. On the displayed page, change parameters if needed.

You can save, cancel, or preview your changes.

- To save your changes, click **Save**.
- To cancel your changes, click **Cancel**.
- To preview your changes, click **Preview**.

----End

14.3.16 Binding an EIP to a Proxy Instance (OBT)

After a proxy instance is created, you can bind an EIP to it. Later, you can also unbind the EIP from the proxy instance as required.

Constraints

To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Billing

- 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 **Databases** > **TaurusDB**.

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

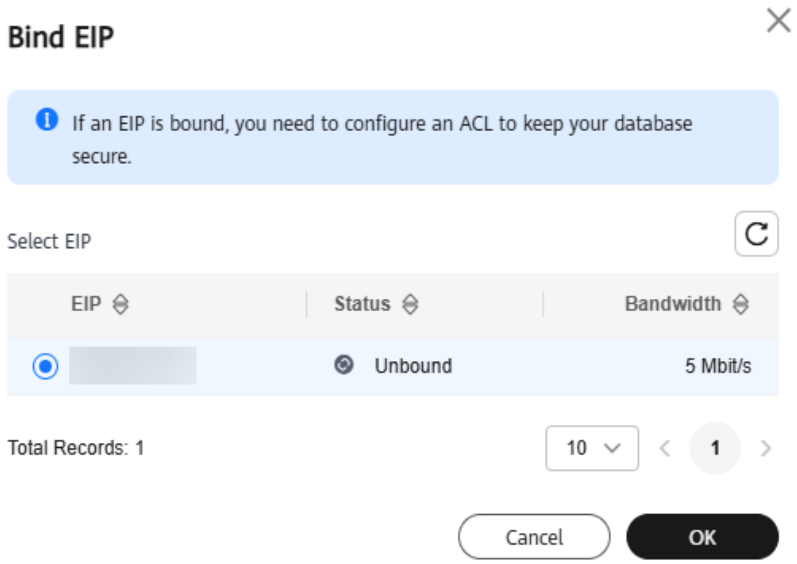
Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click a proxy instance name to go to the **Basic Information** page.

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

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

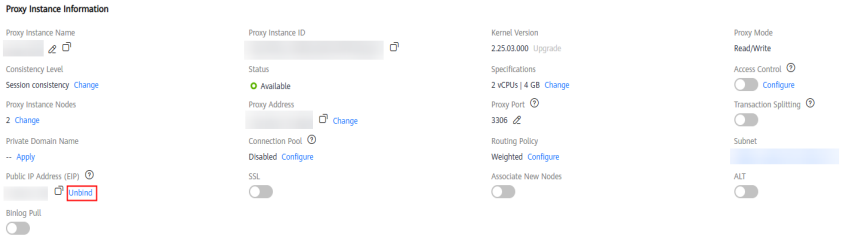
Figure 14-35 Binding an EIP to a proxy instance



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

To unbind an EIP from the proxy instance, click **Unbind** under **Public IP Address (EIP)**. In the displayed dialog box, click **OK**.

Figure 14-36 Unbinding an EIP from a proxy instance



----End

14.4 Proxy Instance Lifecycle

14.4.1 Rebooting a Proxy Instance

Scenarios

You can reboot a proxy instance you have created.

Constraints

- If the proxy instance status is **Abnormal**, the reboot may fail.

- Reboot a proxy instance interrupts the database connection. You are advised to reboot it during off-peak hours. To shorten the time required, reduce database activities during the reboot to reduce rollback of transit transactions.

Procedure

Step 1 Log in to the management console.

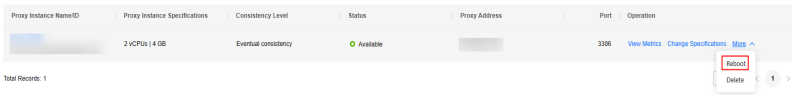
Step 2 Click in the upper left corner and select a region and project.

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

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

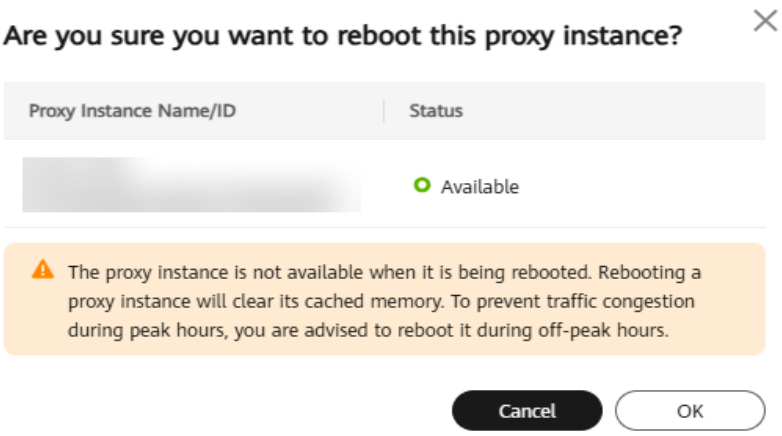
Step 5 In the navigation pane, choose **Database Proxy**, locate the target proxy instance, and choose **More** > **Reboot** in the **Operation** column.

Figure 14-37 Rebooting a proxy instance



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

Figure 14-38 Confirming information



Step 7 Check the reboot progress on the **Task Center** page. If the task status becomes **Completed** and the proxy instance status becomes **Available**, the proxy instance is rebooted successfully.

----End

14.4.2 Deleting a Proxy Instance

You can delete a proxy instance as required.

Constraints

If a proxy instance is deleted, read/write splitting is disabled and workloads using the proxy address are interrupted. You need to connect your applications to the TaurusDB instance address.

Procedure

Step 1 Log in to the management console.

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

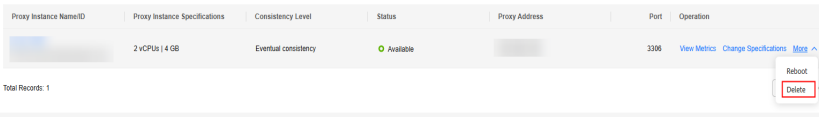
Step 3 Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.

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

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

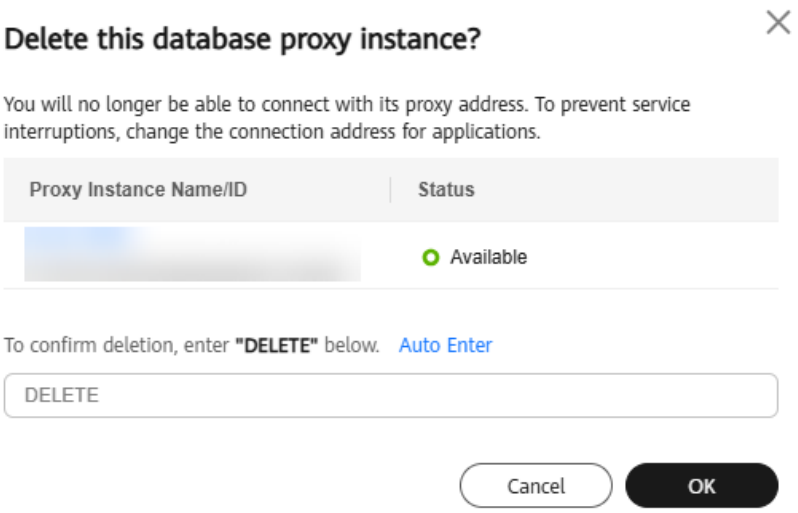
Step 6 Select the target proxy instance and choose **More** > **Delete** in the **Operation** column.

Figure 14-39 Deleting a proxy instance



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

Figure 14-40 Deleting a proxy instance



Step 8 Check the task progress on the **Task Center** page and refresh the **Basic Information** page of the proxy instance later. If the proxy instance status becomes **Available**, the proxy instance is deleted successfully.

----End

APIs

- **Creating a Proxy Instance**
- **Deleting a Proxy Instance**

14.5 Proxy Instance Kernel Versions

14.5.1 Proxy Instance Kernel Version Release History

Released On	Version	Description
2025-01-21	2.24.12.000	New feature: HTAP of Standard Edition Fixed issues: • Optimized the slow query log processing logic. • Optimized binlog pull for proxy instances. • Optimized .NET client adaptation.
2024-11-30	2.24.09.020	New features: • IPv6 for proxy instances • Multi-tenancy for TaurusDB instances Fixed issues: • Fixed the issue that proxy resource reclamation is slow after SSL is enabled. • Fixed the issue that read requests cannot be split after transaction splitting is enabled, **set autocommit=0** is used to start a transaction, and **commit** is used to commit the transaction. • Optimized the batch package processing logic. • Optimized the **resetConnection** processing logic.
2024-07-30	2.24.06.000	• Added binlog pulling through the proxy instance kernel. • Fixed the issue that after transaction splitting is enabled, read requests after SELECT FOR UPDATE are sent to the primary node.

Released On	Version	Description
2024-05-07	2.24.03.000	Added the feature for assigning requests to row and column store nodes.
2024-01-15	2.23.12.000	• Added the feature for collecting statistics on slow query logs of proxy instances. • Fixed the issue that there is a delay when a proxy instance synchronizes authentication information from the database kernel.
2024-01-04	2.23.09.002	Fixed the logic for proxy instances to retry service SQL statements after the database is faulty.
2023-11-13	2.23.09.001	Fixed the issue that an error is occasionally reported during execution of the prepared **SELECT FOR UPDATE** statement.
2023-10-20	2.23.09.000	New features: • **Change User** protocol • Parsing of multiple hints • **SHOW PROCESSLIST** and **KILL** commands Fixed the issue that the **set autocommit** setting is synchronized to read replicas after transaction splitting is enabled.
2023-07-31	2.23.06.001	Resolved the increased backend database connections caused by enabling session connection pool.
2023-07-06	2.23.06.000	• Added binlog pulling through the proxy instance kernel. • Optimized the performance of the **PREPARE STMT** protocol again.
2023-06-11	2.23.02.007	Fixed issues: • Optimized the performance of the **PREPARE STMT** protocol. • Resolved unexpected traffic allocation of the **/* FORCE_SLAVE*/ Hint** statement.
2023-04-27	2.23.02.000	Optimized the proxy instance performance.
2022-12-05	2.22.11.000	Added multi-statement processing modes. Optimized the error messages reported during SQL statement execution in some scenarios.

Released On	Version	Description
2022-09-06	2.22.07.000	New features: • Session-level connection pooling • Dynamic load balancing Optimized the logic for setting session-level transaction isolation levels of proxy instances. By default, the transaction isolation levels are synchronized with those of the database.
2022-06-15	2.7.5.0	Added Application Lossless and Transparent (ALT).
2022-05-06	2.7.4.0	New features: • A query for more than 16 MB of data • Session consistency Optimized the way how metrics of read-only proxy instances are collected by Cloud Eye.
2022-04-01	2.3.9.8	Added batch execution of prepared statements.
2022-02-09	2.3.9.7	New features: • Transaction splitting • Read-only mode Optimized the execution logic of prepared statements to improve performance.
2021-04-23	2.3.9.0	Added proxy instance performance metrics **Front-End Connections Created per Second**, **Transaction Queries per Second**, and **Multi-Statement Queries per Second**. Fixed issues: • Optimized the database proxy performance. • Fixed traffic congestion occurring when your applications connect to a proxy instance over short connections.
2021-01-14	2.3.8.0	Added the feature for obtaining client IP addresses through proxy instances. Fixed issues: • Fixed the issue that monitoring data of database proxy is inaccurate. • Shortened the downtime of proxy instances during a primary/standby switchover.

Released On	Version	Description
2020-10-14	2.3.6.0	Fixed issues: • Fixed the issue of connection failures caused by database overload. • Improved proxies' compatibility with MySQL protocols.
2020-08-14	2.3.1.0	New features: • Maintaining connectivity between clients and database proxies. • Monitoring performance metrics of proxy instances.

14.5.2 Upgrading the Kernel Version of a Proxy Instance

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

Upgrade Methods

A minor kernel version can be upgraded in either of the following ways:

- Upon submission: The system **upgrades the minor kernel version** upon your manual submission of the upgrade request.
- In maintenance window: The system upgrades the minor kernel version during the maintenance window you have specified. For details about how to change the maintenance window, see **Changing the Maintenance Window of a DB Instance**.

If the kernel version of your DB instance has potential risks or major defects, has expired, or has been brought offline, the system will notify you by SMS message or email and deliver an upgrade task during the maintenance window.

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 and choose **Databases** > **TaurusDB**.

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

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

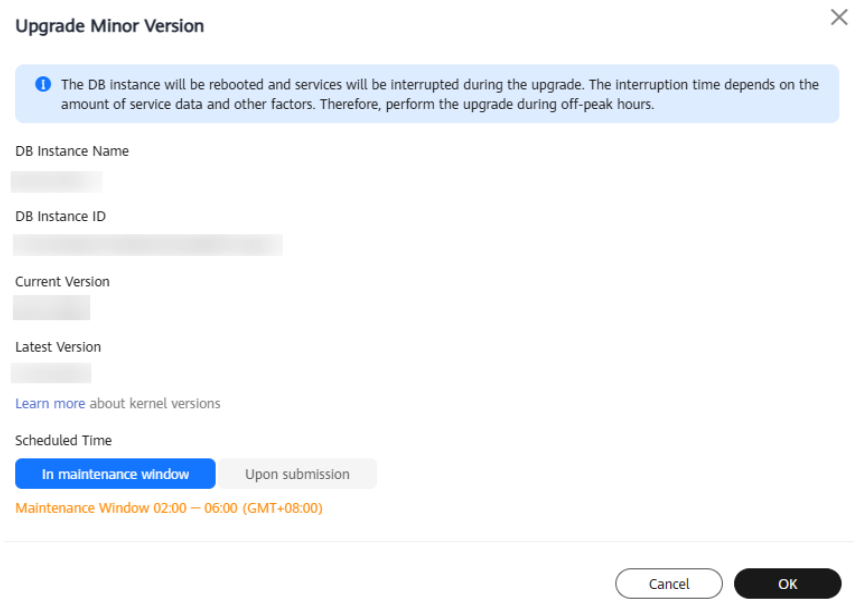
Step 6 Click the name of a proxy instance to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click **Upgrade** under **Kernel Version**.

Step 8 In the displayed dialog box, set **Scheduled Time** and click **OK**.

- **Upon submission**: The system upgrades the proxy instance to the latest version immediately after you submit the request. You can view the task progress in **Task Center** > **Instant Tasks**.
- **In maintenance window**: The system upgrades the proxy instance to the latest version during a maintenance window. You can view the task progress in **Task Center** > **Scheduled Tasks**.

Figure 14-41 Upgrading the minor version of a proxy instance



----End

14.6 Using Hints for Read/Write Splitting

In addition to configuring weights of nodes for read/write splitting, you can use hints in SQL statements to route read and write requests to a primary node or read replica.

Precautions

- Hints are only used as routing suggestions. In non-read-only SQL and non-transaction scenarios, SQL statements cannot be routed to read replicas.
- If you want to connect to a DB instance using the MySQL CLI and hints, add the **-c** option.

Usage

You can add the following hints at the beginning of an SQL statement as needed.

/*FORCE_MASTER*/: The SQL statement is executed on the primary node.

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

For example, if you run **select * from table1**, the SQL statement will be executed on a read replica by default. If you change it to **/*FORCE_MASTER*/ select * from table1**, the SQL statement will be executed on the primary node.

CAUTION

/*FORCE_MASTER*/ only works for read/write addresses. If your primary node is read-only, adding **/*FORCE_MASTER*/** will not help route the SQL statement to the primary node.

15 DBA Assistant

15.1 What Is DBA Assistant?

DBA Assistant provides visualized database O&M and intelligent diagnosis for developers and database administrators (DBAs), making database O&M easy and efficient. By analyzing alarms, resource usage, health status, performance metrics, and storage usage, it helps you quickly locate faults and keep track of instance statuses.

Prerequisites

To use DBA Assistant on the TaurusDB console, IAM users must have the **GaussDB FullAccess**, **DAS FullAccess**, **DAS Administrator**, and **CES FullAccess** permissions. For details, see **Creating a User and Granting TaurusDB Permissions**.

Functions

Table 15-1 lists the functions supported by DBA Assistant.

Table 15-1 Function description

Function	Description	Reference
Dashboard	Shows the status of your instance, including alarms, resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.	**Viewing the Overall Status of a DB Instance**

Function	Description	Reference
Sessions	The **Sessions** page displays slow sessions, active sessions, and total sessions. You can quickly filter slow sessions or active sessions by user, host IP address, or database name. Kill Session and SQL Throttling can be used for urgent instance recovery to ensure database availability.	**Managing Real-Time 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.	**Performance Monitoring**
Storage Analysis	Storage occupied by data and logs and changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DBA Assistant can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner. **Autoscaling**, **Tablespaces**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.	**Managing Storage**
Slow Query Logs	Displays slow queries within a specified time period. You can view top 5 slow query logs by user or IP address, sort statistics, and identify sources of slow SQL statements.	**Viewing Slow Query Logs**
SQL Explorer	After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.	• **Viewing Top SQL Statements** • **Creating a SQL Insights Task**
SQL Throttling	SQL Throttling restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.	**Configuring SQL Throttling**

15.2 Performance Monitoring

15.2.1 Viewing the Overall Status of a DB Instance

The **Dashboard** page allows you to view the overall status of the current DB instance, including alarms, health check results, compute resource usage, storage resource usage, and key performance metrics.

Alarms

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

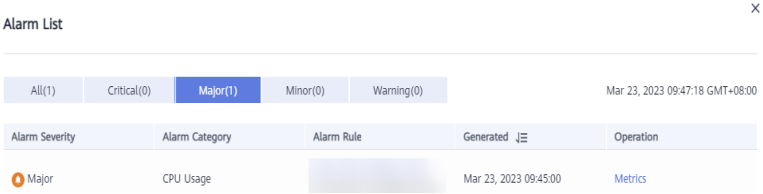
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 On the **Dashboard** page, view instance alarms provided by Cloud Eye.

You can customize alarm rules by adjusting alarm policies and severities for key metrics, such as CPU usage and disk usage. To view alarm details, click the number next to an alarm severity.

Figure 15-1 Alarms

Figure 15-2 Alarm list



----End

Health

In the **Health** area, you can view real-time health check results. By default, the data for high vCPU utilization, memory bottlenecks, high-frequency slow SQL statements, and lock waits are displayed.

For abnormal metrics, click **Diagnose** to view diagnosis details and suggestions. For details, see **Table 15-2**.

For details about metrics, see **Viewing TaurusDB Metrics**. For details about how to configure alarm rules on Cloud Eye, see **Configuring TaurusDB Alarm Rules**.

Figure 15-3 Health

Table 15-2 Health diagnosis and suggestions

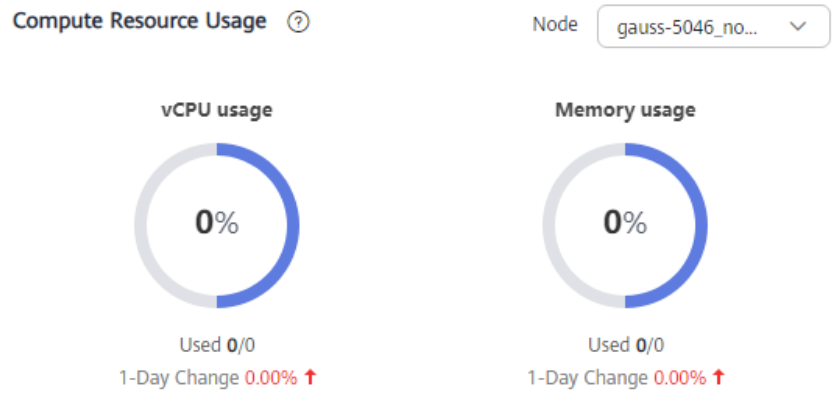
Item	Exception Trigger Condition	Suggestion
High vCPU utilization	Either of the following conditions is met: • After you configure alarm rules on Cloud Eye, an alarm is reported, indicating the CPU usage is high. • The CPU usage exceeds 95% for more than 2.5 minutes of a 5-minute measurement period.	**What Should I Do If the CPU Usage of My TaurusDB Instance Is High?**
Memory bottleneck	Either of the following conditions is met: • After you configure alarm rules on Cloud Eye, an alarm is reported, indicating the memory usage is high. • The memory usage exceeds 95% within a 5-minute measurement period.	**How Do I Handle a Large Number of Temporary Tables Being Generated for Long Transactions and High Memory Usage?**
High-frequency slow SQL	Either of the following conditions is met: • After you configure alarm rules on Cloud Eye, an alarm is reported, indicating there are too many slow logs. • There are more than 100 slow logs within five minutes.	**How Do I Handle Slow SQL Statements Caused by Inappropriate Composite Index Settings?**

Item	Exception Trigger Condition	Suggestion
Lock wait	After you configure alarm rules on Cloud Eye, any of the following alarms is reported: • Row Lock Time • InnoDB Row Locks • Row Lock Waits	What Should I Do If Locks on Long Transactions Block the Execution of Subsequent Transactions?

Compute Resource Usage

In the **Compute Resource Usage** area, the vCPU usage and memory usage are displayed by default. The displayed values are the average values for 5-minute measurement periods.

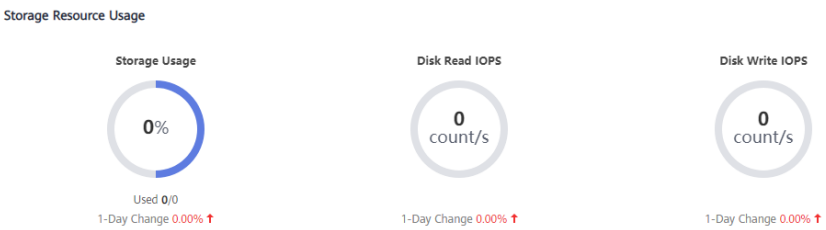
Figure 15-4 Compute Resource Usage



Storage Resource Usage

In the **Storage Resource Usage** area, the storage usage, disk read IOPS, and disk write IOPS are displayed by default. The displayed values are the average values for 5-minute measurement periods.

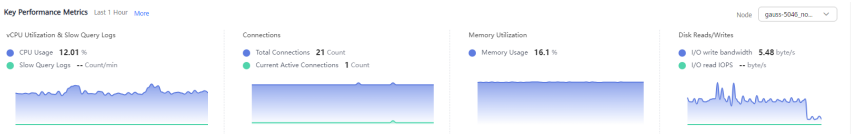
Figure 15-5 Storage Resource Usage



Key Performance Metrics

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

Figure 15-6 Key Performance Metrics



15.2.2 Viewing Real-Time Performance Metrics

TaurusDB allows you to view performance metrics and trends of DB instances in real time, helping you detect and handle potential performance problems in a timely manner.

Procedure

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

Step 2 Click 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**.

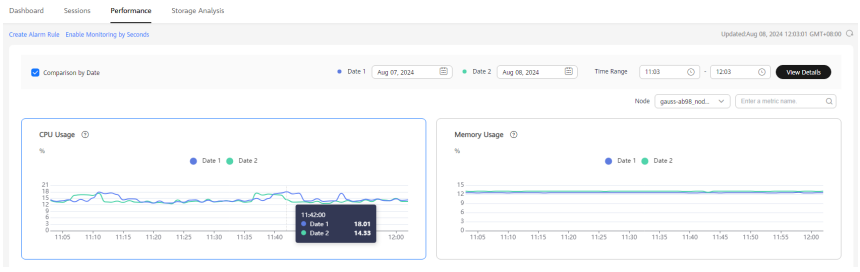
Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Performance** tab to view the performance metrics of your DB instance.

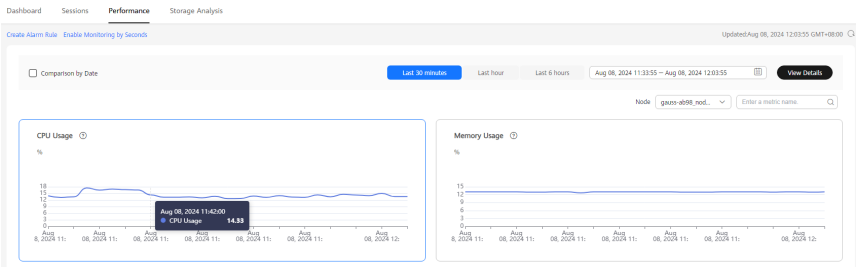
- If you select **Comparison by Date**, you can view metric trends of the DB instance in a time range on different dates. You can move the cursor to a point in time of a chart to view metric values at the point in time on different dates.

  Figure 15-7 Viewing a performance metric at a point in time on different dates

- If you deselect **Comparison by Date**, you can view performance metric trends in the last 30 minutes, last hour, last 6 hours, or a custom time range. You can move the cursor to a point in time of a chart to view the metric value at the point in time.

Figure 15-8 Viewing a performance metric trend in the last 30 minutes

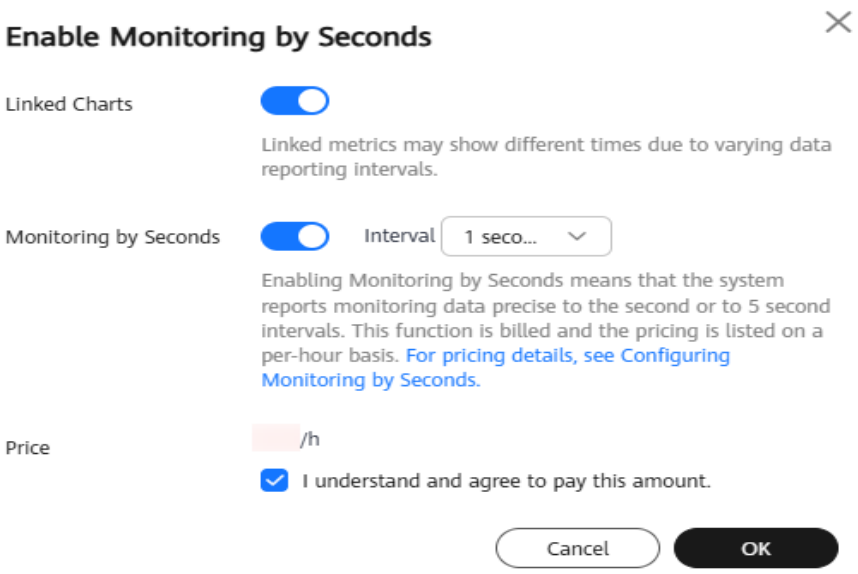


- You can also click **Create Alarm Rule** to set alarm rules for your DB instance. This will allow you to stay informed about the status of your DB instance and receive timely warnings.
- The system monitors performance data every minute by default. You can click **Enable Monitoring by Seconds** on the **Performance** tab to configure linked charts and enable monitoring by seconds.

  Linked Charts: Enabling it means that you can view all metrics at the same time.

  Monitoring by Seconds: Enabling it means that the system reports monitoring data precise to the second or to 5 second intervals. This function is billed and the pricing is listed on a per-hour basis.

Figure 15-9 Enabling monitoring by seconds



----End

15.3 Problem Diagnosis

15.3.1 Managing Real-Time Sessions

Scenarios

You can view current session statistics of your instance and kill abnormal sessions.

Constraints

Killing a session may cause the application to disconnect from the instance. Your application should be able to reconnect to the instance.

Setting a Slow Session Threshold

Step 1 Log in to the management console.

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

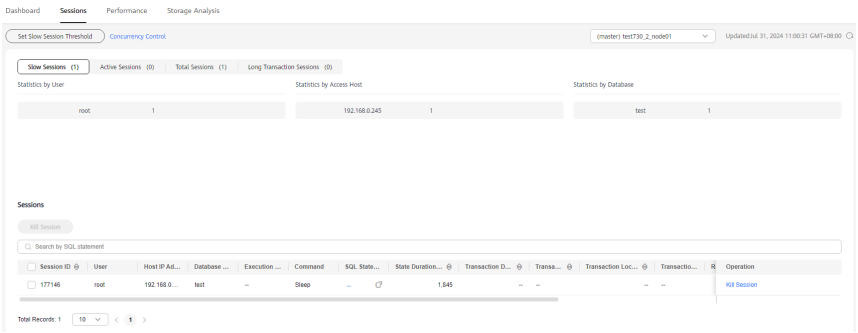
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

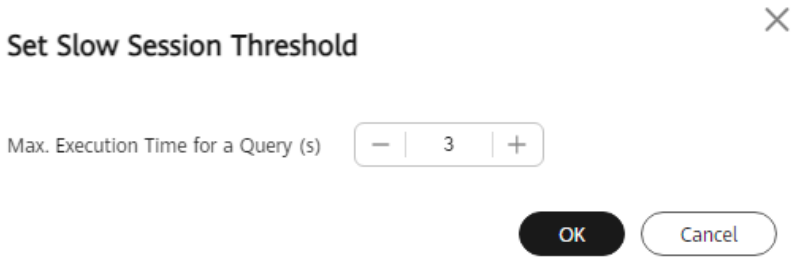
Step 6 Click the **Sessions** tab to view current session statistics by user, access host, and database.

Figure 15-10 Sessions



Step 7 Click **Set Slow Session Threshold**. In the displayed dialog box, configure **Max. Execution Time for a Query (s)** and click **OK**. Sessions whose execution time exceeds the threshold are automatically displayed. Too long SQL statements will be truncated and displayed in the session list.

Figure 15-11 Setting a slow session threshold



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.

To kill sessions automatically, see **Configuring Auto Throttling**.

----End

15.3.2 Managing Storage

Functions

Storage occupied by data and logs and changes of storage usage are important for database performance. On the **Storage Analysis** page, you view the distribution and change trend of the disk space. **Autoscaling**, **Tablespaces**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.

Table 15-3 Functions

Function	Description	Related Operation
Overview	You can view storage usage, available storage, total storage, daily increase in the last week, and estimated available days of storage.	**Viewing Storage Usage**
Tablespaces	You can view tables with abnormal tablespace growth, tables without primary keys, and tables without indexes.	**Tablespaces**
Disk Space Distribution and Used Disk Space	You can view the distribution and change trend of the disk space.	**Viewing Disk Space Distribution**
Top Databases and Tables	You can view the top 50 databases and tables by physical file size and identify the high-usage databases and tables based on disk space distribution.	**Top Databases and Tables**

Viewing Storage Usage

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

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

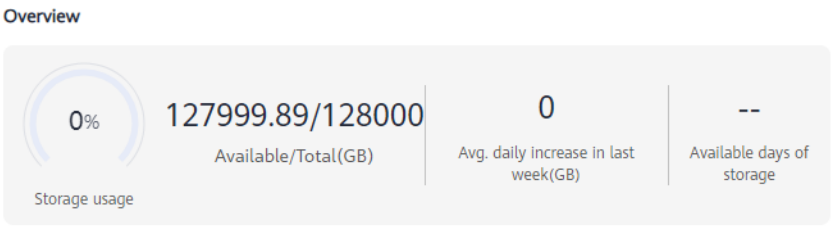
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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.

Figure 15-12 Viewing the storage overview



The following information is displayed:

- Storage usage
- Available and total storage
- Average daily increase in the last week
- Available days of storage

  NOTE

  If the average daily increase in last week is 0 GB, the estimated available days of storage are unlimited and are not displayed.

----End

Tablespaces

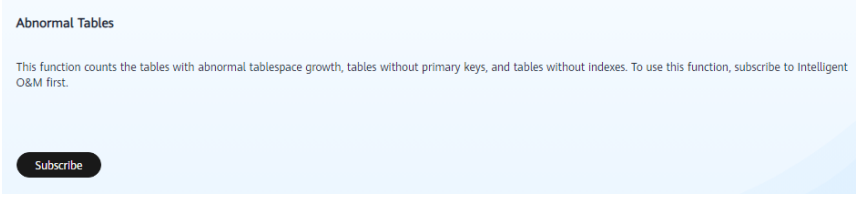
You can view tables with abnormal tablespace growth, tables without primary keys, and tables without indexes through automated or manual diagnosis.

CAUTION

If there are more than 5,000 tables or the vCPU usage exceeds 90%, manual diagnosis is not supported.

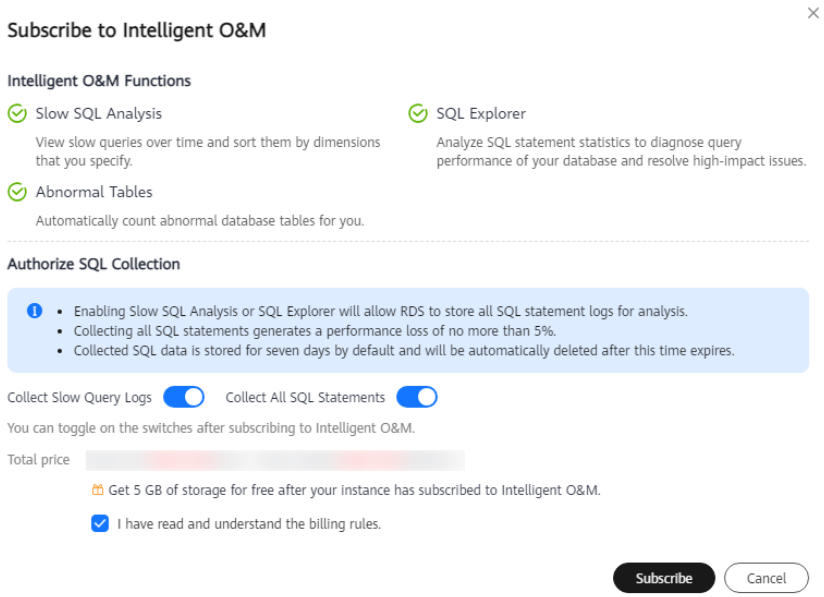
Step 1 In the **Abnormal Tables** area, click **Subscribe**.

Figure 15-13 Abnormal Tables



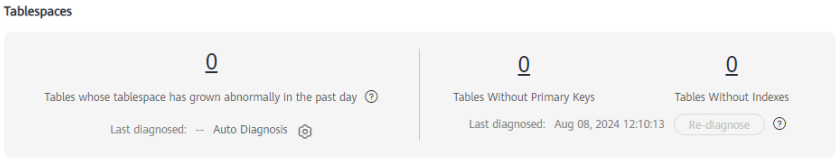
Step 2 In the **Subscribe to Intelligent O&M** dialog box, confirm the information, select the agreement, and click **Subscribe**.

Figure 15-14 Subscribing to Intelligent O&M



Step 3 In the **Tablespaces** area, view table diagnosis results.

Figure 15-15 Viewing table diagnosis results



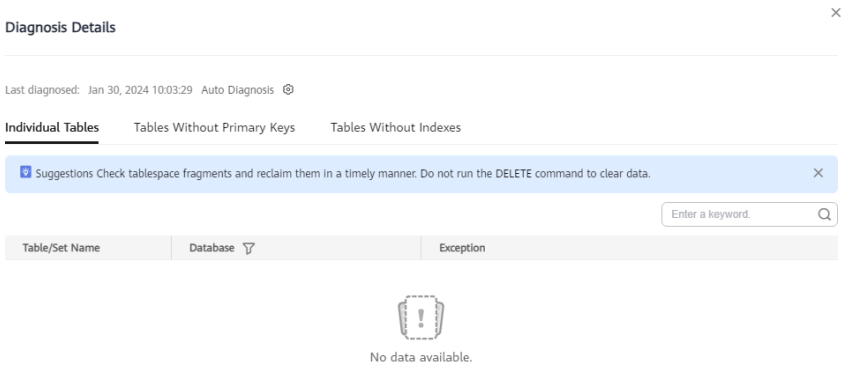
Both automated diagnosis and manual diagnosis are supported.

- Automated diagnosis

  Tables in the **Top 50 Tables** area are automatically diagnosed at about 04:00 every day.

  In the left part of the **Tablespaces** area, you can view tables whose tablespace has grown abnormally in the past day. You can click the number to view the diagnosis details and handle the abnormal tables based on the suggestions provided.

Figure 15-16 Viewing diagnosis details



Any table whose tablespace has grown by more than 10,240 MB in the past day is counted. You can also click on the right of **Auto Diagnosis** to set the upper limit for daily tablespace increase.

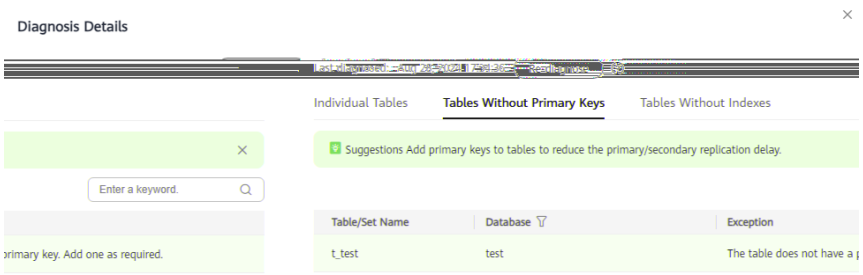
Figure 15-17 Setting the upper limit

- Manual diagnosis

  Click **Re-diagnose** to manually trigger a diagnosis task. This operation can be performed every 10 minutes. The diagnosis scope is not limited.

  Once the diagnosis is complete, you can view the numbers of tables without primary keys and tables without indexes. You can click a number to view the diagnosis details and handle the abnormal tables based on the suggestions provided.

Figure 15-18 Viewing diagnosis details

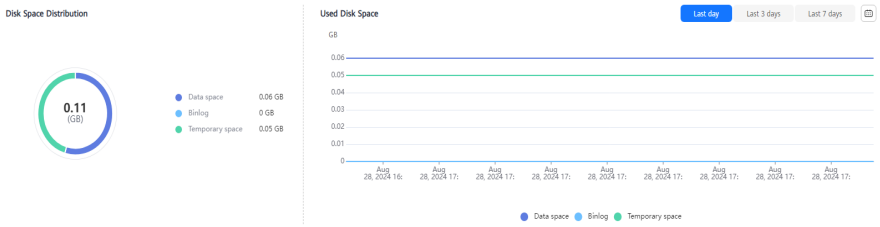


----End

Viewing Disk Space Distribution

You can view the distribution and change trend of the disk space.

Figure 15-19 Viewing disk space distribution



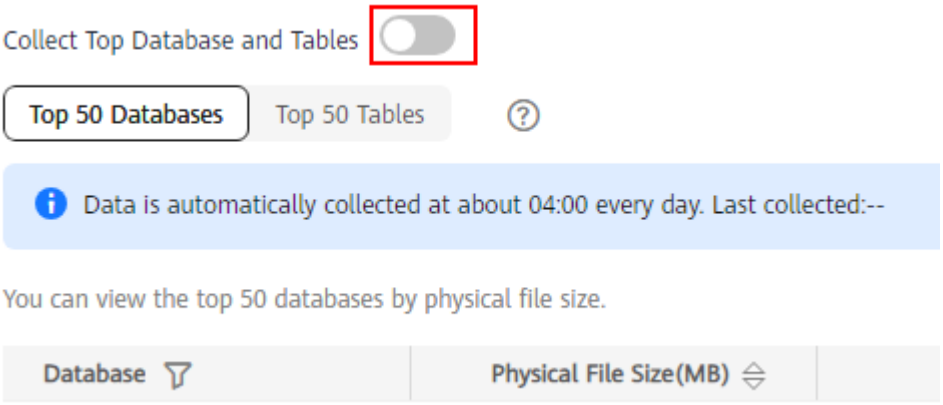
- **Data space**: Disk space occupied by user data
- **Binlog**: Disk space occupied by binlogs
- **Temporary space**: Disk space occupied by temporary files

Top Databases and Tables

Step 1 Click  on the right of **Collect Top Databases and Tables** to enable the function.

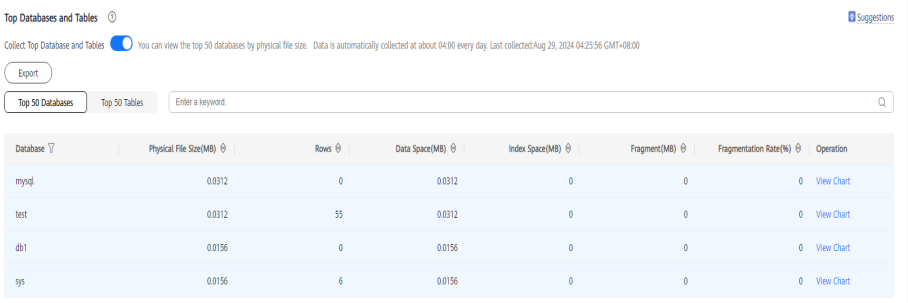
The system automatically collects data of top 50 databases and tables at about 04:00 every day.

Figure 15-20 Enabling Collect Top Databases and Tables



Step 2 View the top 50 databases and tables by physical file size and identify the high-usage databases and tables based on disk space distribution.

Figure 15-21 Viewing top 50 databases and tables



- 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.

Step 3 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

15.3.3 Viewing Anomaly Snapshots

After anomaly diagnosis is enabled, the system checks your instance health status and diagnoses faults. If there is an anomaly, its snapshots will be collected, helping you monitor instance performance in real time.

Diagnosis Item

Table 15-4 Diagnosis item

Item	Description
Transaction uncommitted	There are uncommitted transactions.

Constraints

Enabling anomaly collection will cause about 5% of instance performance loss.

Procedure

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

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

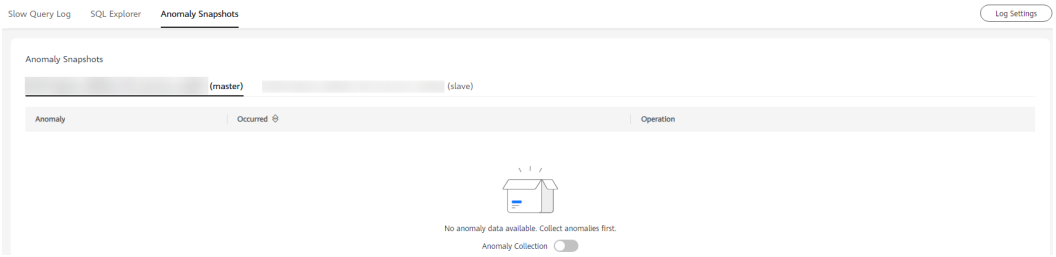
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

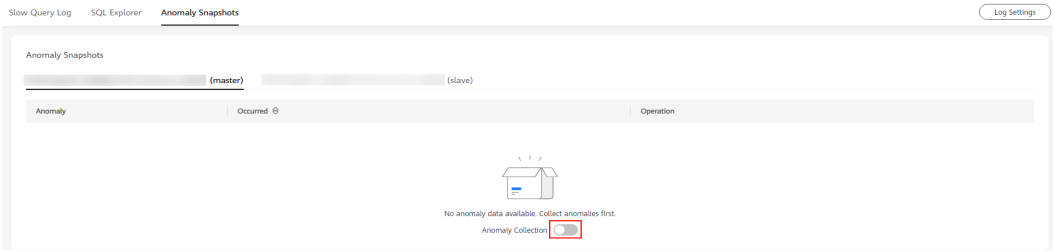
Step 6 Click **Anomaly Snapshots**.

Figure 15-22 Anomaly Snapshots page



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

Figure 15-23 Enabling anomaly diagnosis



After anomaly diagnosis is enabled, if any anomaly listed in **Table 15-4** 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

15.3.4 Managing Locks and Transactions

Functions

Metadata Locks

- Metadata locks (MDLs) are used to ensure consistency between DDL and DML operations. Executing DDL statements on a table generates metadata

write locks. If there is a metadata lock, all subsequent SELECT, DML, and DDL operations on the table will be blocked, causing a connection backlog.

- Metadata locks are displayed in real time. You can quickly identify locking problems and terminate the sessions holding metadata locks to restore blocked operations.
- DML locks are not included on this page. You can view and analyze them on the **InnoDB Locks** page.
- A maximum of 1,000 records can be displayed.

InnoDB Locks

- InnoDB lock waits generated before DML operations are displayed in real time. You can quickly locate the session waits and blocks that happened when multiple sessions update the same piece of data at the same time, and can terminate the source sessions that hold locks to restore blocked operations.
- DDL locks are not included on this page. You can view and analyze them on the **Metadata Locks** page.
- Lock information can be viewed only when **Performance Schema** is enabled. To check the **Performance Schema** status, run **SHOW GLOBAL VARIABLES LIKE "performance_schema"** or go to the **Parameters** page of TaurusDB.

Deadlock Analysis

- This function analyzes the latest deadlock log returned by **SHOW ENGINE INNODB STATUS**. If there have been multiple deadlocks, only the latest deadlock is analyzed.
- You can query lock analysis data of the past seven days.

Full Deadlock Analysis

- The kernel version of your TaurusDB instance must be 2.0.45.230900 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- After this function is enabled, the system regularly examines error logs, extracts deadlock details from them, and conducts a full deadlock analysis.
- The following parameters must be enabled:

  innodb_print_all_deadlocks

  innodb_deadlock_detect (This parameter is enabled by default.)
- A maximum of 10,000 records can be displayed.
- You can query lock analysis data of the past seven days.

Procedure

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

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

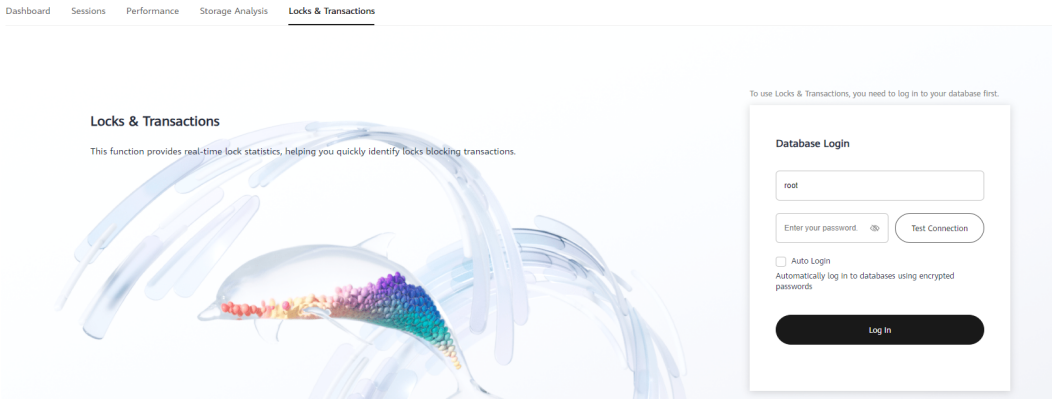
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

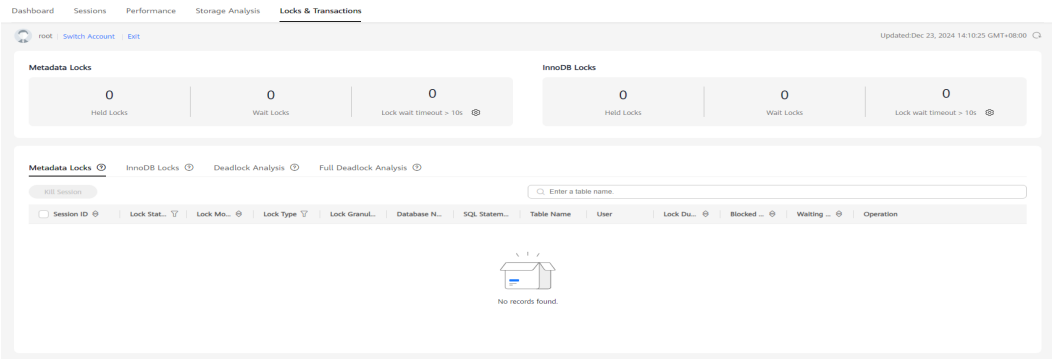
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Locks & Transactions** tab and enter the administrator password to log in to the instance.

Figure 15-24 Logging in to an instance



Step 7 On the **Metadata Locks** page, filter the desired metadata locks by session ID, lock status, lock type, and database name.

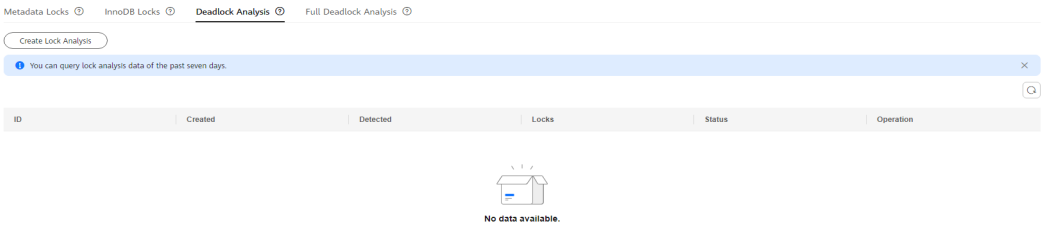


Step 8 Check whether there are any sessions with metadata locks.

If so, select the sessions and click **Kill Session**.

Step 9 On the **InnoDB Locks** page, check whether there are any lock waits.

Step 10 On the **Deadlock Analysis** page, view the latest lock analysis data. You can click **Create Lock Analysis** to create a lock analysis data record.



Step 11 Enable **Full Deadlock Analysis** on the **Full Deadlock Analysis** page and set the **innodb_print_all_deadlocks** parameter to **ON** to view the full deadlock analysis data.

----End

FAQs

How Do I View Deadlock Logs of TaurusDB?

15.4 SQL Analysis and Tunning

15.4.1 Viewing 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

- If you did not subscribe to Intelligent O&M, you can only view the data of the last hour. The data will be automatically deleted when it expires. After you subscribe to Intelligent O&M, 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 will be stored in OBS.
- Incorrectly optimizing slow queries may cause service exceptions. Exercise caution when performing this operation.

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.

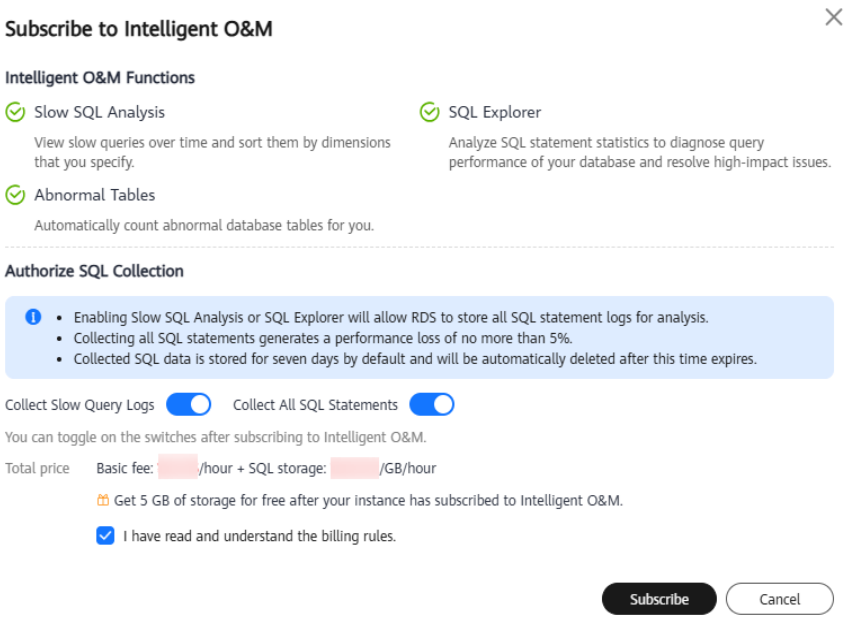
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

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.

Your instance will get 5 GB of storage for free after it has subscribed to Intelligent O&M.

Figure 15-25 Subscribing to Intelligent O&M



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

----End

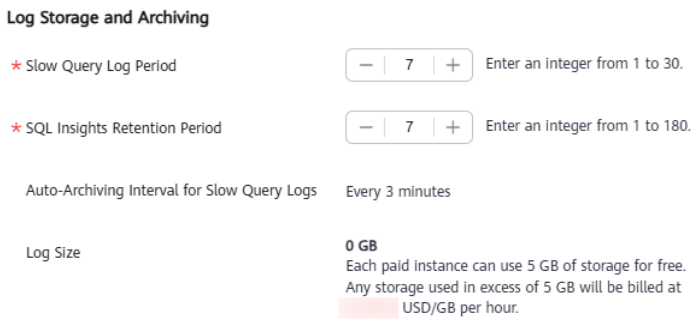
Slow Query Log Storage

- Intelligent O&M subscribed:

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

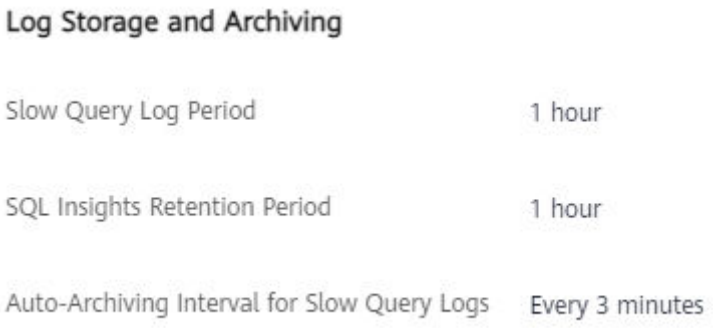
  - **Slow Query Log Period**: The default value is **7**. The value ranges from 1 to 30. After the period expires, the logs are automatically deleted.
  - **SQL Insights Retention Period**: The default value is **7**. The value ranges from 1 to 180.
  - **Log Size**: Each paid instance can use 5 GB of storage for slow query logs for free. Any storage used in excess of 5 GB will be billed on a pay-per-use basis.

Figure 15-26 Log storage and archiving



- 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

Figure 15-27 Log storage and archiving



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**.

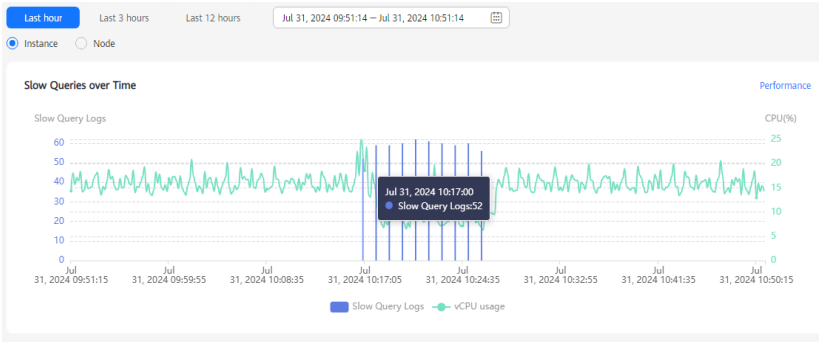
Step 6 Click the **Slow Query Logs** tab.

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.

You can move the cursor to a point in time of the chart to view the number of slow query logs and CPU usage at the point in time.

Figure 15-28 Viewing slow queries over time



----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.

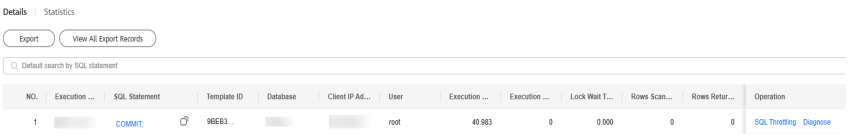
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Slow Query Logs** tab.

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

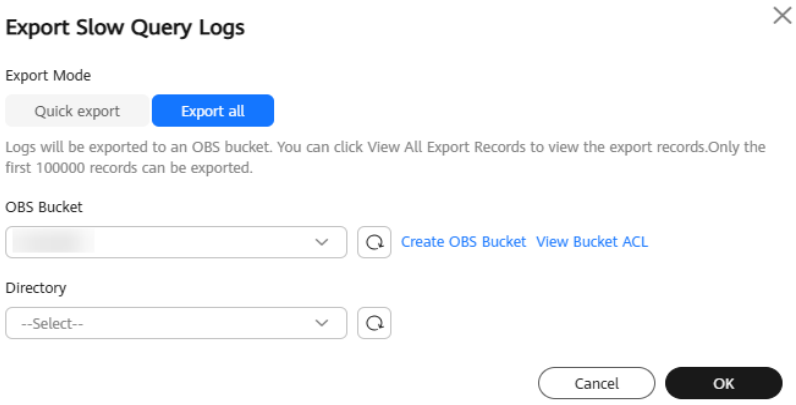
You can view slow query logs in the last 1 hour, 3 hours, 12 hours, or a custom time period.

Figure 15-29 Viewing slow query log details



- Export slow query log details.
  a. Click **Export**.
  b. In the displayed dialog box, select **Quick export** or **Export all** for **Export Mode**.

Figure 15-30 Exporting slow query log details



- **Quick export**: Log details are exported to your local PC. Only the first 1,000 records can be exported.
- **Export all**: Select an OBS bucket and a directory to export log details to the OBS bucket. 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**.

  Creating an OBS bucket is free, but you will be billed for storing data in the bucket. For pricing details, see **OBS Pricing Details**.

  A bucket name:

  - Cannot be the same as that of any existing bucket.
  - Can contain 3 to 63 characters. Only lowercase letters, digits, 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.

c. Click **OK**.

d. After the slow query log details are exported, click **View All Export Records**. In the displayed dialog box, click **Download** in the **Operation** column. After the download is complete, view the slow query log details in the ZIP package.

- To add a SQL throttling rule, locate a record and click **SQL Throttling** in the **Operation** column. In the displayed dialog box, specify **SQL Type**, **Keyword**, and **Max. Concurrent Requests**, and click **OK**. For details, see **Configuring SQL Throttling**.
- To view SQL diagnosis results, locate a record and click **Diagnose** in the **Operation** column. In the displayed dialog box, confirm the slow SQL statement to be diagnosed and click **OK**. In the **Diagnosis Details** area, view the SQL diagnosis results.

Figure 15-31 Diagnosing a SQL statement

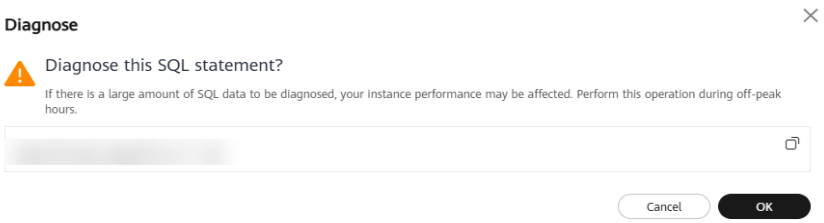


Figure 15-32 Viewing diagnosis details

----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.

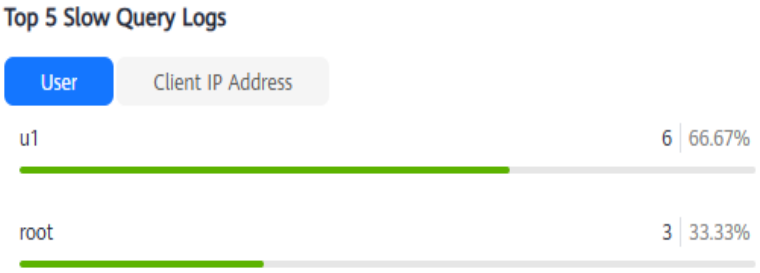
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

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 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.

Figure 15-33 Viewing top 5 slow query logs



----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**.

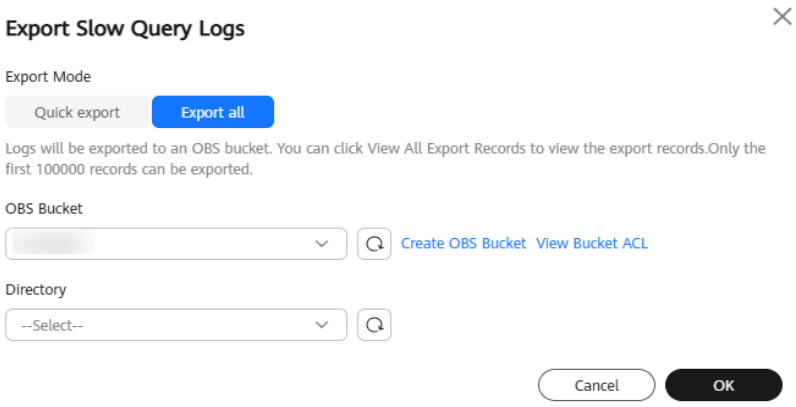
Step 6 Click the **Slow Query Logs** tab.

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

Figure 15-34 Viewing template statistics

- Click **View Sample** in the **Operation** to view the sample of a SQL template.
- Export template statistics.
  a. Click **Export**.
  b. In the displayed dialog box, select **Quick export** or **Export all** for **Export Mode**.

Figure 15-35 Exporting slow query log details



- **Quick export**: Log details are exported to your local PC. Only the first 1,000 records can be exported.
- **Export all**: Select an OBS bucket and a directory to export log details to the OBS bucket. 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**.

  Creating an OBS bucket is free, but you will be billed for storing data in the bucket. For pricing details, see **OBS Pricing Details**.

  A bucket name:

  - Cannot be the same as that of any existing bucket.
  - Can contain 3 to 63 characters. Only lowercase letters, digits, 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.

c. Click **OK**.

d. After the slow query log details are exported, click **View All Export Records**. In the displayed dialog box, click **Download** in the **Operation** column. After the download is complete, view the slow query log details in the ZIP package.

----End

15.4.2 Viewing Top SQL Statements

Scenarios

After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.

Constraints

If Intelligent O&M is not subscribed to, you can only view data in the last hour by default. The data will be automatically deleted when it expires. If Intelligent O&M is subscribed, you can configure how long that top SQL statements are stored for (at most one day).

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

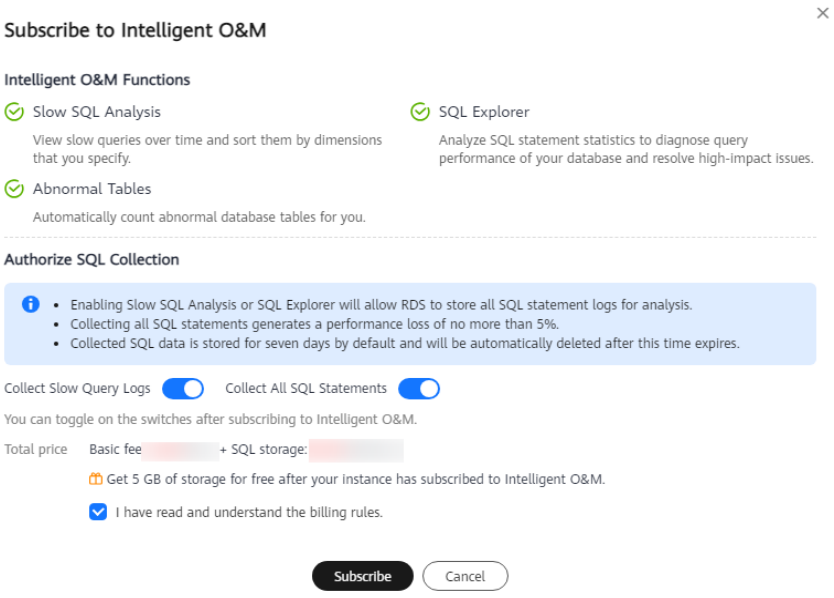
Step 4 On the **Instances** page, click the instance name.

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

Step 6 Choose **SQL Explorer** > **Top SQL**.

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

Figure 15-36 Subscribing to Intelligent O&M



Step 8 View the top SQL statements of the DB instance.

- View execution durations of the top SQL statements in the last 1 hour, last 3 hours, last 12 hours, or a custom time period (no longer than one day).
- Click a point in time or drag to select a time period to view the SQL statistics of 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 **SQL Throttling** in the **Operation** column to add a SQL throttling rule. For details, see **Configuring SQL Throttling**.
- Select **Comparison by Date** and select dates and a time range to view the top SQL statements in the time range on different days.

----End

15.4.3 Creating a SQL Insights Task

Scenarios

SQL Insights allows you to not only query all executed SQL statements, but also analyze and search for the tables that are accessed and updated most frequently, and the SQL statements that have the longest lock wait, helping you quickly identify exceptions.

Constraints

- You need to enable **Collect All SQL Statements** before using SQL Insights. Collecting all SQL statements generates a performance loss of no more than 5%.
- 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 DB Instance**.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

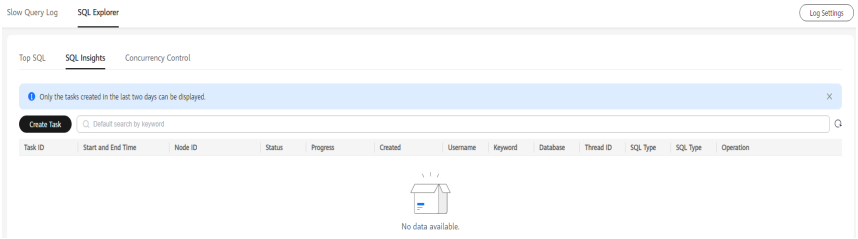
Step 6 Under the **SQL Explorer** tab, click **SQL Insights**.

Click next to **Collect All SQL Statements**.

To disable this function, click **Log Settings** in the upper right corner, toggle off the **Collect All SQL Statements** switch, and click **OK**.

Step 7 Click **Create Task**.

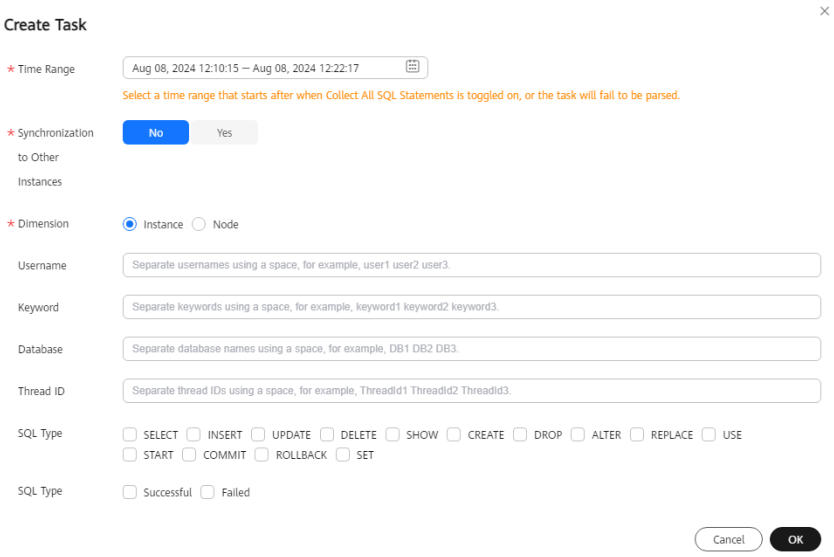
Figure 15-37 Creating a SQL insights task



Step 8 On the displayed page, set **Time Range**, **Synchronization to Other Instances**, **Dimension**, **Username**, **Keyword**, **Database**, **Thread ID**, **SQL Type**, and **Status**.

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

Figure 15-38 Creating a SQL insights task



Step 9 Click **OK**.

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

Step 11 Select a keyword such as **Time Range**, **Username**, **Keyword**, or **Database** to search for the SQL statements executed on the current instance or node.

The selected time range must be after the time when the new task is added.

----End

15.4.4 Configuring SQL Throttling

Scenarios

SQL throttling 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. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
  - 2.0.28.40 > kernel version ≥ 2.0.28.15
  - Kernel version ≥ 2.0.29.1
- SQL throttling rules for INSERT statements are only supported when the kernel version of your TaurusDB instance is 2.0.54.240600 or later.
- A single throttling rule can contain a maximum of 128 keywords.
- Single backslashes (\) or single null characters (' ') cannot be used as keywords.
- Spaces and escape character \'t', \'r', and \'n' at the start and end of a keyword are ignored.
- A throttling rule cannot end with a tilde (~).
- Keywords in a throttling rule will be matched in sequence. For example, if a rule contains **a~and~b**, only **xxx a>1 and b>2** will be matched. **xxx b>2 and a>1** will not be matched.
- Each SQL throttling rule is applied 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 an instance, the rules still apply to the primary node and read replicas after their roles are switched over.
- If a SQL statement matches multiple throttling rules, only the most recently added rule is applied.
- SQL statements that have been executed before a SQL throttling rule is added are not counted.
- If the kernel version of your TaurusDB instance is 2.0.54.240600 or later, the total length of all rules and concurrent requests of SELECT, UPDATE, DELETE, or INSERT statements cannot exceed 4,000 bytes. The length of a single rule cannot exceed 1,000 bytes.
- If the kernel version of your TaurusDB instance is earlier than 2.0.54.240600, the total length of all rules and concurrent requests of SELECT, UPDATE, or DELETE statements cannot exceed 1,024 bytes.
- If there are too many throttling rules, performance of SELECT, UPDATE, DELETE, and INSERT statements will deteriorate.
- Throttling rules are applied based on SQL statement prefixes. For example, if a throttling rule is **SELECT~COUNT~t1**, **SELECT COUNT(*) FROM t1** and **SELECT COUNT(*) FROM t1 LIMIT 1** will both be intercepted.
- If SQL throttling is triggered, error "ERROR 1317 (70100): Query execution was interrupted" is reported.

- This function controls how many statements can run at the same time. However, it does not limit concurrency for:
  - System tables
  - SQL statements not used to query data, such as **select sleep(xxx);**
  - Account **root**
  - SQL statements in stored procedures, triggers, and functions

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

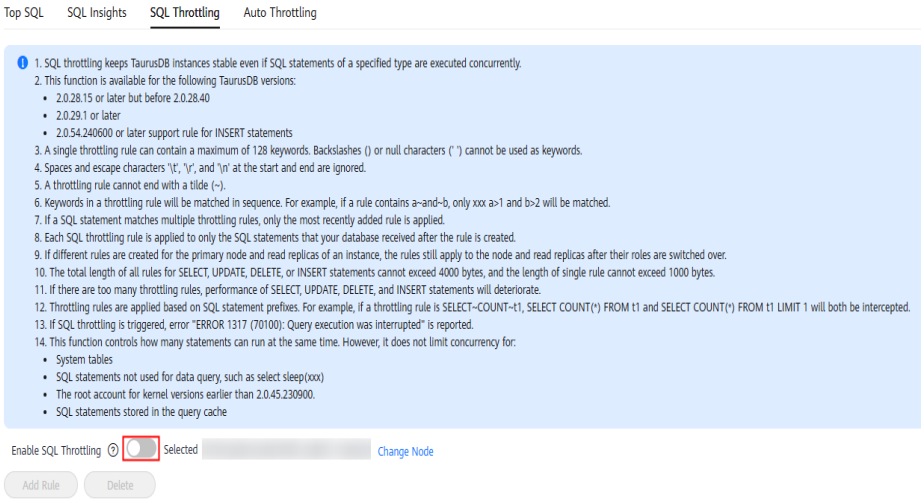
Step 4 On the **Instances** page, click the instance name.

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

Step 6 Choose **SQL Explorer** > **SQL Throttling**.

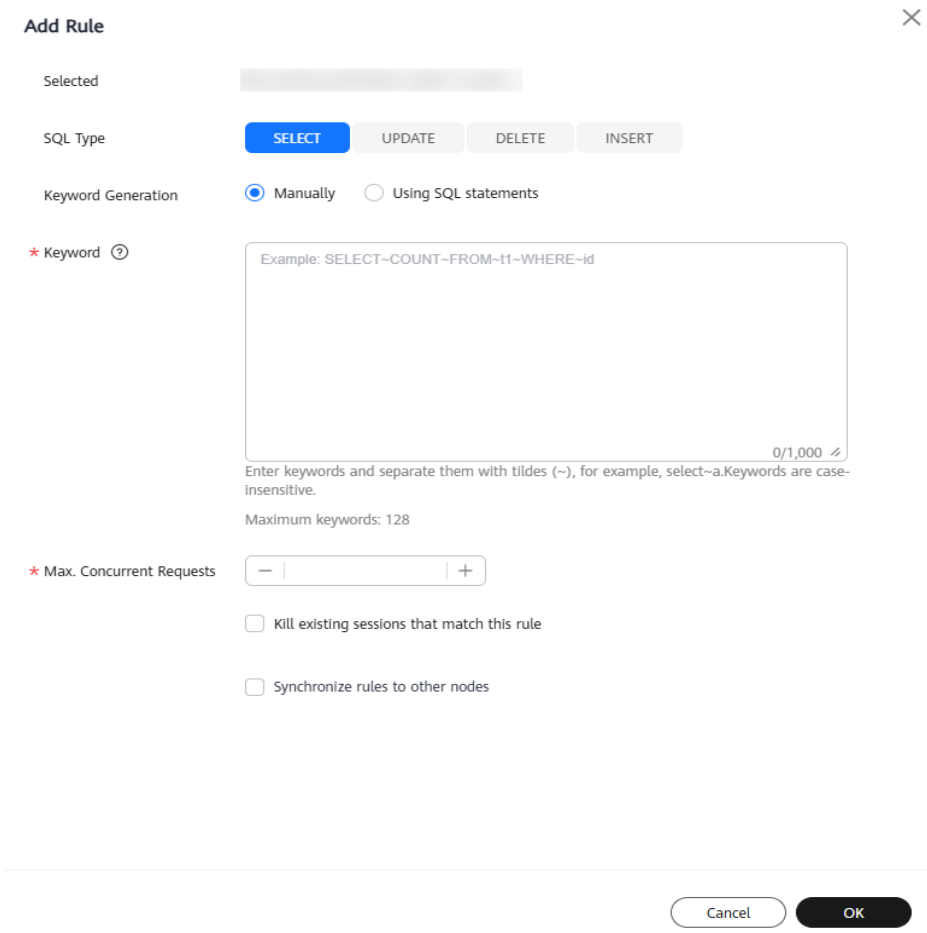
Step 7 On the displayed page, toggle on **Enable SQL Throttling**.

Figure 15-39 Enabling SQL throttling



Step 8 Click **Add Rule**. In the displayed dialog box, specify **SQL Type**, **Keyword**, and **Max. Concurrent Requests**.

Figure 15-40 Adding a SQL throttling rule



- **Keyword**: You can enter keywords or copy an existing SQL statement to the text box and click **Generate Keyword**.

  If you enter **select~a**, the throttling rule is applied to SQL statements containing **select** and **a**.
- **Max. Concurrent Requests**: SQL statements that meet the specified SQL type and keyword and exceed the value of **Max. Concurrent Requests** will not be executed.
- If you select **Kill existing sessions that match this rule**, the sessions that match 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 SQL throttling rule is not required, select the rule and click **Delete** above the rule list. In the displayed dialog box, click **OK**.

----End

15.4.5 Configuring Auto Throttling

Auto throttling allows you to kill all sessions, kill specific sessions by criteria, and view history.

To kill the current session or manually kill a session, see **Managing Real-Time Sessions**.

Functions

Function	Description
Killing specific sessions by criteria	You can add a task for killing sessions. Sessions that meet the criteria will be killed.
Killing all sessions	After you enable **Auto Kill Sessions** and click **Kill All Sessions**, all sessions are automatically deleted.
Viewing history	You can view killed sessions.

Killing Specific Sessions by Criteria

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

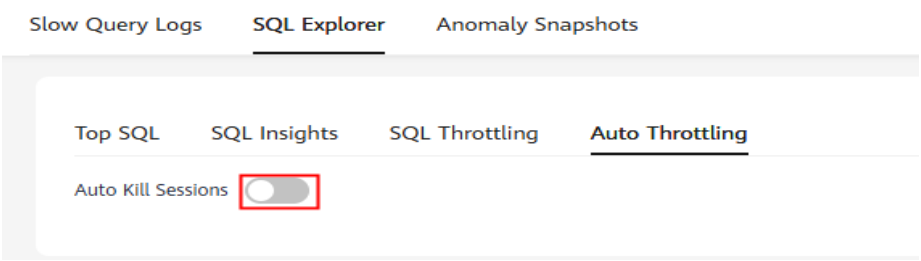
Step 4 On the **Instances** page, click the instance name.

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

Step 6 Click **SQL Explorer** and then **Auto Throttling**.

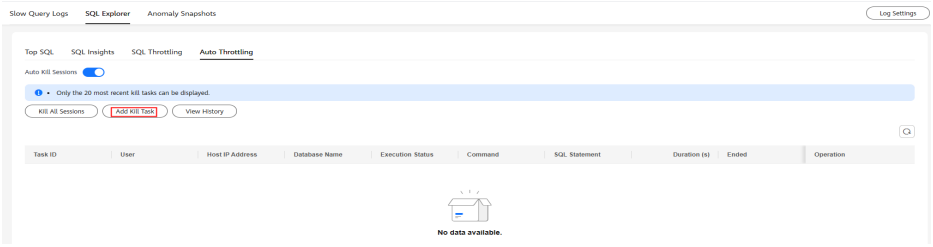
Step 7 Click on the right of **Auto Kill Sessions**. In the displayed dialog box, click **OK**.

Figure 15-41 Enabling Auto Kill Sessions



Step 8 Click **Add Kill Task**.

Figure 15-42 Adding a task for killing sessions



Step 9 In the displayed dialog box, set the criteria for killing sessions. The parameters listed in **Table 15-5** are in a logical AND relationship.

If you only specify **Session Duration (s)** and **Task Duration (s)**, all sessions that meet the criteria will be killed.

A maximum of five conditional kill tasks can be executed at the same time.

Figure 15-43 Setting the criteria for killing sessions

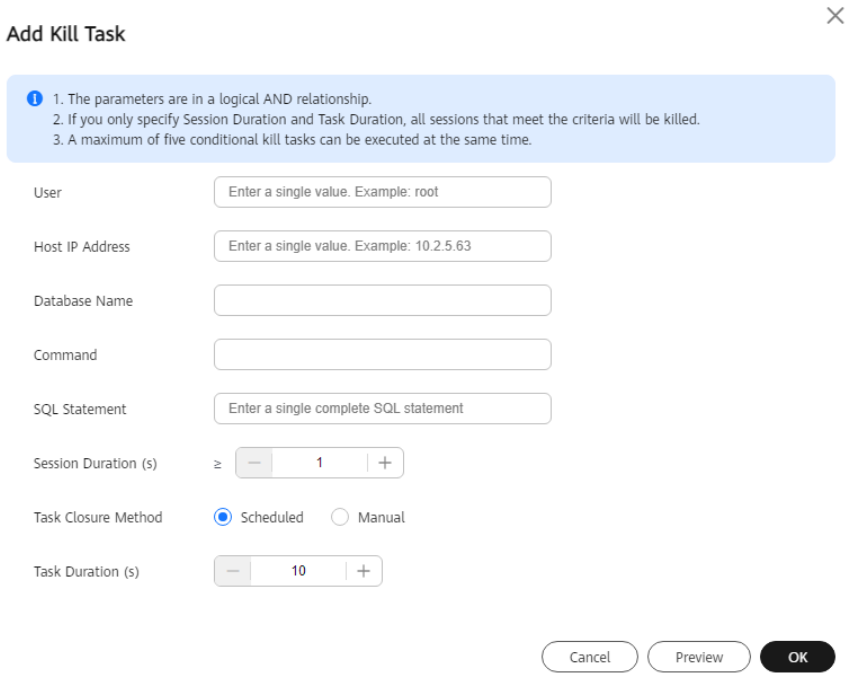


Table 15-5 Parameter description

Parameter	Description
User	Enter a single value, for example, **root**.
Host IP Address	Enter a single value, for example, **168.192.0.0**.
Database Name	Enter a database name.
Command	Enter a command.

Parameter	Description
SQL Statement	Enter a 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

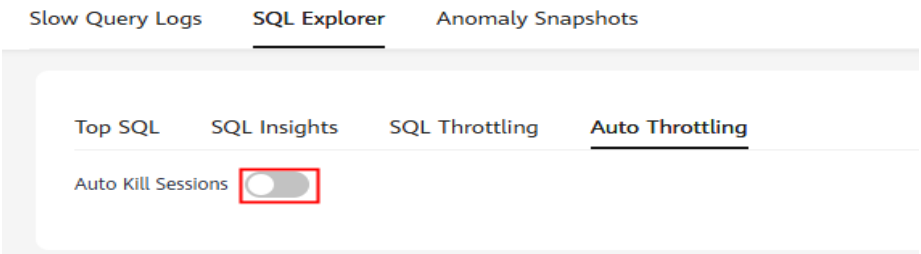
Step 1 On the **Instances** page, click the instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 3 Click **SQL Explorer** and then **Auto Throttling**.

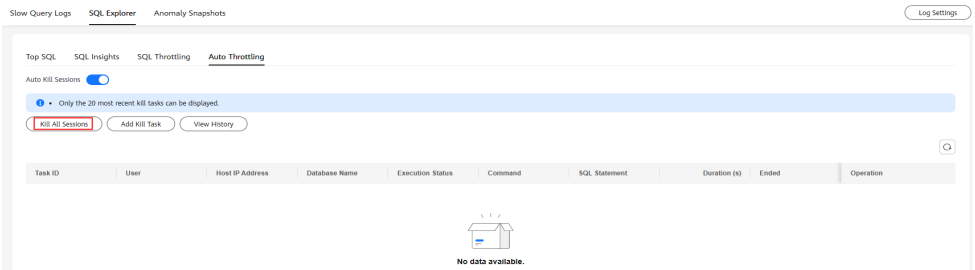
Step 4 Click on the right of **Auto Kill Sessions**. In the displayed dialog box, click **OK**.

Figure 15-44 Enabling Auto Kill Sessions



Step 5 Click **Kill All Sessions**.

Figure 15-45 Killing all sessions



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

----End

Viewing History

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

Step 2 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 3 Click **SQL Explorer** and then **Auto Throttling**.

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

16 Security and Encryption

16.1 Configuring Database Security

Password Strength Requirements

For database password strength requirements on the TaurusDB console, see the database configuration table in **Buying a DB Instance**.

TaurusDB has a password security policy for newly created database users. Passwords must:

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

When you create instances, your password strength is checked. You can modify the password strength as user **root**. For security reasons, use a password that is at least as strong as the default one.

Account Description

To provide O&M services, the system automatically creates system accounts when you create TaurusDB instances. These system accounts are unavailable to you.

WARNING

Attempting to delete, rename, or change passwords or permissions for these accounts will result in an error. Exercise caution when performing these operations.

- rdsAdmin: a management account with superuser permissions, which is used to query and modify instance information, rectify faults, migrate data, and restore data.
- rdsRepl: a replication account, which is used to synchronize data from the primary node to read replicas.

- rdsBackup: a backup account, which is used to back up data in the background.
- rdsMetric: a metric monitoring account, which is used by watchdog to collect database status data.
- rdsProxy: a database proxy account, which is used for authentication when the database is connected through the proxy address. This account is automatically created when you enable read/write splitting.

16.2 Resetting the Administrator Password

Scenarios

If you forget the password of your database account when using TaurusDB, you can reset the password.

If an error occurs on the **root** account, for example, if your **root** account credentials are lost or deleted, you can restore the **root** account permissions through resetting the password.

You cannot reset the administrator password under the following circumstances:

- Your account is frozen.
- The database port is being changed.
- The instance status is **Creating**, **Restoring**, **Rebooting**, **Changing port**, **Changing instance specifications**, **Promoting to primary**, or **Abnormal**.

Precautions

- If you have changed the administrator password of a DB instance, the passwords of the read replicas associated with the instance will also be changed accordingly.
- The time it takes for the new password to take effect depends on the amount of service data currently being processed by the primary node.
- To protect against brute force hacking and improve system security, change your password periodically, such as every three or six months.
- The instance may have been restored from a backup before you reset the administrator password.
- If you have logged in to your instance as the **root** user, resetting the password may interrupt services. 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 and choose **Databases** > **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, reset the password using either of the following methods:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click **Reset Password**.
- On the **Instances** page, click the instance name to go to the **Basic Information** page. Expand **Instance Information**. In the **Configuration** area, click **Reset Password** under **Administrator**.

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

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

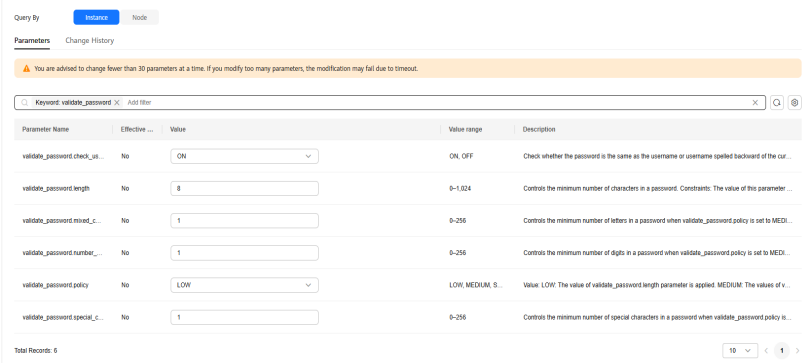
Step 6 In the displayed dialog box, enter and confirm the new password.

The new password must:

- Consist of 8 to 32 characters.
- Contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.).
- Comply with the values of **validate_password** parameters.

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

  Figure 16-1 Checking the password-related parameters



Step 7 Click **OK**.

Keep this password secure. The system cannot retrieve it.

----End

16.3 Changing the Security Group of a DB Instance

Scenarios

You can change the security group associated with your DB instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Instance Information** area, click **Expand** to expand all instance information.

Step 6 In the **Network Information** area, click **Modify** under **Security Group**.

Step 7 In the displayed dialog box, select a new security group and click **OK**.

----End

APIs

Changing a Security Group

16.4 Configuring SSL for a DB Instance

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

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

By default, SSL is enabled for new DB instances. Enabling SSL increases the network connection response time and CPU usage, and you are advised to evaluate the impact on service performance before enabling SSL.

You can use a client to connect to your DB instance through a non-SSL or SSL connection.

- If SSL is enabled for your DB instance, you can connect to your DB instance using SSL, which is more secure.
- If SSL is disabled, you can only connect to your DB instance using a non-SSL connection.

Constraints

Enabling or disabling SSL will reboot the instance immediately. During the reboot, the instance is unavailable. Rebooting an instance will clear the cached memory in it. You are advised to reboot it during off-peak hours.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the **Instance Information** area, click **Expand** to expand all instance information.

Step 6 In the **Configuration** area, click  under **SSL**.

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

Step 8 Wait for some seconds and check that SSL has been enabled on the **Basic Information** page.

To disable SSL, click  . In the displayed dialog box, click **OK**.

----End

APIs

Enabling or Disabling SSL

16.5 Enabling TDE for a DB Instance

Transparent Data Encryption (TDE) performs real-time I/O encryption and decryption on data files. Data is encrypted before being written to disks and is decrypted when being read from disks to memory. This effectively protects the security of databases and data files.

Constraints

- To enable TDE, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- To configure TDE, you must have the iam:agencies:createServiceLinkedAgencyV5 permission. If you do not have this permission, **create a custom policy**.
- You need to enable Key Management Service (KMS) for your DB instance first. The data keys used for encryption are generated and managed by KMS. 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. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- TDE can only be enabled for single-node and cluster DB instances.
- TDE can only be enabled when a DB instance is created. After the instance is created, TDE cannot be enabled or disabled.

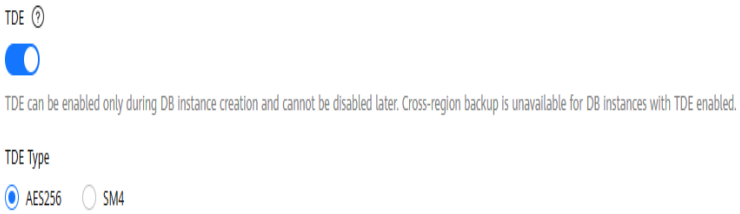
- TDE encrypts instance data, including full backups but excluding incremental backups.
- After TDE is enabled, the cryptographic algorithm cannot be changed later.
- Only instance-level encryption is supported.
- After TDE is enabled for a DB instance, you cannot:
  - Enable cross-region backup for the DB instance.
  - Restore the data of the DB instance to an existing DB instance.

Procedure

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

Step 2 On the displayed **Custom Config** page, toggle on **TDE** and select a cryptographic algorithm.

Figure 16-2 Enabling TDE



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

----End

17 Parameter Management

17.1 Viewing Parameters of a DB Instance

You can view the parameter settings of your DB instance on the console or through the CLI.

Viewing Parameters of a DB Instance on the Console

NOTE

You can only view the parameters in the parameter list on the console. To view all parameters of a DB instance, see **Viewing Parameters of a DB Instance Through the CLI**.

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

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

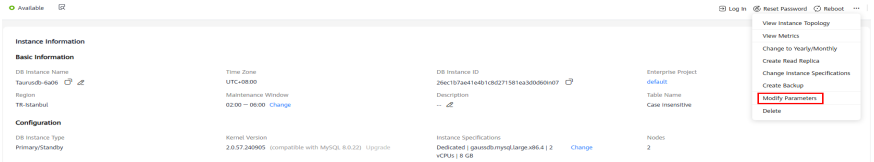
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

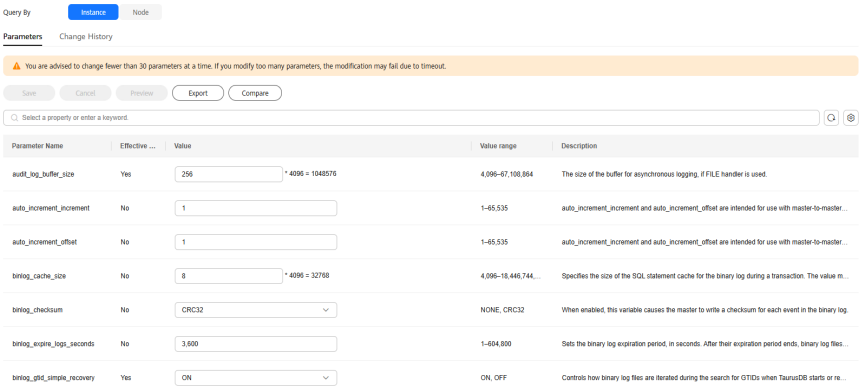
Alternatively, click in the upper right corner of the **Basic Information** page and choose **Modify Parameters**.

Figure 17-1 Choosing **Modify Parameters**



Step 6 On the **Parameters** tab, set **Query By** to **Instance** (default value) or **Node** to view the parameter settings of the current instance or node.

Figure 17-2 Viewing parameters of a DB instance



You can search for the desired parameter by parameter name.

----End

Viewing Parameters of a DB Instance Through the CLI

Step 1 Connect to a DB instance.

For details about the connection methods, see **Connection Methods**.

Step 2 Run the following command to view all parameter settings of the DB instance:

SHOW VARIABLES;

Run the following command to view the setting of a specified parameter:

SHOW VARIABLES LIKE '*<parameter_name>***';**

NOTE

A percent sign (%) can appear anywhere in *<parameter_name>* for fuzzy search. Examples:

- Querying all parameters that start with **binlog**:
  SHOW VARIABLES LIKE 'binlog%';
- Querying all parameters that end with **binlog**:
  SHOW VARIABLES LIKE '%binlog';
- Querying all parameters that start with **thread** and end with **size**:
  SHOW VARIABLES LIKE 'thread%size';
- Querying all parameters:
  SHOW VARIABLES LIKE '%';

----End

17.2 Modifying Parameters of a DB Instance

You can modify parameters of a DB instance to optimize performance if needed.

Precautions

- To ensure DB instance stability, you can only modify the parameters that are available on the console.

- To apply certain parameter modifications, you need to reboot the DB instance. After you modify a parameter value, check the value in the **Effective upon Reboot** column. You are advised to perform the operation during off-peak hours.
- The value of **validate_password.length** cannot be smaller than that of **validate_password.number_count+validate_password.special_char_count +(2 * validate_password.mixed_case_count)**. Otherwise, the allowed minimum value of **validate_password.length** is used when the parameter template is applied.
- If you want to use a custom parameter template during instance creation, ensure that the value of **validate_password.length** in the template is at most 16. Otherwise, the DB instance fails to be created.
- If you want to use a custom parameter template during instance creation, ensure that the values of **validate_password.mixed_case_count**, **validate_password.number_count**, and **validate_password.special_char_count** are at most 4. Otherwise, the DB instance may fail to be created. The default value **1** is recommended.
- The value of **rds_compatibility_mode** depends on the TaurusDB kernel version.
- Before modifying parameters, make sure you understand their meanings and fully verify the changes in a test environment to avoid instance or service exceptions caused by inappropriate parameter settings.

Modifying Parameters of a DB Instance

Step 1 Log in to the management console.

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

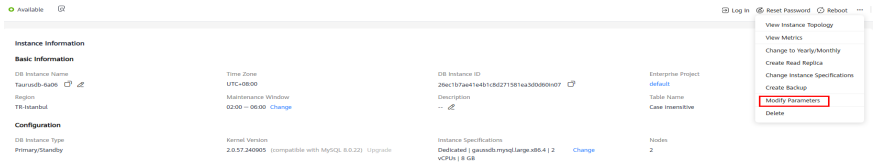
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

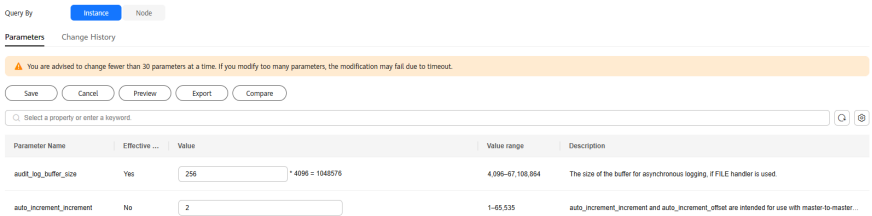
Alternatively, click in the upper right corner of the **Basic Information** page and choose **Modify Parameters**.

Figure 17-3 Choosing **Modify Parameters**



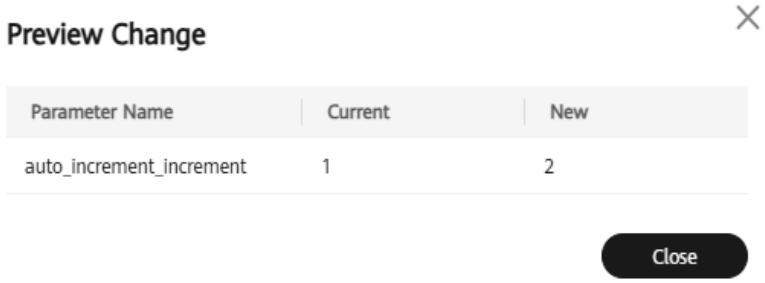
Step 6 On the **Parameters** tab, modify parameters.

Figure 17-4 Modifying parameters of a DB instance



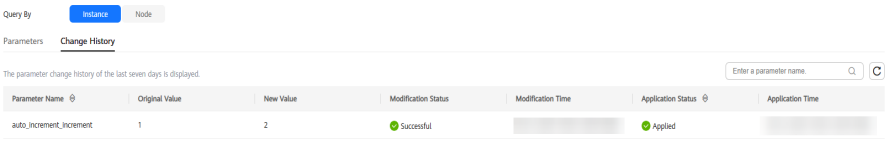
- To save the modifications, click **Save**. In the displayed dialog box, click **OK**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Figure 17-5 Previewing parameters



Step 7 After the parameters are modified, click **Change History** to view the modification records.

Figure 17-6 Viewing the modification records



----End

Modifying Parameters in a Parameter Template

You can modify parameters in a custom parameter template and then apply the template to multiple DB instances.

Step 1 Log in to the management console.

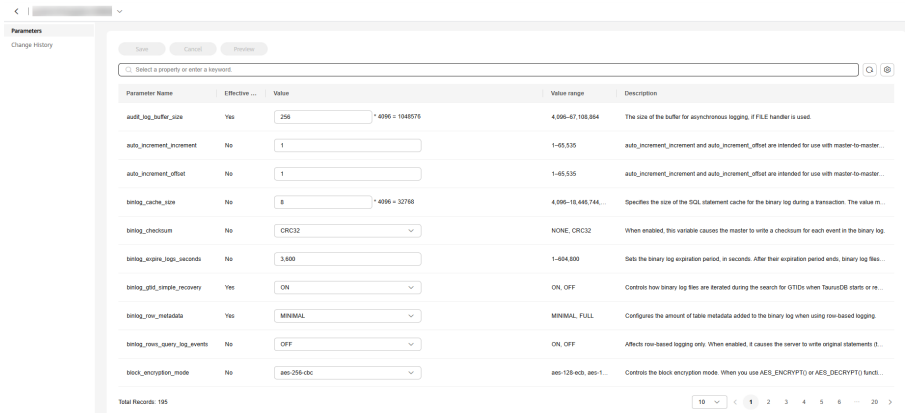
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 On the displayed **Parameters** page, modify parameters as required.

Figure 17-7 Modifying parameters in a parameter template

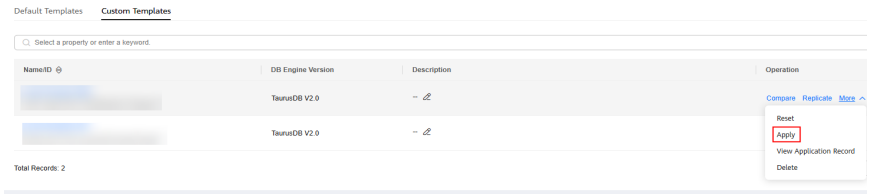


- To save the modifications, click **Save**. In the displayed dialog box, click **OK**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

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

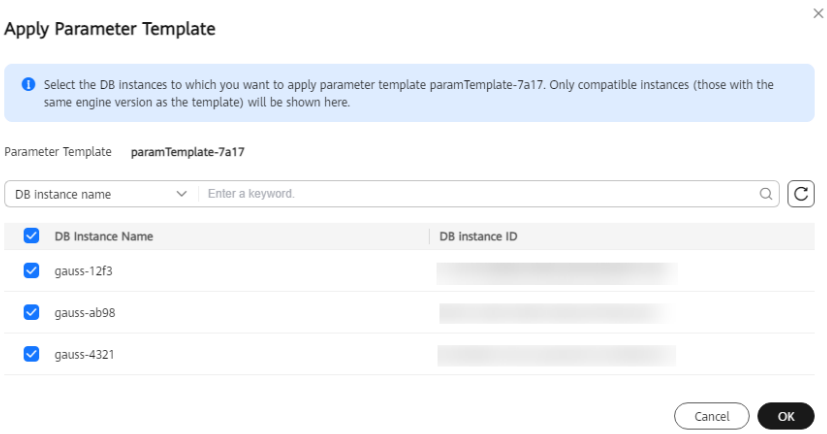
Step 7 After a parameter template is modified, the modification will not take effect until the template is applied to DB instances. On the **Parameter Templates** page, locate the parameter template you want to apply and choose **More** > **Apply** in the **Operation** column.

Figure 17-8 Applying a parameter template to a DB instance



Step 8 Select one or more DB instances and click **OK**.

Figure 17-9 Selecting DB instances



Step 9 After the parameter template is applied, click the DB instance name and check whether the parameters have been modified on the **Parameters** page.

----End

Common Parameters

Table 17-1 Common parameters

Parameter	Description	Reference
time_zone	Specifies the time zone of the server.	**How Do I Change the Time Zone?**
default_password_lifetime	Specifies the global automatic password expiration policy, in days.	**How Do I Configure a Password Expiration Policy for TaurusDB Instances?**
character_set_server	Specifies the server character set.	**How Do I Use the utf8mb4 Character Set to Store Emojis in a TaurusDB Instance?**
collation_server	Specifies the collation for the character set of the server. The collation must match the character set specified by character_set_server. Otherwise, the database cannot be started or restarted.	-
group_concat_max_len	Specifies the maximum permitted result length in bytes for the GROUP_CONCAT() function.	-

Parameter	Description	Reference
max_connections	Specifies the maximum number of concurrent client connections. If this parameter is set to **default**, the parameter value depends on how much memory there is.	**What Is the Maximum Number of Connections to a TaurusDB Instance?**
max_prepared_stmt_count	Limits the total number of prepared statements in the server. Too many statements may cause the server to run out of memory (OOM) and risk denial-of-service attacks. Configure this parameter as needed.	-
innodb_flush_log_at_trx_commit	Controls the balance between strict ACID compliance for commit operations, and higher performance that is possible when commit-related I/O operations are rearranged and done in batches. When this parameter is set to 0, the content of the InnoDB log buffer is written to the log file approximately once per second and the log file is flushed to disk. The default value of 1 is required for full ACID compliance. With this value, the contents of the InnoDB log buffer are written out to the log file at each transaction commit and the log file is flushed to disk. When this parameter is set to 2, the contents of the InnoDB log buffer are written to the log file after each transaction commit and the log file is flushed to disk approximately once per second.	**Viewing Suggestions on TaurusDB Parameter Tuning**
sql_mode	Specifies the SQL server mode.	-
binlog_expire_logs_seconds	Specifies the binary log expiration period in seconds. After their expiration period ends, binary log files can be automatically removed.	-

APIs

- Modifying Parameters in a Parameter Template
- Querying Parameter Templates
- Obtaining Details About a Parameter Template

17.3 Viewing Suggestions on TaurusDB Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect database performance. This section describes some important parameters for your reference. For details, see **MySQL official website**.

For details on how to modify TaurusDB parameters on the console, see **Modifying Parameters of a DB Instance**.

innodb_flush_log_at_trx_commit

- Default value: **1**
- Function: Controls the balance between strict ACID compliance for commit operations and higher performance.
- Parameter setting:
  - **1**: InnoDB writes data in the log buffer to log files and then flushes the data to distributed storage at each transaction commit. The ACID properties of transactions are ensured.
  - **0**: InnoDB writes data in the log buffer to log files and then flushes the data to distributed storage every second.
  - **2**: InnoDB writes data in the log buffer to the file system cache at each transaction commit, and flushes the data to distributed storage every second.
- Impact: If this parameter is not set to **1**, data security is not guaranteed. One second of transactions can be lost in a crash.
- Recommended value for PoC: **0**. This setting can significantly improve write performance in low concurrency scenarios.

rds_global_sql_log_bin

CAUTION

- In 2.0.42.230601 and earlier versions, binlog is enabled by default. To enable or disable it, you need to configure the **log-bin** parameter and then reboot your instance.
- In 2.0.45.230900 and later versions, binlog is disabled by default. To enable or disable it, you need to configure the **rds_global_sql_log_bin** parameter.

- Default value: **OFF**
- Function: Controls whether to enable or disable binlog. TaurusDB uses a shared storage architecture. The primary node and read replicas in a DB instance do not depend on binlogs for data synchronization. You can disable binlog as required.
- Parameter setting:
  - **OFF**: Binlog is disabled. The setting is applied to both existing and new connections without an instance reboot.
  - **ON**: Binlog is enabled. The setting is applied to both existing and new connections without an instance reboot.
- Impact: Enabling or disabling it does not affect your instance.
- Recommended value for PoC: **OFF**. This setting can improve write performance.

rds_plan_cache

CAUTION

- This feature can be enabled in 2.0.51.240300 and later versions.
- **rds_plan_cache** uses the memory allocated by the **stmt mem** memory area instead of the **innnodb_buffer** memory.
- **rds_plan_cache_allow_change_ratio**: Table data change rate caused by query operations such as DML. If the change rate exceeds this parameter value, plan caches become invalid. If this parameter is set to **0**, plan caches are not affected by the table data change ratio. They are always valid.

- Default value: **OFF**
- Function: Controls whether to cache the execution plan of a PREPARE statement.
- Parameter setting: If **rds_plan_cache** is set to **ON**, the execution plan of the PREPARE statement is cached. The cached execution plan can be reused in the next execution, improving query performance.
- Impact: The query performance of the PREPARE statement is greatly improved, and the select_random_ranges test model of sysbench is significantly enhanced.
- Recommended value for PoC: **ON**. This setting can improve query performance.

17.4 Introducing the High-Performance Parameter Template

To improve database performance, TaurusDB provides a high-performance parameter template. You can select this template when buying an instance.

This section explains the parameter settings in the high-performance parameter template and how the template enhances performance.

Introduction

The high-performance parameter template is a set of optimized configuration parameters that aim to enhance the performance and reliability of database servers. The parameter settings in the template can be adjusted based on different application scenarios and hardware configurations.

The parameters in the high-performance parameter template are as follows.

Table 17-2 Parameter description

Parameter	Description	Value in the High-Performance Template	Value in the Default Template
innodb_flush_log_at_trx_commit	If this parameter is set to 0, logs are not flushed to disks when transactions are committed. Instead, they are only flushed once per second or when the log buffer (**innodb_log_buffer_size**) is full. This provides low durability but high performance.	0	1
rds_plan_cache	If this parameter is set to **ON**, an execution plan is cached. The next time the same query is executed, the cached execution plan can be reused, which improves the database's query performance.	ON	OFF

Application Scenarios and Potential Risks

Generally, the high-performance parameter template can improve database performance. However, it should be adjusted based on specific application scenarios and hardware configurations.

While the template is designed to enhance performance and reliability of database servers, it does come with some risks during database usage.

- Setting **innodb_flush_log_at_trx_commit** to **0** can improve low-concurrency write performance, but in extreme cases, it may result in data loss of up to one second.
- Setting **rds_plan_cache** to **ON** can improve query performance because the execution plan of a PREPARE statement is cached and the optimizer does not need to generate an execution plan again. However, it may not be effective in all read/write scenarios.

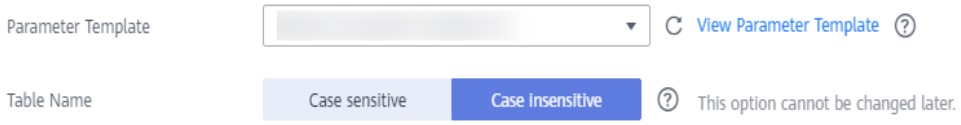
Constraints

The kernel version of your TaurusDB instance must be 2.0.51.240300 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**

Usage

You can select the high-performance parameter template when buying an instance.

Figure 17-10 Selecting the high-performance parameter template



Performance Comparison

Test environment:

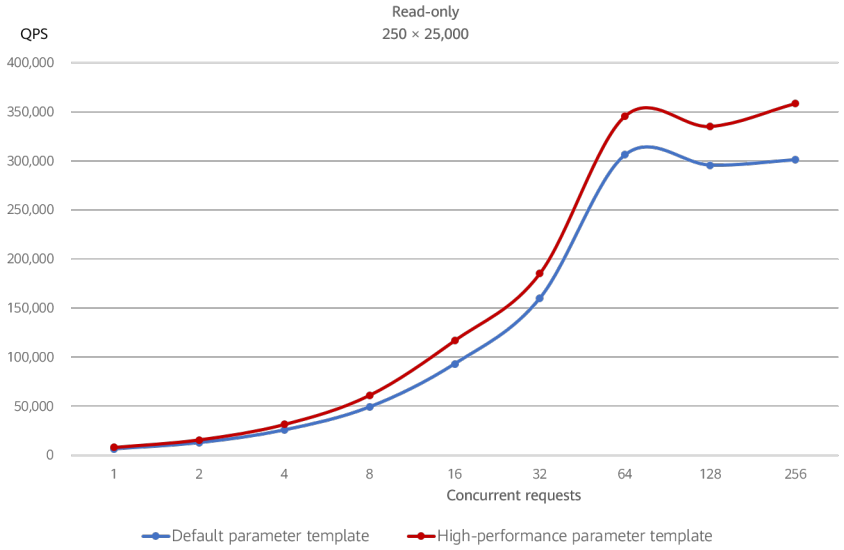
- TaurusDB instance specifications: Dedicated, 8 vCPUs | 32 GB
- Kernel version: 2.0.51.240300

Sysbench test process:

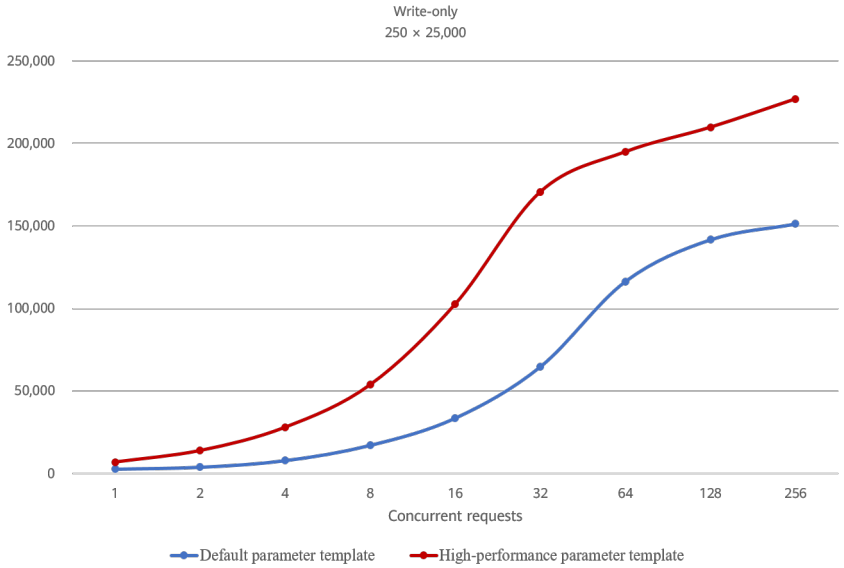
- Test scenarios: read-only, write-only, and read/write
- Data volume: 250 tables x 25,000 rows of data in read-only scenarios, 250 tables x 25,000 rows of data in write-only scenarios, and 25 tables x 250,000 rows of data in read/write scenarios
- Performance metric: queries per second (QPS) in 1, 2, 4, 8, 16, 32, 64, 128, and 256 concurrent requests. QPS indicates the number of SQL statements executed by the database per second.

Test results:

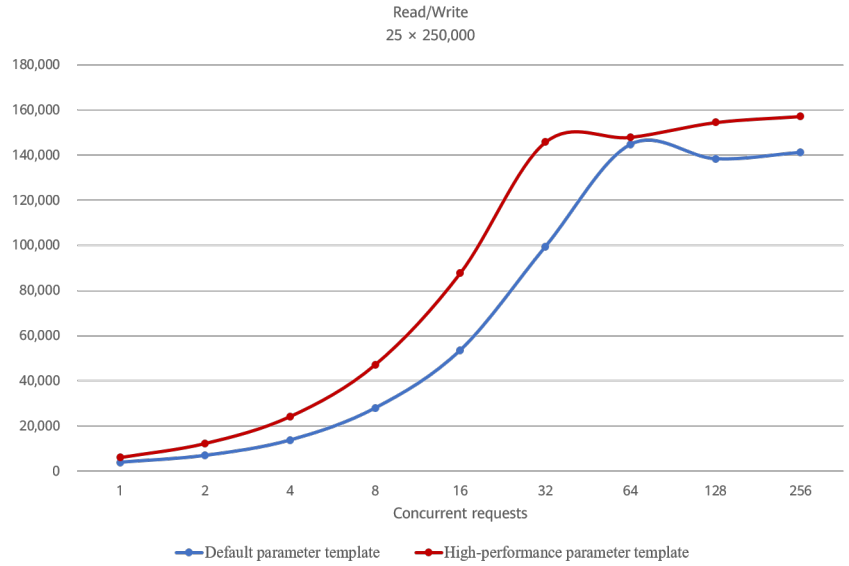
- Read-only model



- Write-only model



- Read/Write model



Conclusion: The preceding sysbench test results show that the high-performance parameter template significantly improves the database performance.

17.5 Using a Parameter Template

17.5.1 Creating a Custom Parameter Template

You can create custom parameter templates and apply them to one or more DB instances.

There are default parameter templates and custom parameter templates.

Precautions

- Each user can create a maximum of 100 parameter templates.
- All TaurusDB engines share the parameter template quotas.

Procedure

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

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

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

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

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

Figure 17-11 Creating a parameter template

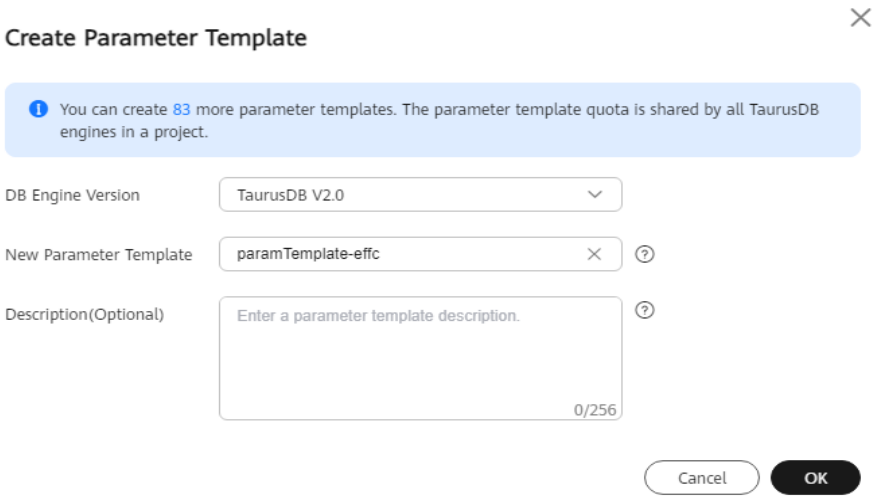


Table 17-3 Parameter description

Parameter	Description
DB Engine Version	Select **TaurusDB V2.0**.
New Parameter Template	The template name consists of 1 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.) are allowed.
Description	The description can consist of up to 256 characters. It cannot contain carriage returns or any of the following special characters: >!<"&'=

Step 6 Check that the custom template is displayed on the **Custom Templates** tab.

----End

APIs

- Creating a Parameter Template
- Querying Parameter Templates
- Obtaining Details About a Parameter Template

17.5.2 Applying a Parameter Template

After a parameter template is created or modified, you need to apply it to the desired DB instances.

Precautions

- The parameter **innodb_buffer_pool_size** is determined by the memory. Instances of different specifications have different value ranges. If this

parameter value is out of range of the instance to which the parameter template is applied, the maximum value within the range is used.

- A parameter template can be applied only to instances of the same DB engine version.

Procedure

Step 1 Log in to the management console.

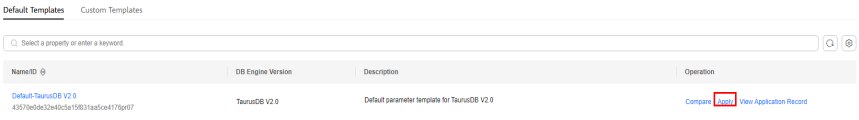
Step 2 Click in the upper left corner and select a region and project.

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

Step 4 On the **Parameter Templates** page, apply a default template or a custom template to DB instances.

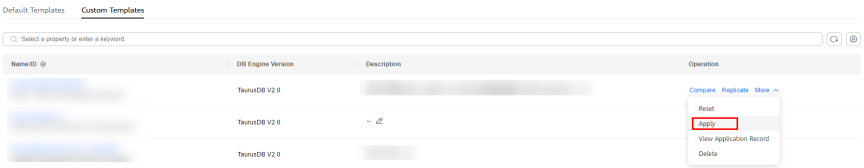
- To apply a default template, click **Default Templates**, locate a parameter template and click **Apply** in the **Operation** column.

  Figure 17-12 Applying a default parameter template to DB instances



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

  Figure 17-13 Applying a custom parameter template to DB instances



Step 5 In the displayed **Apply Parameter Template** dialog box, select DB instances and click **OK**.

Step 6 After the parameter template is applied, view the name or ID of the DB instance to which the parameter template is applied, application status, application time, and failure cause.

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

----End

APIs

Applying a Parameter Template

17.5.3 Replicating a Parameter Template

If you already have a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template. You can also replicate the parameter template to generate a new parameter template for future use.

Precautions

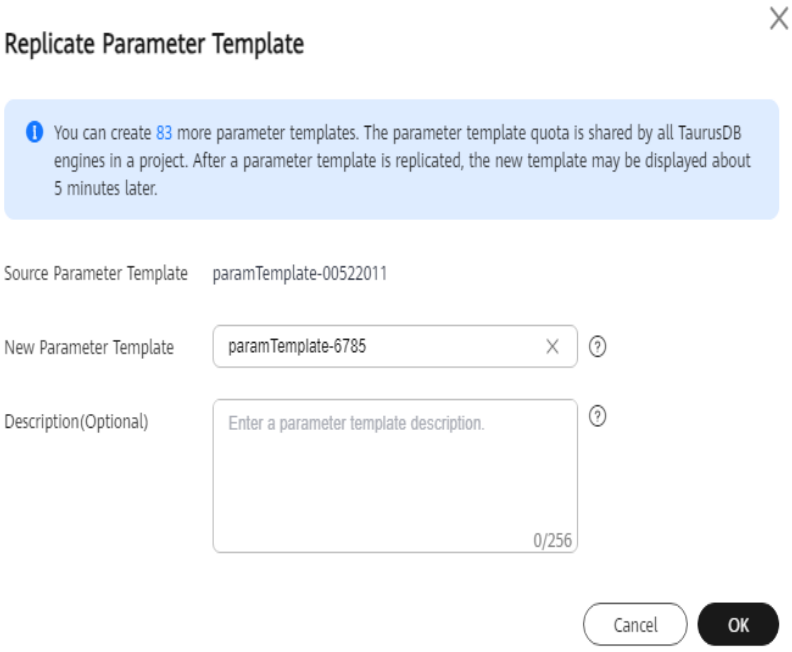
- After the parameter template is replicated, the new template will be displayed about 5 minutes later.
- Default parameter templates cannot be replicated, but you can create custom parameter templates based on those default templates.

Replicating a Custom Parameter Template

Step 1 On the **Parameter Templates** page, click **Custom Templates**. Locate the parameter template to be replicated and click **Replicate** in the **Operation** column.

Step 2 In the displayed dialog box, set required parameters and click **OK**.

Figure 17-14 Replicating a Custom Parameter Template



- The template name consists of 1 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), underscores (_), and periods (.) are allowed.
- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:

  >!<"&'=

Step 3 Check that a new template is generated on the **Custom Templates** tab.

----End

17.5.4 Resetting a Parameter Template

You can reset all parameters in a custom parameter template to their default settings.

Procedure

Step 1 Log in to the management console.

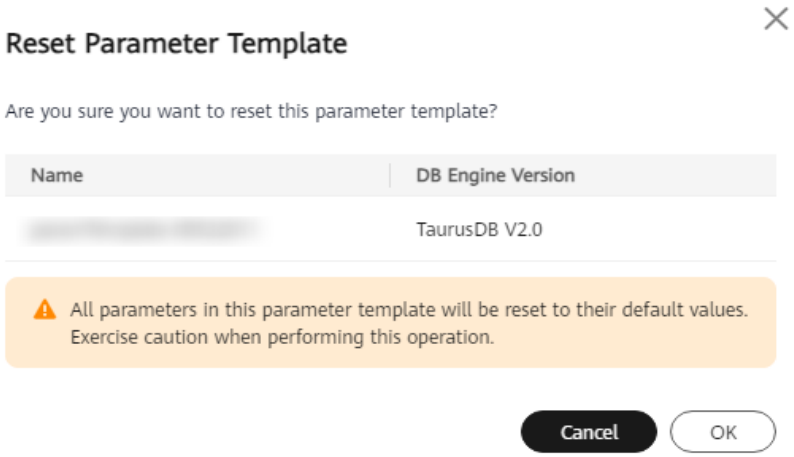
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page, choose **Databases** > **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 **OK** to reset all parameters to their default values.

Figure 17-15 Confirming the reset



NOTE

After you reset a parameter template, view the status of the instance to which the parameter template applies in the instance list. If the status is **Parameter change. Pending reboot**, you must reboot the instance.

----End

17.5.5 Comparing Parameter Templates

You can compare instance parameters with a parameter template to see the differences of parameter settings. You can also compare parameter templates to see the differences of parameter settings.

Comparing Instance Parameters with Those in a Specified Parameter Template

Step 1 Log in to the management console.

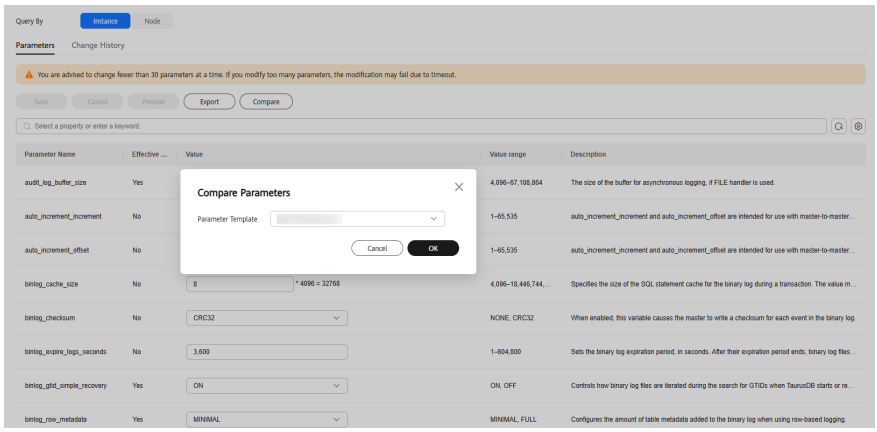
Step 2 Click in the upper left corner and select a region and project.

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

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

Step 5 In the navigation pane, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Figure 17-16 Comparing instance parameters with those in a specified parameter template



Step 6 In the displayed dialog box, select a parameter template and click **OK** to compare two parameters.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

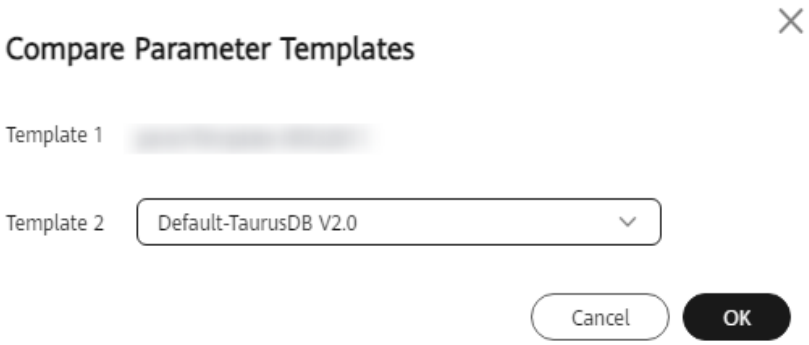
----End

Comparing Parameter Templates

Step 1 On the **Parameter Templates** page, click **Default Templates** or **Custom Templates**. Locate a parameter template and click **Compare** in the **Operation** column.

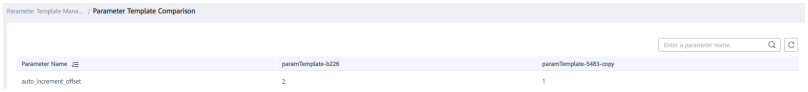
Step 2 In the displayed dialog box, select a parameter template and click **OK**.

Figure 17-17 Selecting a parameter template to be compared



- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

Figure 17-18 Comparing parameter templates



----End

17.5.6 Exporting a Parameter Template

You can export parameter template details (parameter names, values, and descriptions) of an instance to a custom template or an Excel file for review and analysis.

Procedure

Step 1 Log in to the management console.

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

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

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

Step 5 In the navigation pane, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

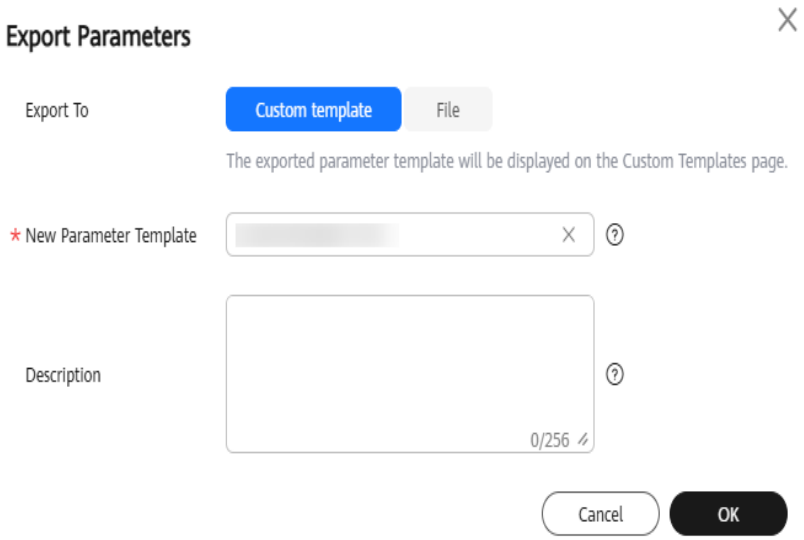
Step 6 In the displayed dialog box, set **Export To** to **Custom template** or **File**, enter the template or file name, and click **OK**.

The template or file name must start with a letter and consist of 4 to 81 characters. It can contain only letters, digits, hyphens (-), and underscores (_).

- **Custom template**: Parameter template details (parameter names, values, and descriptions) of the instance is exported to a new custom template, which will be displayed on the **Custom Templates** tab of the **Parameter Templates** page.

- **File**: Parameter template details (parameter names, values, and descriptions) of the instance is exported to an Excel file.

Figure 17-19 Exporting parameters



----End

17.5.7 Modifying the Description of a Parameter Template

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

Precautions

You cannot modify the description of any default parameter template.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page, choose **Databases** > **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 **OK** to submit the modification or **Cancel** to cancel the modification.

- After the modification is successful, you can view the new description in the **Description** column.
- The description consists of up to 256 characters. It cannot contain carriage returns or any of the following special characters:

>!<"&'=

----End

APIs

- **Modifying Parameters in a Parameter Template**
- **Querying Parameter Templates**
- **Obtaining Details About a Parameter Template**

17.5.8 Deleting a Parameter Template

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

Precautions

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

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

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

Step 3 Click  in the upper left corner of the page, choose **Databases** > **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 Click **OK**.

----End

APIs

Deleting a Parameter Template

18 Log Management

18.1 Configuring Log Reporting

Scenarios

You can view database-level logs on the **Logs** page, including error logs and slow SQL query logs.

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

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 a given log stream.
- You can enable or disable log reporting for a maximum of 10 instances at a time.
- 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.
- The log reporting configuration is not applied immediately. There is a delay of about 10 minutes.
- If log reporting is disabled, logs generated for the DB instance will not be reported to LTS.
- You can only enable either error log reporting to LTS or slow log reporting to LTS.
- After this function is enabled, audit logs record all requests sent to your DB instance and are stored in LTS.

Enabling Log Reporting

Step 1 Log in to the management console.

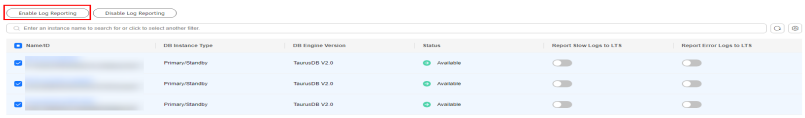
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

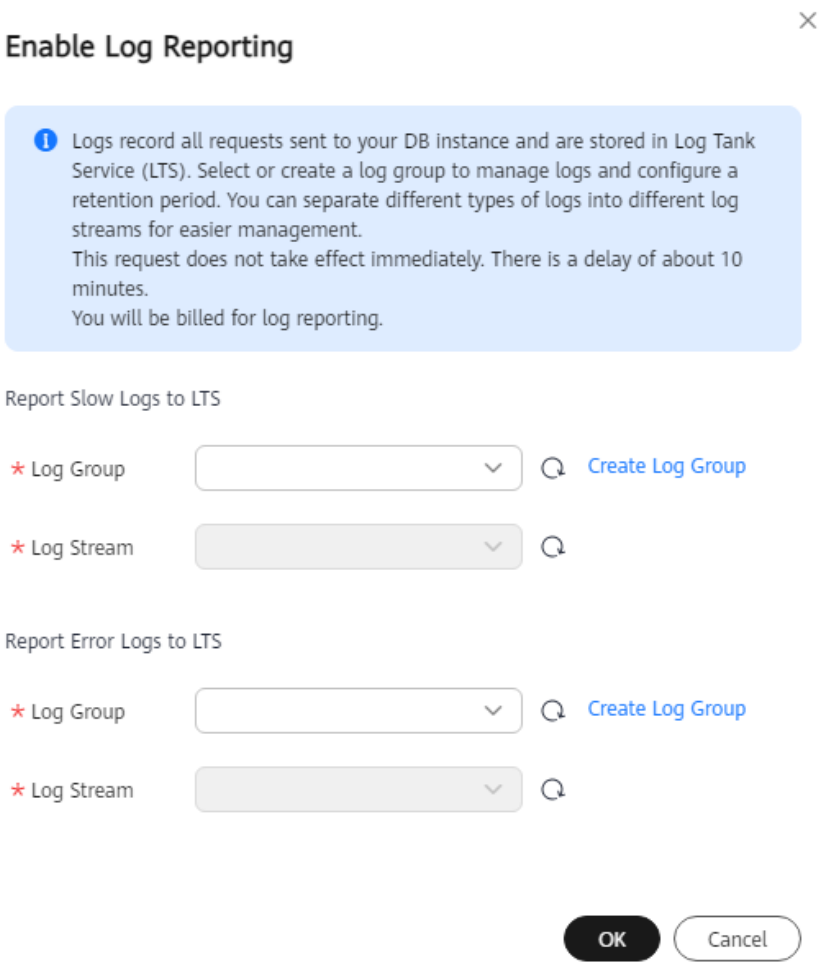
Step 5 Select one or more instances and click **Enable Log Reporting**.

Figure 18-1 Enabling log reporting for multiple instances



Step 6 In the displayed dialog box, select a log group and log stream, and click **OK**.

Figure 18-2 Enabling log reporting



----End

Disabling Log Reporting

Step 1 Log in to the management console.

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

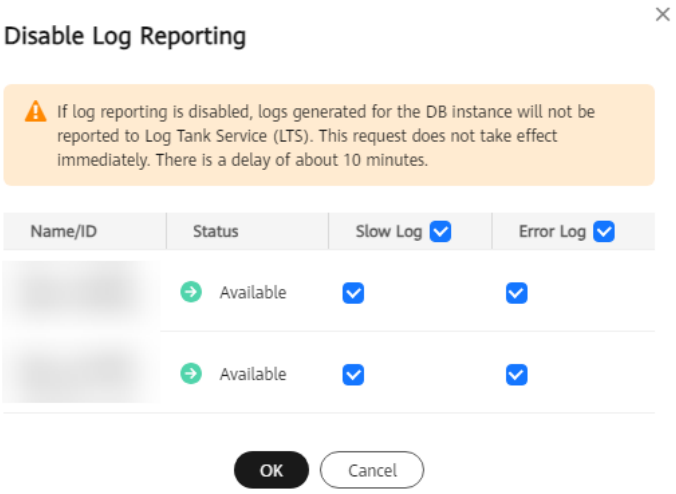
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 Disable log reporting in either of the following ways:

- Disabling log reporting for multiple instances
  a. Select one or more instances and click **Disable Log Reporting**.
  b. In the displayed dialog box, click **OK**.

  Figure 18-3 Disabling log reporting



- Disabling log reporting for a single instance
  a. Locate an instance and click in the **Report Error Logs to LTS** or **Report Slow Logs to LTS** column.
  b. In the displayed dialog box, click **Yes**.

Figure 18-4 Disabling slow log reporting

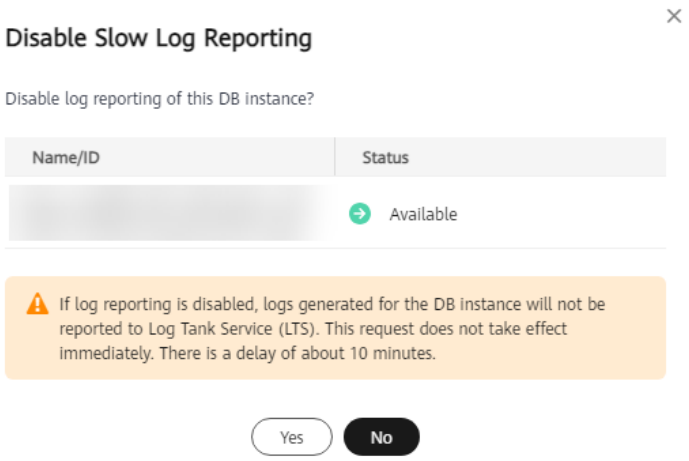
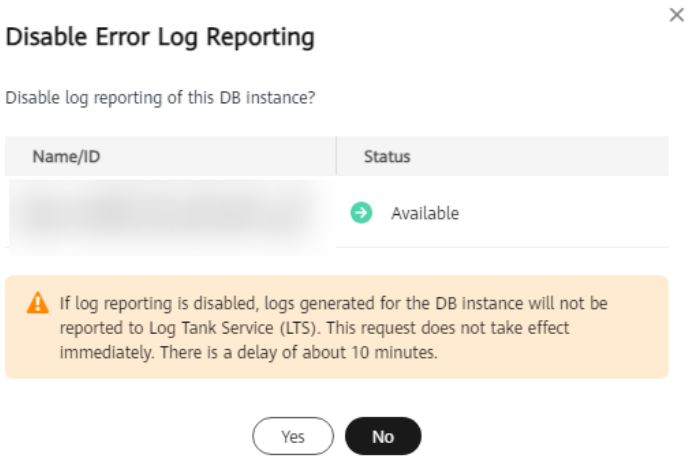


Figure 18-5 Disabling error log reporting



----End

18.2 Managing Error Logs of a DB Instance

Error logs contain logs generated while the database is running. They can help you analyze problems with the database.

Viewing Log Details

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

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

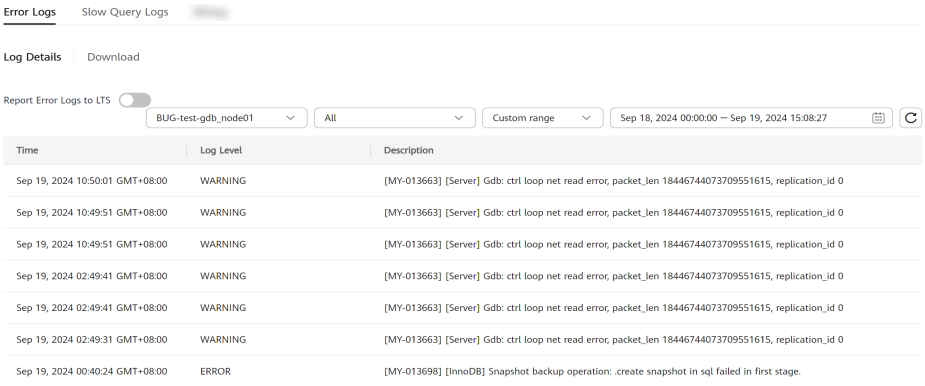
Step 6 On the **Error Logs** page, view error logs of different nodes, at different log levels, and within a specified time range.

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.

Figure 18-6 Viewing error logs



----End

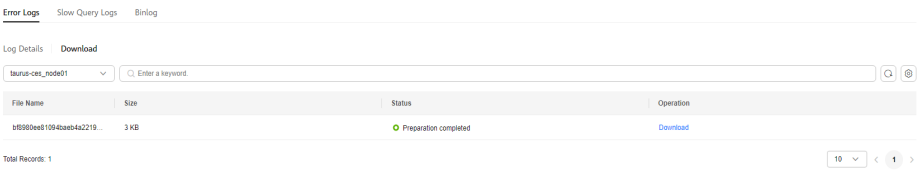
Downloading an Error Log

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

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

Step 3 On the **Error Logs** tab, click **Download**. Locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 18-7 Downloading an error log



- The system automatically loads the download preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.

  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** state cannot be downloaded.
- Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.
- You can select the logs to be downloaded by node.

----End

Reporting Error Logs to LTS

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

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

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

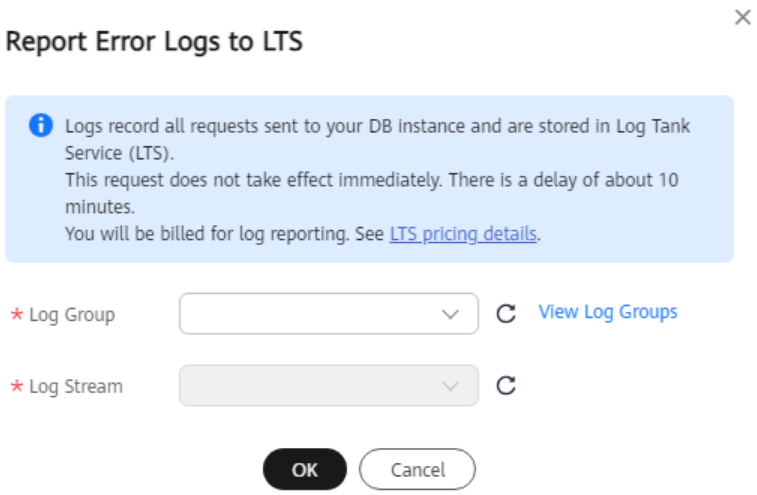
Step 4 On the **Instances** page, click the instance name.

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

Step 6 On the **Error Logs** tab, click next to **Report Error Logs to LTS**.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 18-8 Reporting error logs to LTS



----End

APIs

Querying Error Logs

18.3 Managing Slow Query Logs of a DB Instance

Scenarios

Slow query logs record statements that exceed **long_query_time** (10 seconds by default). You can view log details and statistics to identify statements that are executing slowly and optimize the statements.

Parameter Description

Table 18-1 Parameters related to slow queries

Parameter	Description
long_query_time	Specifies how many seconds a SQL query has to take to be recorded in slow query logs. The default value is **10**. You are advised to set this parameter to **1**. The lock wait time is not calculated into the query time.
log_queries_not_using_indexes	Controls whether to record queries that do not use indexes. The default value is **OFF**. The parameter is not affected by the value of **long_query_time**.
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**.

Constraints

- You can view the slow query log records of a specified statement or a specific time period.
- Only SELECT statements return the number of result rows. The number of result rows for the INSERT, UPDATE, DELETE, and CREATE statements is 0 by default.
- You can view slow query logs of a specified database name (which cannot contain any special characters). The database name supports only exact search.
- Slow query logs only record executed statements whose execution duration exceeds the threshold.
- The **long_query_time** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **long_query_time** is changed from 1s to 0.1s, TaurusDB starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. For example, a 1.5s SQL statement that was recorded when the threshold was 1s will not be deleted now that the new threshold is 2s.

- If the length of a single line of a SQL statement exceeds 16 KB, the statement will be truncated. When you view slow query log details, the SQL statement may be incomplete after special processing and is for reference only.

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 **Logs**.

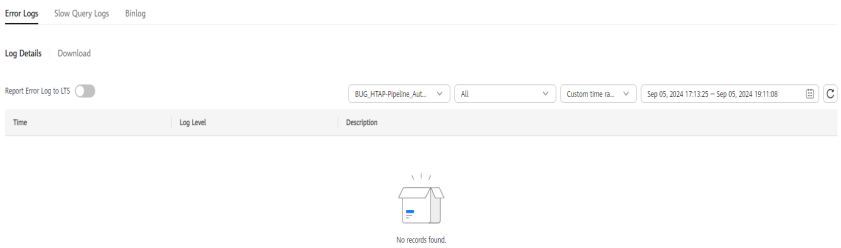
Step 6 On the **Slow Query Logs** page, view the slow query log details.

You can view slow query logs of different nodes and SQL statement types in a given database.

Supported SQL statement types include SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, and DROP.

You can also view slow query logs in a specified time period by clicking and specifying a time period.

Figure 18-9 Viewing slow query logs



----End

Enabling Show Original Log

NOTE

By default, SQL statements are displayed anonymously. If **Show Original Log** is enabled, SQL statements in the logs will be displayed in plaintext.

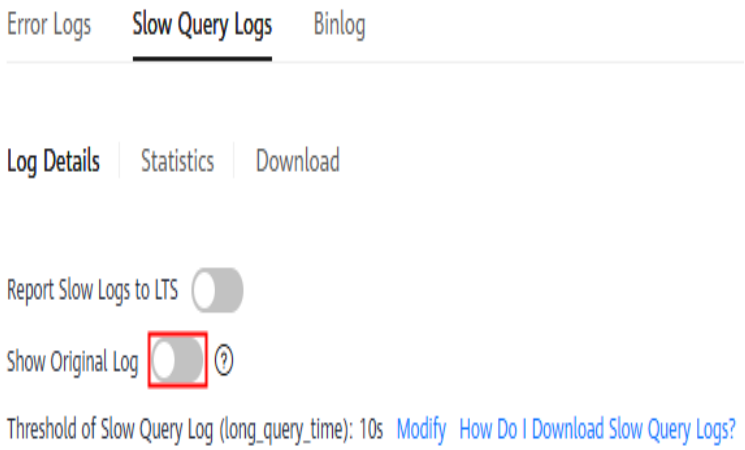
Logs displayed in plaintext will be automatically deleted 30 days later. If a DB instance is deleted, its related logs will also be deleted.

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

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

Step 3 On the **Slow Query Logs** tab, click next to **Show Original Log**.

Figure 18-10 Enabling Show Original Log



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

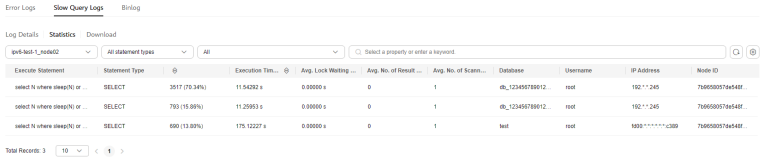
----End

Viewing Statistics

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

Step 2 In the navigation pane, choose **Logs**. On the **Slow Query Logs** tab, click **Statistics** to view details.

Figure 18-11 Statistics



- 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, or time period. The database name supports only exact search.

- If any database name in the slow log statistics contains special characters such as < > ', the special characters will be escaped.

----End

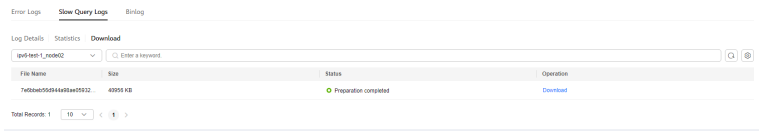
Downloading a Slow Query Log

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

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

Step 3 On the **Slow Query Logs** tab, click **Download**. Locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 18-12 Downloading a slow query log



- The system automatically loads the download preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation completed**.
  - If the preparation for download fails, the log status is **Abnormal**.

  Logs in the **Preparing** or **Abnormal** state cannot be downloaded.
- Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.
- You can select the logs to be downloaded by node.

----End

Reporting Slow Logs to LTS

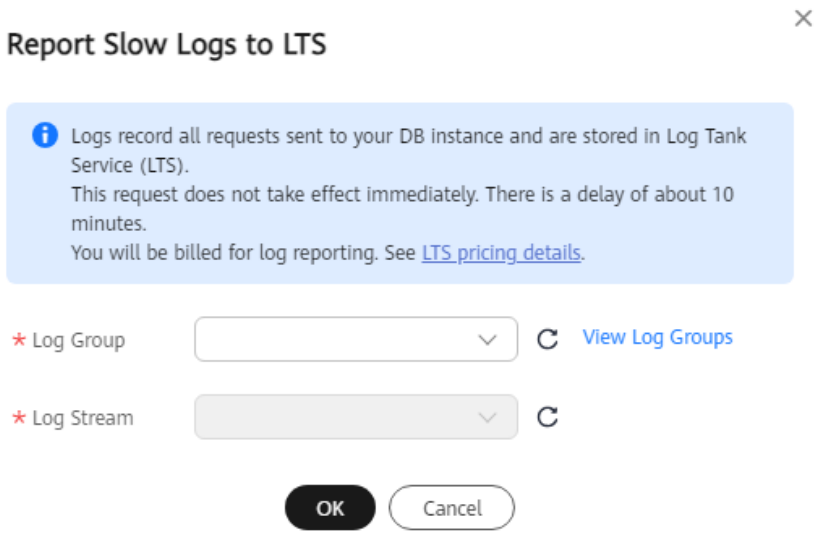
Step 1 On the **Instances** page, click the instance name.

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

Step 3 On the **Slow Query Logs** page, click  next to **Report Slow Logs to LTS**.

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

Figure 18-13 Reporting slow logs to LTS



----End

APIs

Querying Slow Query Logs

18.4 Configuring SQL Explorer for a DB Instance

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

You can enable SQL Explorer on the **DAS** console.

Constraints

SQL Explorer cannot record all data. It has the following constraints:

- Some data cannot be recorded if a buffer overrun occurs.
- If the size of a SQL statement exceeds the value of **rds_sql_tracer_max_record_size**, the statement is not recorded by default.

  rds_sql_tracer_max_record_size controls the maximum size of a SQL statement. To change its value, see **Modifying Parameters of a DB Instance**.

18.5 Querying and Downloading Binlogs (OBT)

Binlogs record all DDL and DML statements (except data query statements). You can download binlogs to a local PC for further analysis.

This section describes how to enable binlog, and then query and download binlogs on the TaurusDB console.

Billing

Binlogs are stored in OBS buckets. For the billing details, see **How Is TaurusDB Backup Data Billed?**

Prerequisites

- Binlog can only be enabled when the following conditions are met:
  - If the kernel version of your DB instance is earlier than 2.0.45.230900, the value of **log-bin** must be **ON**. To view and modify the parameter value, see **Modifying Parameters of a DB Instance**.
  - If the kernel version of your DB instance is 2.0.45.230900 or later, the value of **rds_global_sql_log_bin** must be **ON**.

  For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- Before viewing and downloading binlogs, you need to enable binlog by referring to **Enabling Binlog**.

Constraints

To use binlogs, set the **log-bin** or **rds_global_sql_log_bin** parameter to **ON**. To query or download binlogs on the management console, **submit a service ticket**.

Enabling Binlog

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

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

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

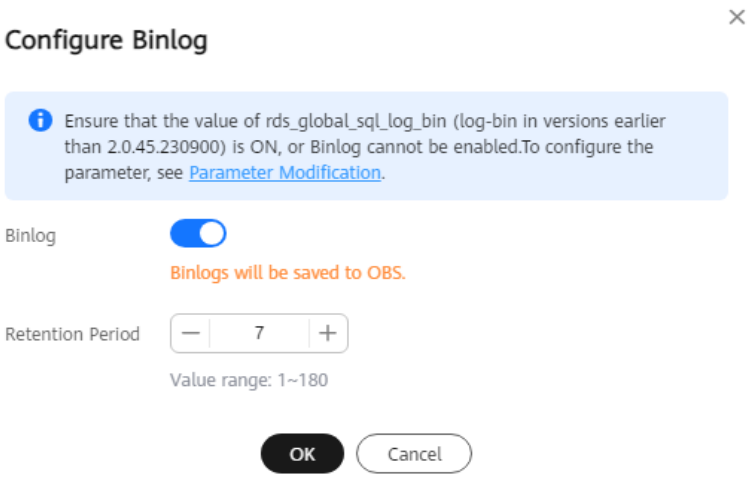
Step 4 On the **Instances** page, click the instance name.

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

Step 6 Click the **Binlog** tab.

Step 7 Click **Configure Binlog**. In the displayed dialog box, enable **Binlog** and set **Retention Period**.

Figure 18-14 Configuring binlog



- The retention period ranges from 1 to 180 days.
- After binlog is disabled, the generated logs will be automatically deleted after the retention period expires. Deleted logs cannot be restored. Exercise caution when disabling binlog.

----End

Querying and Downloading Binlogs

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

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

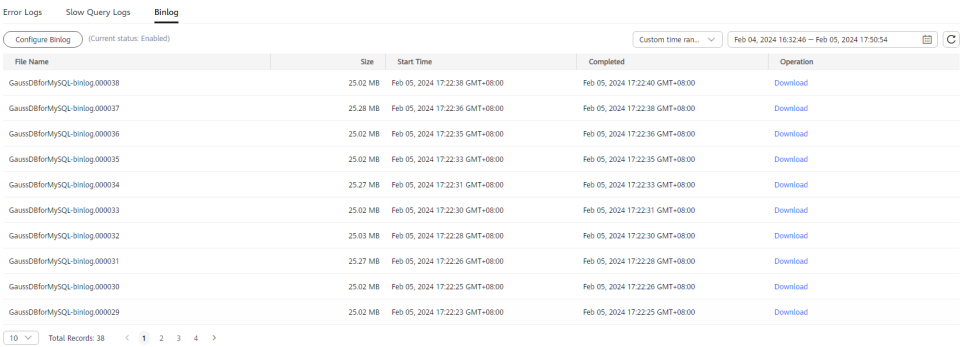
Step 3 Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.

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

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

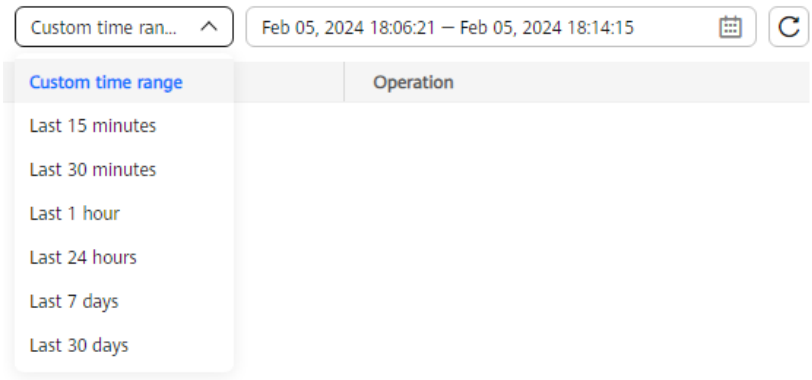
Step 6 Click the **Binlog** tab.

Figure 18-15 Viewing binlogs



- View binlogs generated in the last 15 minutes, last 30 minutes, last 1 hour, last 24 hours, last 7 days, last 30 days, or a custom time range.

Figure 18-16 Selecting a time range



- Click **Download** in the **Operation** column to download a binlog to a local PC.

----End

18.6 Enabling SQL Audit (OBT)

After you enable the SQL audit function, all SQL operations will be recorded in log files. You can **download** audit logs to view details.

By default, SQL audit is disabled because enabling this function may affect database performance. This section describes how to enable, modify, or disable SQL audit.

Supported Kernel Versions

The kernel version must be 2.0.60.241200 or later. If your DB engine version is too early, upgrade it to the latest version by referring to **Upgrading the Minor Version of a DB Instance**. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**

Constraints

Table 18-2 Audit log constraints

Audit Log Type	Constraint
SQL audit logs	• This function is now in the OBT phase. To use it, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. During the OBT, you will not be billed for enabling this function or for the OBS storage used for audit logs. • This function is available to TaurusDB Enterprise Edition instances and serverless instances, but not to multi-primary instances. • Only SQL statements that have been executed in the kernel are logged. • During specification changes, configurations need to be synchronized. As a result, some SQL operations are not logged. • SQL audit logs use the Coordinated Universal Time (UTC) time, which is not affected by the time zone configuration. • After SQL audit is enabled, TaurusDB records SQL operations in audit logs. The generated audit logs are temporarily stored in the instance and then uploaded to OBS and stored in the backup space. • SQL audit logs that are beyond the retention period are cleared every hour. If you change the retention period of audit logs, expired audit logs will be deleted one hour later.

Enabling SQL Audit

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

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

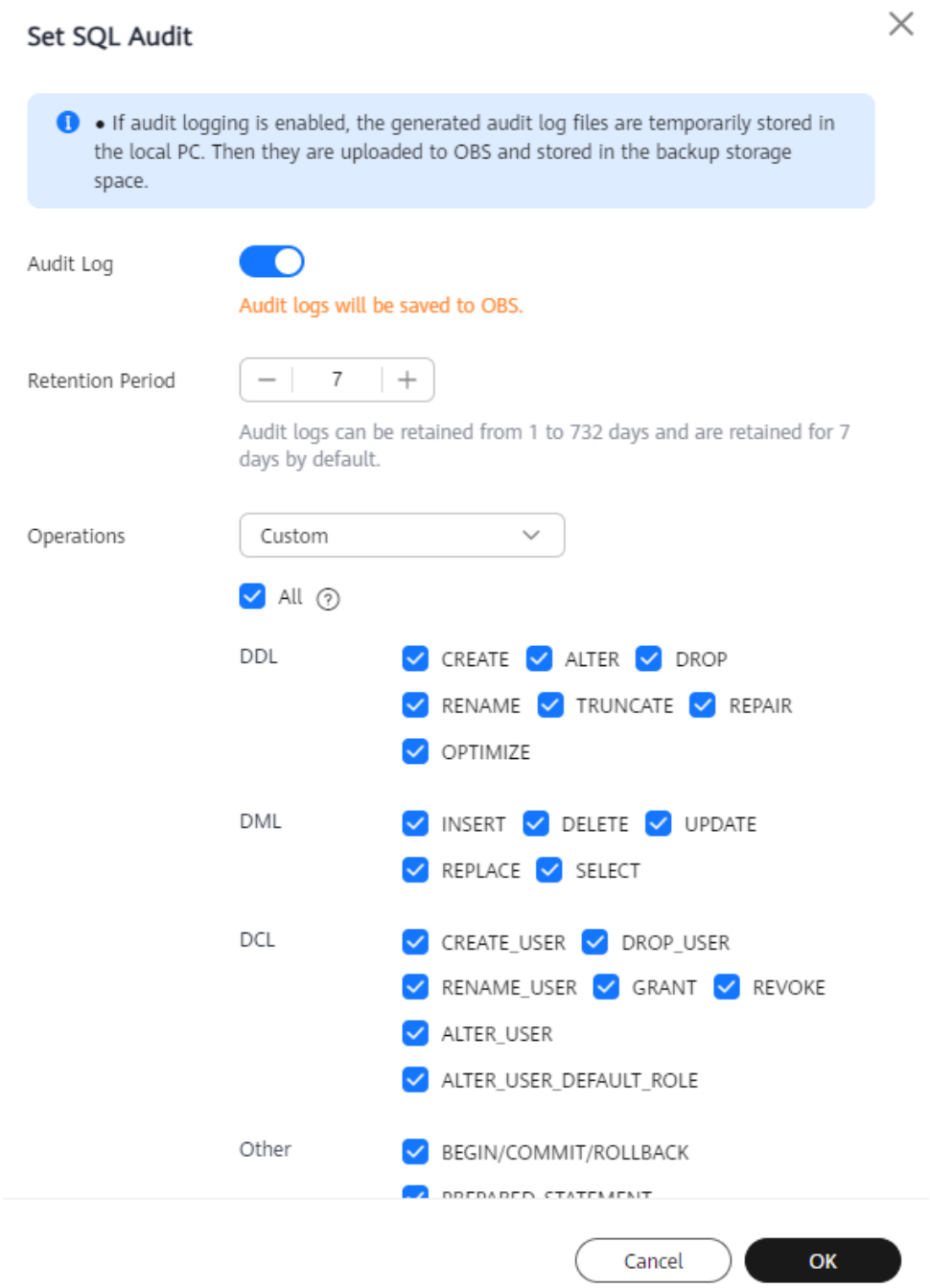
Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit** above the list. In the displayed dialog box, configure information as required and click **OK**.

- **Enabling or setting SQL audit**
  - To enable SQL audit, toggle  (disabled) to  (enabled).
  - Audit logs can be retained from 1 to 732 days and are retained for 7 days by default.

Figure 18-17 Setting SQL audit



The SQL statements executed by PreparedStatement and scheduled tasks through a MySQL client will be treated as **PREPARED_STATEMENT** and **CREATE**, respectively. However, the SQL statements executed by PreparedStatement through JDBC will not be recorded.

You can change the value of the **audit_log_buffer_size** parameter on the console to adjust the size of the audit log buffer. For details, see **Modifying Parameters of a DB Instance**.

Table 18-3 Parameter description

Parameter	Level	Description
audit_log_buffer_size	GLOBAL	Size of the audit log buffer, in bytes. Default value: **1048576**

After SQL audit is enabled, Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and other operation types are supported. The details are as follows:

Table 18-4 DDL types and operations

Type	Operation
CREATE	create_db, create_event, create_function, create_index, create_procedure, create_table, create_trigger, create_udf, create_view
ALTER	alter_db, alter_db_upgrade, alter_event, alter_function, alter_instance, alter_procedure, alter_table, alter_tablespace
DROP	drop_db, drop_event, drop_function, drop_index, drop_procedure, drop_table, drop_trigger, drop_view
RENAME	rename_table
TRUNCATE	truncate
REPAIR	repair
OPTIMIZE	optimize

Table 18-5 DML types and operations

Type	Operation
INSERT	insert, insert_select
DELETE	delete, delete_multi
UPDATE	update, update_multi
REPLACE	replace, replace_select
SELECT	select

Table 18-6 DCL types and operations

Type	Operation
CREATE_USER	create_user
DROP_USER	drop_user
RENAME_USER	rename_user
GRANT	grant_roles, grant
REVOKE	revoke, revoke_all, revoke_roles
ALTER_USER	alter_user
ALTER_USER_DEFAULT_ROLE	alter_user_default_role

Table 18-7 Other types and operations

Type	Operation
BEGIN/COMMIT/ROLLBACK	begin, commit, release_savepoint, rollback, rollback_to_savepoint, savepoint
PREPARED_STATEMENT	execute_sql,prepare_sql, dealloc_sql
CALL_PROCEDURE	call_procedure
KILL	kill
SET_OPTION	set_option
CHANGE_DB	change_db
UNINSTALL_PLUGIN	uninstall_plugin
INSTALL_PLUGIN	install_plugin
SHUTDOWN	shutdown
SLAVE_START	slave_start
SLAVE_STOP	slave_stop
LOCK_TABLES	lock_tables
UNLOCK_TABLES	unlock_tables
FLUSH	flush
XA	xa_commit,xa_end,xa_prepare,xa_recover,xa_rollback,xa_start

- **Disabling SQL audit**

  To disable SQL audit, toggle  (enabled) to  (disabled).

If you select the check box "I acknowledge that after audit log is disabled, all audit logs are deleted." and click **OK**, all audit logs will be deleted.

CAUTION

Deleted audit logs cannot be recovered. Exercise caution when performing this operation.

----End

18.7 Downloading SQL Audit Logs

If you **enable SQL audit**, the system records all SQL operations and uploads logs every half an hour or when the size is accumulated to 100 MB. You can download audit logs to view details. The minimum time unit of audit logs is second. By default, SQL audit is disabled. Enabling this function may affect database performance.

Constraints

You are advised to download no more than six audit log files at a time. Too many files can fail to be downloaded completely due to the limit on the number of concurrent requests of the browser.

Downloading SQL Audit 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 **SQL Audits**.

Step 6 On the displayed page, select a time range in the upper right corner, select SQL audit logs to be downloaded in the list, and click **Download** above the list to download SQL audit logs in batches.

Alternatively, select an audit log and click **Download** in the **Operation** column to download an individual SQL audit log.

Step 7 The following figure shows the SQL audit log content. For field descriptions, see **Figure 18-18**.

Figure 18-18 TaurusDB audit logs

```
"6","408543","1159","Connect",2020-03-20T03:35:05 UTC,"","","","","","[redacted]","
"7","408543","0","Quit",2020-03-20T03:35:05 UTC,"","","","","","[redacted]","
"8","408544","1159","Connect",2020-03-20T03:35:20 UTC,"","","","","","[redacted]","
"9","408544","0","Quit",2020-03-20T03:35:20 UTC,"","","","","","[redacted]","
"10","408546","1159","Connect",2020-03-20T03:35:35 UTC,"","","","","","[redacted]","
"11","408546","0","Quit",2020-03-20T03:35:35 UTC,"","","","","","[redacted]","
"12","408547","1159","Connect",2020-03-20T03:35:50 UTC,"","","","","","[redacted]","
"13","408547","0","Quit",2020-03-20T03:35:50 UTC,"","","","","","[redacted]","
```


Table 18-8 Audit log field description

Parameter	Description
record_id	ID of a single record, which is the unique global ID of each SQL statement recorded in the audit log.
connection_id	ID of the session executed for the record, which is the same as the ID in the **show processlist** command output.
connection_status	Session status, which is usually the returned error code of a statement. If a statement is successfully executed, the value **0** is returned.
name	Recorded type name. Generally, DML and DDL operations are QUERY, connection and disconnection operations are CONNECT and QUIT, respectively.
timestamp	UTC time for the record.
command_class	SQL command type. The value is the parsed SQL type, for example, SELECT or UPDATE. (This field does not exist if the connection is disconnected.)
sqltext	Executed SQL statement content. (This field does not exist if the connection is disconnected.)
user	Login account.
host	Login host. The value is **localhost** for local login and is empty for remote login.
external_user	External username.
ip	IP address of the remotely-connected client. For local connection, the field is empty.
default_db	Default database on which SQL statements are executed.

----End

19 Cold and Hot Data Separation (OBT)

19.1 What Is Cold and Hot Data Separation?

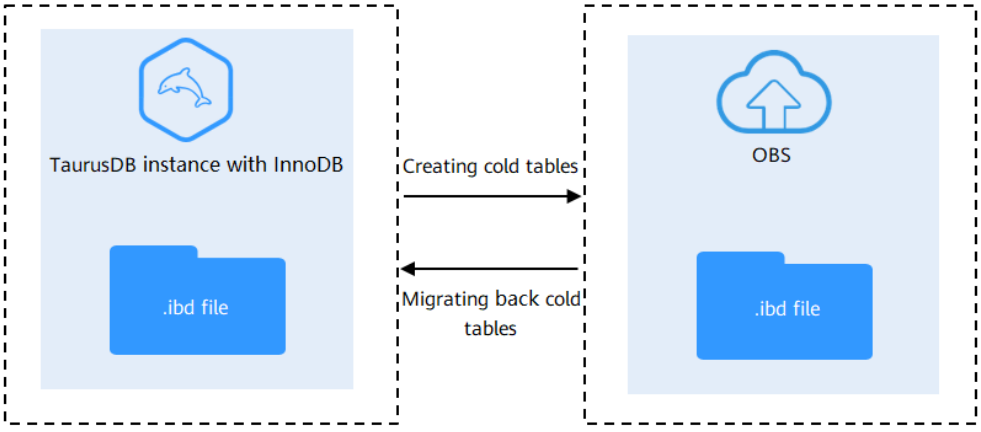
Scenarios

If there are some tables with almost no data reads or writes in your DB instance, you can use cold and hot data separation to dump their data to OBS. This helps reduce costs by managing cold and hot data more efficiently.

How It Works

TaurusDB allows you to dump infrequently used InnoDB tables to OBS. These dumped tables are called cold tables.

Figure 19-1 Diagram



- Creating cold tables

  To free up space and reduce storage costs, you can select tables that do not need to be modified temporarily and take up a lot of space as cold tables. Such tables will be dumped to OBS.
- Migrating back cold tables

To modify or frequently query certain cold tables, you can migrate the tables back to your DB instance, and the data of these tables will still be stored in OBS.

Billing

Cold data stored on OBS is billed based on the backup space usage.

Constraints

- To use cold and hot data separation, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- The kernel version of your DB instance must be 2.0.57.240905 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- Cold and hot data separation is unavailable for serverless instances, multi-primary instances, instances in a RegionlessDB cluster, or instances with TDE or cross-region backup enabled.
- After cold and hot data separation is enabled, cross-region backup and manual backup are not supported.
- Cold and hot data separation cannot be disabled after being enabled.
- After cold and hot data separation is enabled, only full data restoration is supported. Data can only be restored to a new or the original instance using backups or point-in-time recovery (PITR).
- Temporary tables and views cannot be archived.
- The first partition of a partitioned table cannot be archived.
- HASH, KEY, LINEAR HASH, LINEAR KEY, or LIST DEFAULT HASH partitions cannot be archived.
- Subpartitions or partitions with subpartitions cannot be archived.
- System catalogs cannot be archived.
- Tables with full-text indexes cannot be archived.
- Cold table operations are unavailable for frozen DB instances.

Precautions

- Do not run DDL or DML statements when creating a cold table.
- A DB instance with a cold table created cannot be used as the destination instance for full restoration or point-in-time restoration.
- Cold tables can only be queried and the query is slow. They do not support DDL or DML statements. You are advised to convert rarely accessed tables that store archived data to cold tables.
- If the table to be archived is a partitioned table, only one partition can be archived at a time. If multiple partitions need to be archived, archive each partition separately.
- If data is archived by partition, DDL statements can only be executed on partitions other than the archived partition. The first partition does not support DROP. REMOVE PARTITIONING is not supported. Table-level DDL statements are not supported.

- If there is a foreign key in a cold table, DDL statements cannot be executed on the primary table corresponding to the foreign key.
- During archiving, special characters in the database name, table name, and partition name need to be escaped.
- The bucket for storing archived cold data uses a single-AZ parallel file system. If there is an AZ-level fault, you may not be able to access the cold data.
- After a cold table is migrated back, the cold data will still be stored in the bucket.
- After migrating a cold table back, you must manually delete the cold data on the console after the automated backup retention period expires to avoid ongoing billing for the cold data.
- Restoring data from an instance with cold and hot data separation enabled to a new instance involves copying the cold data from the bucket, leading to a longer restoration duration.

How to Use

- Dumping cold table data

  You can create cold tables on the console. The data of the cold tables is stored on OBS, freeing up storage space and reducing storage costs. For details, see **Configuring a Cold Table on the Console**.
- Querying cold table data

  Just like querying data from any ordinary table, you can run SELECT statements to query cold table data. For details, see **Configuring a Cold Table Using SQL Statements**.
- Modifying cold table data

  To modify a cold table that has been dumped to OBS, you can migrate the table back to your instance on the console, but the data of the table is still stored in OBS. To delete data from OBS, **submit a service ticket**.
- Deleting cold table data

  After a cold table is created, DDL statements cannot be executed on the cold table. This means that the table cannot be deleted directly, and neither can the database it belongs to.

  To delete a cold table, migrate it back and run the **drop** command.

Disclaimer

- During migration using DRS, if you want to retain cold table data of a source database, migrate cold tables back to the source database and then migrate the database using DRS.
- It takes longer to query data in cold tables. If there are too many cold tables, a large number of slow query logs may be generated.
- To prevent an operation failure, do not create or migrate back a cold table when any other operation is being performed on the instance.

19.2 Configuring a Cold Table

This section describes how to configure a cold table.

You can configure a cold table in either of the following ways:

- **On the console**: You can create and migrate back a cold table on the console.
- **Using SQL statements**: You can create, query, and migrate back a cold table using SQL statements. If there are more than 100,000 tables in your DB instance, you can create and migrate back a cold table only using SQL statements.

Constraints

- To use cold and hot data separation, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- The kernel version of your TaurusDB instance must be 2.0.57.240905 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- Cold data archiving can only be enabled for cluster instances.
- Cold tables can only be queried using SELECT statements and the query speed is slow. Do not configure tables storing frequently updated data as cold tables.
- Hot and cold data separation cannot be enabled for instances with TDE or cross-region backup enabled.
- Only the structure of a cold table can be backed up. Cold data cannot be backed up or restored.
- During migration using DRS, if you want to retain cold table data of a source database, migrate cold tables back to the source database and then migrate the database using DRS. Otherwise, the cold tables will be migrated to empty tables of the destination database.
- To prevent a creation failure, do not run DDL or DML statements on the selected table.

Configuring a Cold Table on the Console

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

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

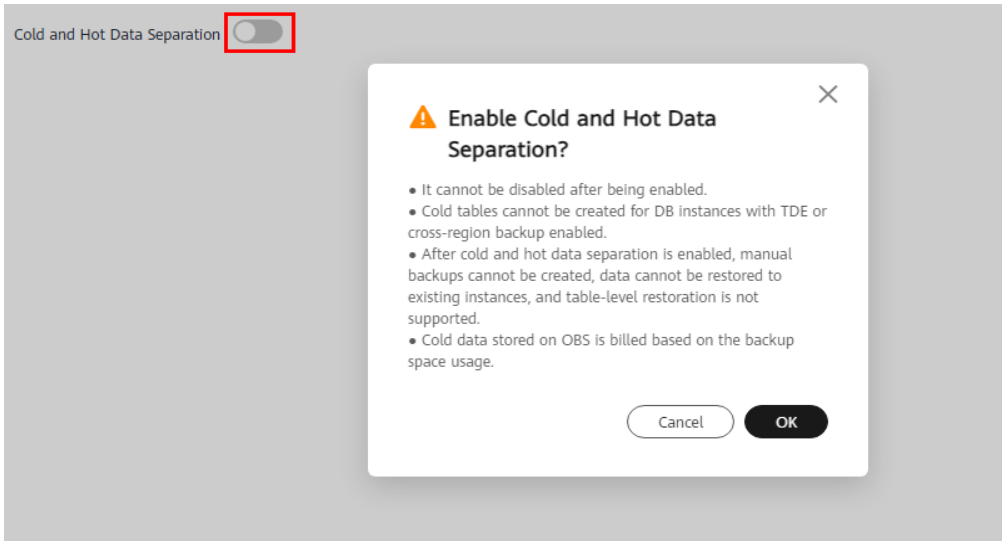
Step 3 Click  in the upper left corner of the page, choose **Databases** > **TaurusDB**.

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

Step 5 In the navigation pane, click **Cold and Hot Data Separation**.

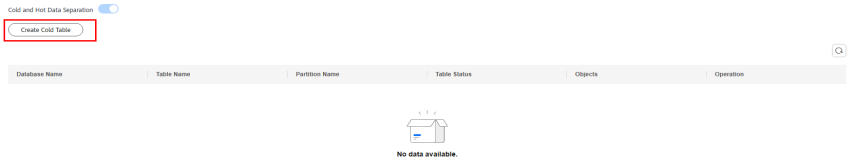
Step 6 Click on the right of **Cold and Hot Data Separation**. In the displayed dialog box, click **OK**.

Figure 19-2 Enabling cold and hot data separation



Step 7 Click **Create Cold Table**.

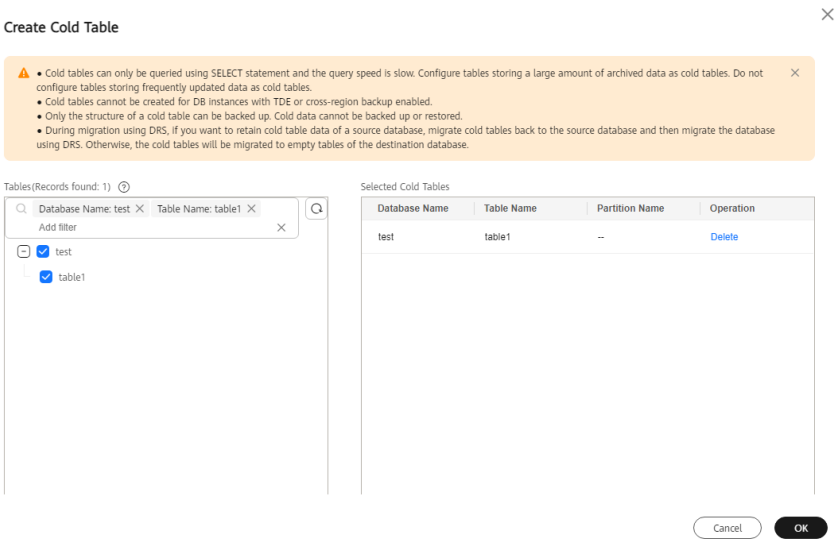
Figure 19-3 Creating a cold table



Step 8 In the displayed dialog box, search for the name of the database, table, or partition to be archived as cold data.

Step 9 Select the tables or partitions to be archived as cold data.

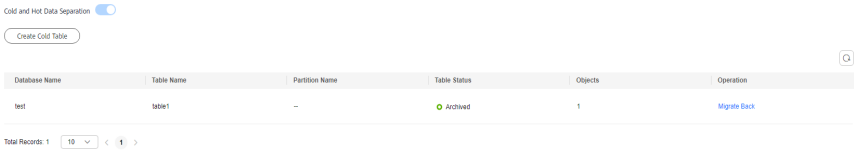
Figure 19-4 Selecting the tables to be archived



Step 10 Click **OK**.

Step 11 After the cold table is created, view its details.

Figure 19-5 Viewing details about a cold table

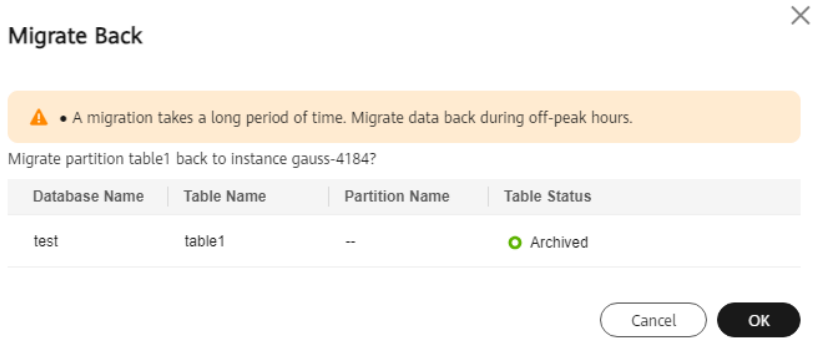


Step 12 If you need to modify a cold table or frequently query the table, click **Migrate Back** in the **Operation** column to migrate the table back to the DB instance.

NOTE

You are advised to migrate back cold tables during off-peak hours because this operation can take a long time.

Figure 19-6 Migrating back a cold table



Confirm the task and click **OK**.

----End

Configuring a Cold Table Using SQL Statements

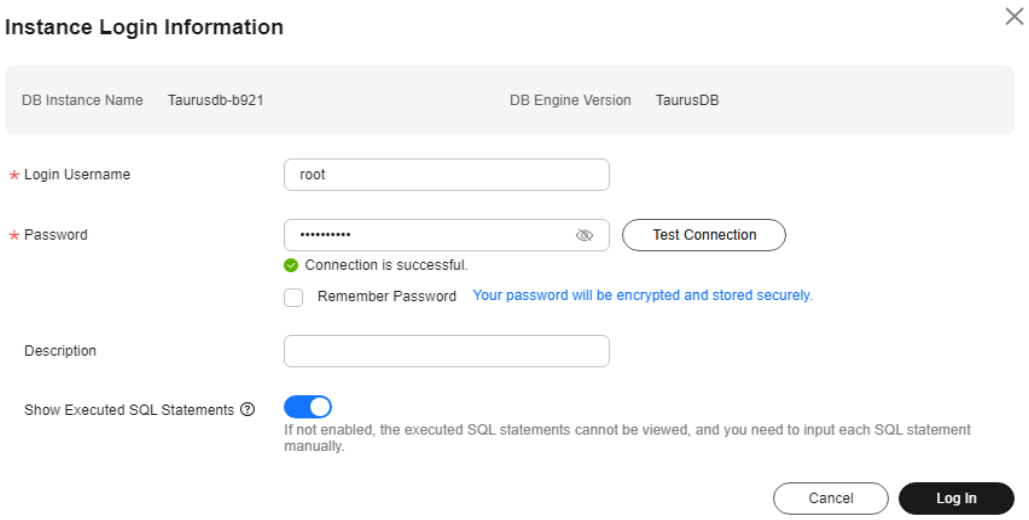
When configuring a cold table using SQL statements, you need to use DAS or a client (such as the mysql client) to connect to your TaurusDB instance and then run the corresponding SQL statements. The following procedure uses DAS as an example.

Step 1 On the **Instances** page, locate an instance and click **Log In** in the **Operation** column.

Figure 19-7 Logging in to an instance

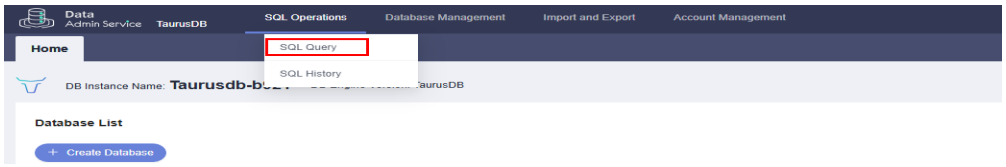
Step 2 On the displayed page, enter the correct username and password and click **Test Connection**. After the connection is successful, click **Log In**.

Figure 19-8 Login page



Step 3 Choose **SQL Operations** > **SQL Query**.

Figure 19-9 SQL Operations



Step 4 Configure a cold table using SQL statements.

- Creating a cold table

  CALL dbms_schs.make_io_transfer("start", "*database_name*", "*table_name*", "*partition_name*", "", "obs");

  Example:

  CALL dbms_schs.make_io_transfer("start", "test", "table1", "", "", "obs");

  Figure 19-10 Creating a cold table

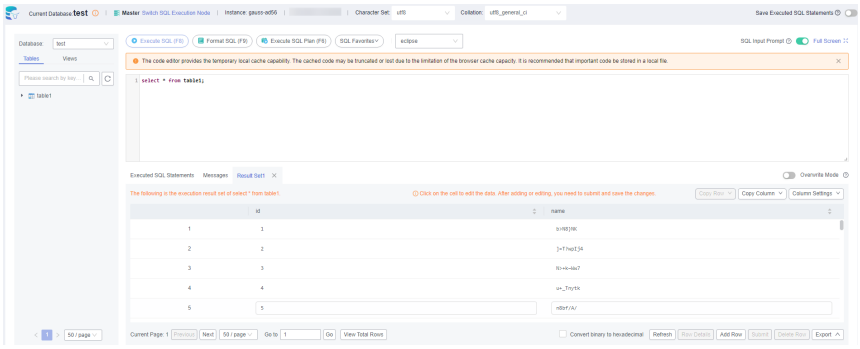
- Querying data from a cold table

  select * from *table_name*;

  Example:

  select * from table1;

Figure 19-11 Querying data from a cold table



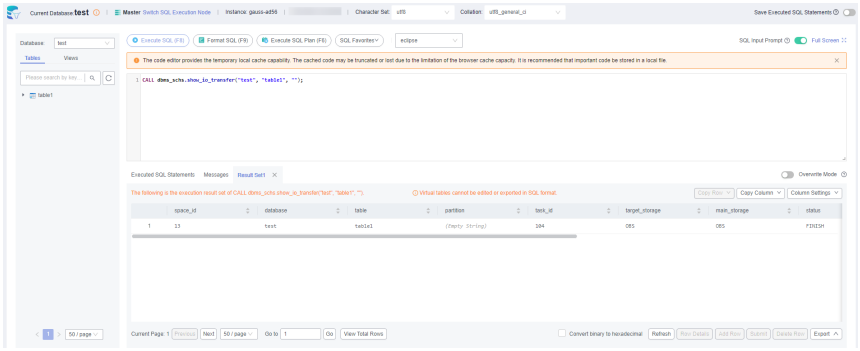
- Querying the archiving or migration status

  CALL dbms_schs.show_io_transfer("*database_name***", "***table_name***", "***partition_name***");**

  Example:

  CALL dbms_schs.show_io_transfer("test", "table1", "");

Figure 19-12 Querying the archiving or migration status



- Migrating back a cold table

  CALL dbms_schs.make_io_transfer("start", "*database_name***", "***table_name***", "***partition_name***", "obs", "");**

  Example:

  CALL dbms_schs.make_io_transfer("start", "test", "table1", "", "obs", "");

Figure 19-13 Migrating back a cold table

- Querying all cold tables on an instance as user **root**

  CALL sys.schs_show_all("*database_name***", "***table_name***", "***partition_name***");**

  Examples:

a. Querying all cold tables on an instance

   CALL sys.schs_show_all("", "", "");

b. Querying all cold partitions or cold tables whose database name is **test**

   CALL sys.schs_show_all("test", "", "");

c. Querying cold partitions or cold tables whose database name is **test** and table name is **table1**

   CALL sys.schs_show_all("test", "table1", "");

----End

20 HTAP Analysis (Standard Edition)

20.1 What Is HTAP of Standard Edition?

Hybrid Transactional and Analytical Processing (HTAP) is a data architecture that handles both online transactional processing (OLTP) and online analytical processing (OLAP) workloads.

It uses the column-based storage engine and Single Instruction Multiple Data (SIMD) for parallel compute. In massive data analysis scenarios, HTAP analysis provided by TaurusDB frees you from having to independently maintain data extraction and synchronization links. It reduces data management costs and provides simple and efficient real-time data analysis.

HTAP of Standard Edition is developed based on the open-source StarRocks.

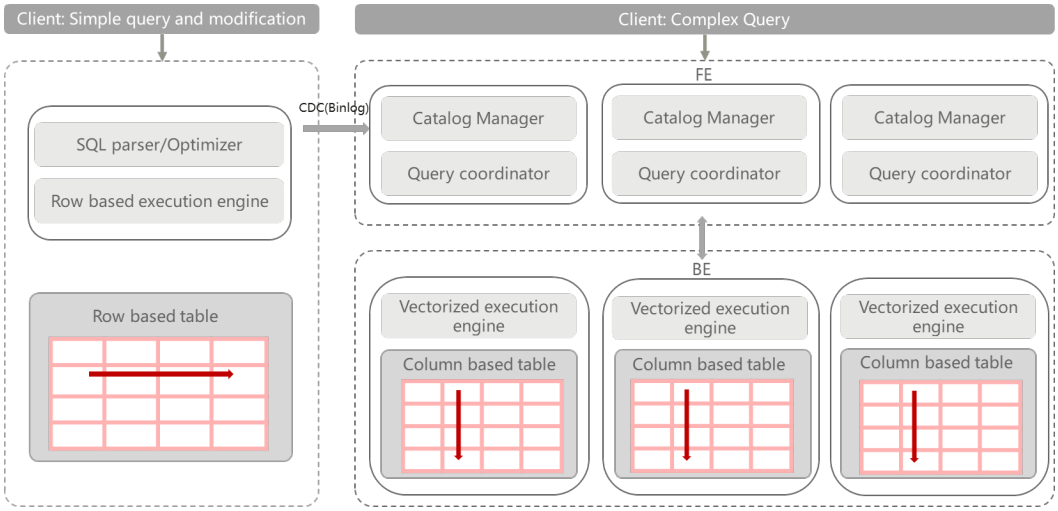
Product Architecture

HTAP instances are deployed on ECSs and use extreme SSDs or ultra-high I/O disks.

You can **enable binlog for your TaurusDB instance** to synchronize data and operations to HTAP instances. Synchronized operations include inserting table, deleting tables, and changing table structures. After data is synchronized to an HTAP instance, you can access the HTAP instance through its private IP address for data analysis.

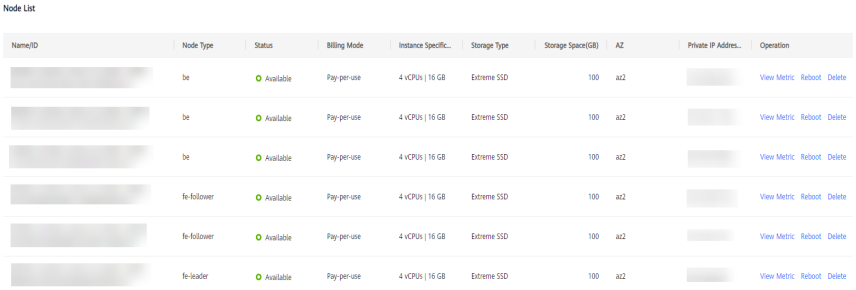
An HTAP instance of the standard edition provides frontend (FE) and backend (BE) nodes. The FE nodes manage metadata, manage client connections, and plan and schedule queries. Each FE node stores and maintains a complete metadata backup in the memory to ensure data consistency between FE nodes. The BE nodes are for data storage and SQL computing.

Figure 20-1 Product architecture



There are three roles for FE nodes.

Figure 20-2 FE node roles



- The fe-leader nodes read and write metadata. The fe-follower and fe-observer nodes can only read metadata and route write requests for metadata to the fe-leader nodes. The fe-leader nodes update the metadata and synchronize the metadata changes to the fe-follower and fe-observer nodes.
- The fe-follower nodes can only read metadata.
- The fe-observer nodes synchronize and replay logs from the fe-leader nodes to update metadata. The fe-observer nodes are used to increase query concurrency of a cluster.

Main Features

- Massively Parallel Processing (MPP) architecture

  Multiple nodes are used to execute queries in parallel.
- High performance

  It supports vectorized engines and CBO optimizers and excels in queries for large and wide tables and multi-table join operations.
- Standard SQL

  Query statements comply with the SQL-92 standard.

- Data compression for storage

  Column-based storage and data compression greatly reduce your storage costs for any given set of conditions.
- Aggregation of multiple data sources

  Data in multiple TaurusDB databases can be synchronized to a given HTAP instance.

Constraints

HTAP of Standard Edition is only available in the following regions:

- AP-Singapore
- AF-Johannesburg

Billing

For details, see **HTAP Instance Billed Items**.

Precautions

- When you query data in an HTAP instance, character string comparison and the names of databases, tables, views, users, and roles are case sensitive, but the names of columns and partitions are case insensitive.
- A Duplicate Key table is used when tables without primary keys in a TaurusDB instance are synchronized to an HTAP instance.
- A primary key value can be up to 128 bytes long.
- Strings must be encoded using UTF8.
- Some DDL statements executed on TaurusDB instances cannot be synchronized to HTAP instances, which may cause synchronization failures or data inconsistencies.

  The DDL statements that can be synchronized and cannot be synchronized are as follows:

  – DDL statements that can be synchronized

Table 20-1 DDL statements that can be synchronized

DDL Name	SQL Example
Creating a database	CREATE DATABASE db_name; **CAUTION** DDL statements for creating a database are only supported in instance-level full synchronization scenarios.
Deleting a database	DROP DATABASE db_name; **CAUTION** DDL statements for deleting a database are only supported in instance-level full synchronization scenarios.

DDL Name	SQL Example
Creating a table	CREATE TABLE tbl_name (c_id int not null, c_d_id integer not null, primary key (c_id)); **CAUTION** When creating a table using CREATE TABLE *t1* LIKE *t2*, ensure that the tables in the statement are in the same synchronization task.
Dropping a table	DROP TABLE tbl_name;
Renaming a table	RENAME TABLE tbl_name to new_tbl_name; ALTER TABLE tbl_name RENAME TO new_tbl_name;
Clearing table data	TRUNCATE TABLE tbl_name;
Altering table comments	ALTER TABLE tbl_name COMMENT='test';
Adding a column (non-primary key column)	ALTER TABLE tbl_name ADD c_varchar varchar(2000) AFTER c_tinytext;
Deleting a column (non-primary key column)	ALTER TABLE tbl_name DROP c_vchar;
Changing the type and sequence of a column (non-primary key column)	ALTER TABLE tbl_name CHANGE c_vchar c_vchar varchar(2000) default 'test' AFTER c_tinytext; **CAUTION** The column name and default value cannot be changed. ALTER TABLE tbl_name MODIFY c_vchar varchar(2100) default 'test' AFTER c_tinytext; **CAUTION** The default value cannot be changed.
Creating a view	CREATE VIEW view_name as select * from tbl_name;
Dropping a view	DROP VIEW view_name;
Altering a view	ALTER VIEW view_name AS select * from tbl_name;

- DDL statements that cannot be synchronized

  After a database synchronization task, only tables and data can be synchronized. Operations for databases, functions, stored procedures, triggers, partitions (DELETE operations), primary keys (INSERT/DELETE/ALTER operations), transactions, users, roles, privileges, and events cannot be synchronized.

Table 20-2 list partitioned table-related operations that cannot be synchronized

Table 20-2 Partitioned table-related operations that cannot be synchronized

DDL Name	SQL Example
Analyzing a table partition	ALTER TABLE {db}.tp ANALYZE PARTITION p0;
Checking a table partition	ALTER TABLE {db}.tp CHECK PARTITION p0;
Optimizing a table partition	ALTER TABLE {db}.tp OPTIMIZE PARTITION p0;
Re-building a table partition	ALTER TABLE {db}.tp REBUILD PARTITION p0;
Repairing a table partition	ALTER TABLE {db}.tp REPAIR PARTITION p0;
Creating a database	CREATE DATABASE ddl_test_2;
Modifying a row format	ALTER TABLE tbl_name ROW_FORMAT = row_format;
Setting persistent table statistics	ALTER TABLE tbl_name STATS_PERSISTENT=0, STATS_SAMPLE_PAGES=20,STATS_AUTO_RECALC =1, ALGORITHM=INPLACE, LOCK=NONE;
Setting a table character set	ALTER TABLE tbl_name CHARACTER SET = charset_name;
Converting a table character set	ALTER TABLE tbl_name CONVERT TO CHARACTER SET charset_name;
Rebuilding a table without data	ALTER TABLE tbl_name ENGINE=InnoDB;
Adding a table partition	ALTER TABLE {db}.tp ADD PARTITION (PARTITION p3 VALUES LESS THAN (2006));
Setting the default character set and verification rules for a table	ALTER TABLE tbl_name DEFAULT CHARACTER SET = utf8 COLLATE = utf8_general_ci;

Table creation statements cannot contain CHECK or table options.

During data synchronization, operations in Table 20-3 may cause data inconsistency between HTAP instances and TaurusDB instances. You should avoid these operations.

They do not affect data query and analysis on HTAP instances.

Table 20-3 DDL operations that result in data inconsistency

DDL Name	SQL Example
Deleting a primary key	ALTER TABLE tbl_name DROP PRIMARY KEY;
Adding a primary key	ALTER TABLE {db}.t1 ADD PRIMARY KEY (id);
Adding a primary key and deleting a primary key	ALTER TABLE tbl_name DROP PRIMARY KEY, ADD PRIMARY KEY (column);
Setting a primary key to NULL	ALTER TABLE tbl_name MODIFY COLUMN key_column_name data_type;
Changing the type of a primary key	ALTER TABLE tbl_name MODIFY COLUMN key_column_name data_type not null;
Adding a column **NOTE** Common columns can be added. If columns contain the following default values, they cannot be added. Functions, character strings, and identifiers that cannot be found in HTAP instances	ALTER TABLE tbl_name ADD COLUMN column_name column_definition c VARCHAR(10) DEFAULT (CONCAT('1', '2'));
Setting the default value of a column **NOTE** If columns contain the following default values, you cannot reset default values for the columns. Functions, character strings, and identifiers that cannot be found in HTAP instances	ALTER TABLE tbl_name ALTER COLUMN col SET DEFAULT literal;
Changing NULL in tables to NOT NULL	ALTER TABLE tbl_name MODIFY COLUMN column_name data_type NOT NULL;
Changing the column name and type at the same time	ALTER TABLE t1 CHANGE b b1 VARCHAR(100);
Changing the name of a column	ALTER TABLE t1 RENAME COLUMN a TO b;

DDL Name	SQL Example
Creating a table without a primary key	ALTER TABLE t1 ADD COLUMN (c2 INT GENERATED ALWAYS AS (c1 + 1)STORED);
Adding a STORED derived column	ALTER TABLE {db}.t1 ADD COLUMN (st2 INT GENERATED ALWAYS AS (c2 + 2)STORED), ALGORITHM=COPY;
Adding a VIRTUAL derived column	ALTER TABLE t1 ADD COLUMN (c2 INT GENERATED ALWAYS AS (c1 + 1)VIRTUAL);
Dropping a table partition	ALTER TABLE {db}.tp DROP PARTITION p4;
Discarding a table partition	ALTER TABLE {db}.tp DISCARD PARTITION p2 TABLESPACE;
Importing a table partition	ALTER TABLE {db}.tp IMPORT PARTITION p2 TABLESPACE;
Truncating a table partition	ALTER TABLE {db}.tp TRUNCATE PARTITION p2;
Truncating a partitioned table	TRUNCATE {db}.tp;
Coalescing table partitions	ALTER TABLE {db}.tp_hash COALESCE PARTITION 2;
Reorganizing table partitions	ALTER TABLE {db}.tp REORGANIZE PARTITION p0,p1,p2,p3 INTO (...);
Exchanging table partitions	ALTER TABLE {db}.tp EXCHANGE PARTITION p0 WITH TABLE {db}.tp2;
Removing a table partition	ALTER TABLE {db}.tp REMOVE PARTITIONING;
Using a REPLACE clause	CREATE OR REPLACE TABLE;
Renaming a view	RENAME TABLE old_view_name TO new_view_name;

Table 20-4 DDL operations that have been ignored during synchronization (no impacts)

DDL Name	SQL Example
Adding an index	ALTER TABLE tbl_name ADD INDEX name;
Renaming an index	ALTER TABLE tbl_name RENAME INDEX old_index_name TO new_index_name;

DDL Name	SQL Example
Dropping an index	DROP INDEX name ON table;
Adding a full-text index	CREATE FULLTEXT INDEX name ON table(column);
Adding a spatial index	ALTER TABLE geom ADD SPATIAL INDEX(g);
Modifying the type of an index	ALTER TABLE tbl_name DROP INDEX i1, ADD INDEX i1(key_part,...) USING BTREE;
Adding an index constraint	ALTER TABLE tbl_name ADD CONSTRAINT UNIQUE USING BTREE (column); ALTER TABLE tbl_name ADD CONSTRAINT UNIQUE USING HASH(column);
Optimizing a table	OPTIMIZE TABLE tbl_name;
Rebuilding a table using the FORCE option	ALTER TABLE tbl_name FORCE;
Renaming a tablespace	ALTER TABLESPACE tablespace_name RENAME TO new_tablespace_name;
Adding a foreign key	ALTER TABLE tbl1 ADD CONSTRAINT fk_name FOREIGN KEY index (col1)REFERENCES tbl2(col2) referential_actions;
Deleting a foreign key	ALTER TABLE tbl DROP FOREIGN KEY fk_name;

- The names of the databases and tables to be synchronized cannot contain Chinese characters.
- To improve performance, you can use the following methods to optimize queries:
  - Simplify SQL statements by reducing invalid calculations, deleting unused fields, and avoiding SELECT.
  - Instead of querying all columns, delete those that are unnecessary.
- Tables to be synchronized use the OLAP engine and primary key model by default.
- The column names of tables to be synchronized can contain the following characters:
  - Letters (A–Z and a–z)
  - Digits (0–9)
  - Underscores (_)
- Here are descriptions and restrictions on view synchronization:
  - By default, views are not synchronized. If the **sync_view** parameter is set to **true**, data and views are synchronized. If the **sync_view** parameter is set to **only_sync_view**, only views are synchronized.

  - You can only create a view synchronization task after creating a data synchronization task.
  - View synchronization can be enabled for only one synchronization task of a given database.
  - When you select a synchronization view, the source database name must be the same as the destination database name.
  - Some view-related DDL statements cannot be synchronized, for example, the **rename table** statement used to rename a view.
  - A cross-database view may fail to be synchronized if dependent databases are not synchronized.
  - If a function or syntax that is not supported by HTAP instances is used in a view, the view cannot be synchronized.
  - If a view fails to be synchronized, you can manually create it on an HTAP instance after the fault is rectified.
  - When view synchronization fails, alarms and error information are reported. You can connect to an HTAP instance and run **show sync job** to view **SyncErrViews** (views that failed to be synchronized) and **SyncErrMsg** (detailed error information). To clear alarms and error information, you can run **alter sync** *synchronization_task_name* **setting "SyncErrViewMsg" = "", "SyncErrViews"="";**
  - View synchronization failures do not affect table data synchronization.
  - The kernel version of your TaurusDB instance must be 2.0.57.240900 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**

- After connecting to a standard HTAP instance, run the following command to view the databases synchronized from TaurusDB to the standard HTAP instance and the synchronization status.

  As this command queries the binlog information in TaurusDB, it may consume the hourly query quota. You are advised to run this command at most once a minute.

```
show sync job;
```

20.2 Connecting to an HTAP Instance for Complex OLAP Queries

You can let an application directly connect to an HTAP instance to enable complex OLAP queries.

Operation Process

Step 1: Buy a Standard HTAP Instance

Step 2: Synchronize TaurusDB Instance Data to the Standard HTAP Instance

Step 3: Connect to the HTAP Instance for OLAP Queries

Prerequisites

- Parameters have been configured for a TaurusDB instance according to the following table.

Table 20-5 Parameter description

Parameter	Value	How to Modify
default_authentication_plugin	mysql_native_password	**Modifying Parameters of a DB Instance**
binlog_expire_logs_seconds	86400 **NOTE** It is recommended that the binlog retention period be greater than one day. 86,400s = 60 (seconds) x 60 (minutes) x 24 (hours). This prevents incremental replication failures caused by a short binlog retention period.	**Modifying Parameters of a DB Instance**
log_bin **NOTE** To use this parameter, ensure that the TaurusDB kernel version is earlier than 2.0.45.230900. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**	ON	**How Do I Enable and View Binlog of My TaurusDB Instance?**
rds_global_sql_log_bin **NOTE** To use this parameter, ensure that the TaurusDB kernel version is 2.0.45.230900 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**	ON	**How Do I Enable and View Binlog of My TaurusDB Instance?**
binlog_format	ROW	Run the **SHOW VARIABLES;** command to check the parameter value. If you need to change the parameter value, **submit a service ticket**.

Parameter	Value	How to Modify
binlog_row_image	FULL	Run the **SHOW VARIABLES;** command to check the parameter value. If you need to change the parameter value, **submit a service ticket**.
log_bin_use_v1_row_events	OFF	Run the **SHOW VARIABLES;** command to check the parameter value. If you need to change the parameter value, **submit a service ticket**.

- Databases and tables have been created for the TaurusDB instance.

Step 1: Buy a Standard HTAP Instance

1. **Log in to the management console.**
2. Click in the upper left corner and select a region and project.
3. Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.
4. On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.
5. In the navigation pane, choose **HTAP Analysis**. On the displayed page, click **Create HTAP Instance**.
6. In the **DB Instance Information** area, check the current TaurusDB instance information.

   Figure 20-3 Checking TaurusDB instance information

7. Set parameters for the standard HTAP instance.

Figure 20-4 Setting parameters for the standard HTAP instance

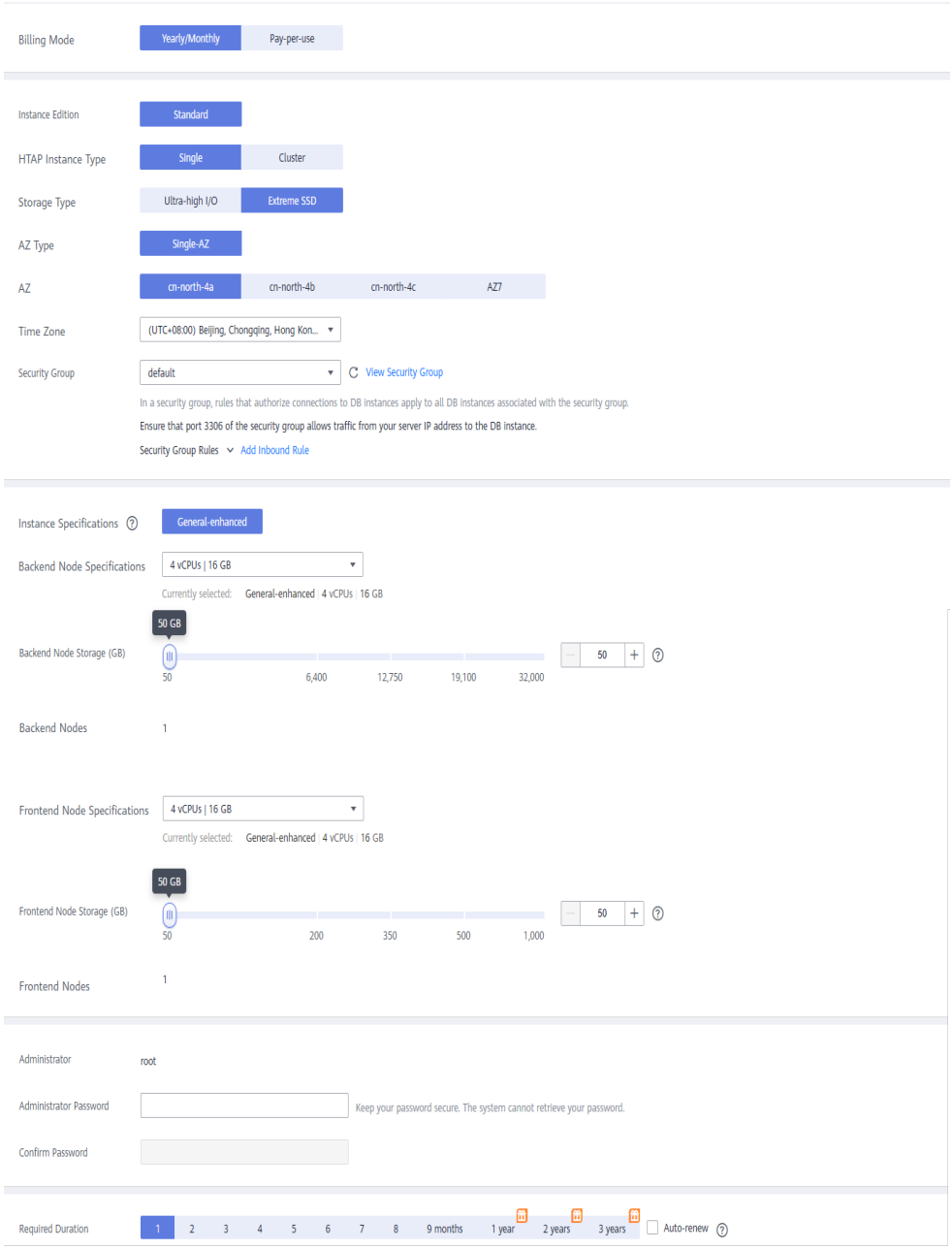


Table 20-6 Parameter description

Parameter	Description
Billing Mode	Select **Pay-per-use** or **Yearly/Monthly**.

Parameter	Description
HTAP Instance Type	Select **Single** or **Cluster**. • **Single**: There is only one FE node and one BE node. It is used only for function experience and testing and does not ensure **SLA**. • **Cluster**: There are at least three FE or BE nodes and at most 10 FE or BE nodes.
Storage Type	Select **Extreme SSD** or **Ultra-high I/O**. • **Extreme SSD**: uses a 25GE network and RDMA to provide you with up to 1 million random read/write performance per disk and low latency per channel. • **Ultra-high I/O**: uses multi-disk striping to balance I/O loads among multiple disks, improving read/write bandwidth. The maximum throughput is 1.7 GB/s.
AZ Type	Only single-AZ is available.
AZ	Select an AZ as needed.
Time Zone	Select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.
Security Group	The default value is the security group of the TaurusDB instance. You are advised to keep the security group consistent with that of the TaurusDB instance.
Instance Specifications	Only general-enhanced is available.
Backend Node Specifications	Select the BE node specifications. The BE nodes are for data storage and SQL computing.
Backend Node Storage (GB)	Select the storage for BE nodes. The default storage is 50 GB and can be expanded to up to 32,000 GB.
Backend Nodes	• A single-node instance has only one BE node. • A cluster instance has 3 to 10 BE nodes. You can apply for a maximum of 10 nodes at a time.
Frontend Node Specifications	Select the FE node specifications. The FE nodes manage metadata, manage client connections, and plan and schedule queries.
Frontend Node Storage (GB)	Select the storage for FE nodes. The default storage is 50 GB and can be expanded to up to 1,000 GB.
Frontend Nodes	• A single-node instance has only one FE node. • A cluster instance has 3 to 10 FE nodes. You can apply for a maximum of 10 nodes at a time.

Parameter	Description
Administrator	The default username is **root**.
Administrator Password	The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,()&$|.). Enter a strong password and periodically change it to improve security and defend against threats such as brute force cracking attempts.
Confirm Password	Enter the administrator password again.
Required Duration	This parameter is only available for yearly/monthly instances. The system will automatically calculate the fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	• This parameter is only available for yearly/monthly instances and is not selected by default. • If you select this parameter, the auto-renew cycle is determined by the selected required duration.

8. Click **Next** in the lower right corner.
   - For pay-per-use instances, go to **9**.
   - For yearly/monthly instances, go to **10**.
9. Confirm the information and click **Submit**.

   To modify the instance information, click **Previous**.
10. Confirm the order.
    - To modify the instance information, click **Previous**.
    - If you do not need to modify your settings, click **Pay Now**. On the order page, complete the payment.
11. On the HTAP instance list page, view and manage the HTAP instance.

Step 2: Synchronize TaurusDB Instance Data to the Standard HTAP Instance

1. On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.
2. In the navigation pane, choose **HTAP Analysis**.
3. Click the name of an HTAP instance to access the **Basic Information** page.
4. In the navigation pane, choose **Data Synchronization**. On the displayed page, click **Create Synchronization Task**.
5. Configure required parameters.

Figure 20-5 Creating a synchronization task

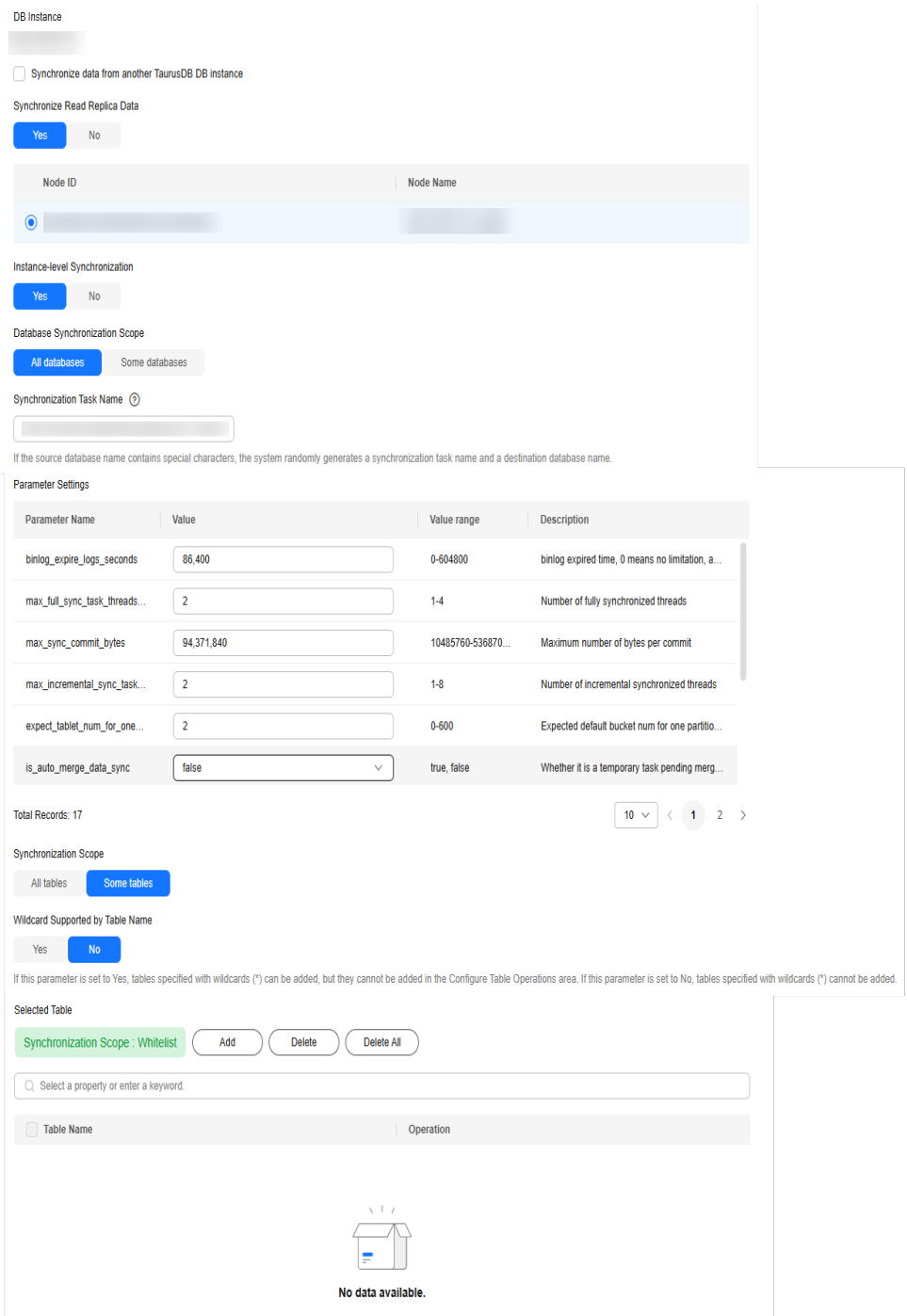


Table 20-7 Parameter description

Parameter	Description
Synchronize Read Replica Data	Controls whether to synchronize full data from a read replica. During a full synchronization, ensure that the read replica is available, or the synchronization will fail and you will need to perform the synchronization again. • **Yes**: Full data is synchronized from the selected read replica, preventing query load on the primary node during a full synchronization. If there is only one read replica, this node is selected by default. • **No**: No synchronization is performed.
Instance-level Synchronization	Controls the synchronization of one or multiple databases. This parameter is in the OBT phase. To use this parameter, **submit a service ticket**. • **Yes**: A synchronization task can synchronize multiple or all databases. • **No**: A synchronization task can synchronize only one database.
Database Synchronization Scope	This parameter is only displayed when **Instance-level Synchronization** is set to **Yes**. • **All databases**: All databases are synchronized by default. You do not need to specify any database name. • **Some databases**: You need to specify two or more database names.
Synchronization Task Name	The name can contain 3 to 128 characters. Only letters, digits, underscores (_) are allowed.
Destination Database	The name can contain 3 to 128 characters. Only letters, digits, underscores (_) are allowed. This parameter is not displayed when **Database Synchronization Scope** is set to **All databases**. When Assign Requests to Row and Column Store Nodes is enabled, the source database name must be the same as the destination database name.
Database to be Synchronized	Select a database that the data will be synchronized to from the drop-down list. You can modify the database parameters of the HTAP instance as required. The drop-down list is hidden when **Database Synchronization Scope** is set to **All databases**.
Synchronization Scope	Select **All Tables** or **Some Tables**.

Parameter	Description
Wildcard Supported by Table Name	This parameter is only displayed when **Instance-level Synchronization** is set to **Yes** and **Synchronization Scope** is set to **Some tables**. In an instance-level synchronization scenario, you can determine whether table names in the blacklist or whitelist support wildcards * and ?. The wildcard * matches zero or more characters, and the wildcard ? matches exactly one character.
Synchronization Scope: Whitelist	If **Synchronization Scope** is set to **Some Tables**, you need to configure tables for the blacklist or whitelist. • You can set either a blacklist or a whitelist. If you select the whitelist, only the tables in the whitelist are synchronized. If you select the blacklist, the tables in the blacklist are not synchronized. • The tables to be synchronized must contain primary keys or a non-empty unique key, or they cannot be synchronized to the HTAP instance. • Extra disk space may be used during backend data combination and query. You are advised to reserve 50% of the disk space for the system. • When setting the table blacklist or whitelist, you can enter multiple tables in the search box at a time. The tables can be separated by commas (,), spaces, or line breaks (\n). After entering multiple tables, you need to click  . These tables will be selected by default and displayed in the **Selected Table** area.
Configure Table Operations	Enable or disable it as required. • **Enabled**: Select a synchronized table on the left and perform operations on its columns. The operations include order by, key columns, distributed by, partition by, data_model, buckets, replication_num, and enable_persistent_index. Multiple operations are separated by semicolons (;). For details about the syntax, see **Table 20-8**. • **Disabled**: No operations are required.

Table 20-8 Operation syntax

Operation Type	Syntax
order by	order by (column1, column2) or order by column1,column2
key columns	key columns (column1, column2) or key columns column1,column2

Operation Type	Syntax
distributed by	distributed by (column1, column2) buckets 3 **NOTE** **buckets** is optional. If it is not set, the default value is used.
partition by	There are expression partitions and list partitions. For details, see the partition syntax example.
data_model	Specifies the table type. The value can be **primary key**, **duplicate key**, or **unique key**. Syntax: data_model=primary key, data_model=duplicate key, or data_model=unique key
replication_num	replication_num=3 **NOTE** The value cannot exceed the number of BE nodes, or the verification fails.
enable_persistent_index	Specifies whether to make the index persistent. Syntax: enable_persistent_index=true or enable_persistent_index=false
Combined scenario	data_model=duplicate key;key columns column1, column2;

Partition syntax example:

You only need to set a partition expression (time function expression or column expression) when creating a table. During data import, an HTAP instance automatically creates partitions based on the data and the rule defined in the partition expression.

Partitioning based on a time function expression: If data is often queried and managed based on a continuous date range, you only need to specify a partition column of the date type (DATE or DATETIME) and a partition granularity (year, month, day, or hour) in the time function expression. An HTAP instance automatically creates partitions and sets the start time and end time of the partitions based on the imported data and partition expression.

Syntax:

```
PARTITION BY expression
...
[ PROPERTIES( 'partition_live_number' = 'xxx' ) ]

expression ::=
    { date_trunc ( <time_unit> , <partition_column> ) |
      time_slice ( <partition_column> , INTERVAL <N> <time_unit> [ , boundary ] ) }
```


Table 20-9 Parameter description

Parameter	Mandatory	Description
expression	Yes	Currently, only the **date_trunc** and **time_slice** functions are supported. If you use **time_slice**, you do not need to configure the **boundary** parameter because this parameter can only be set to **floor** by default. It cannot be set to **ceil**.
time_unit	Yes	Partition granularity. Currently, the value can only be **hour**, **day**, **month**, or **year**. It cannot be **week**. If the partition granularity is **hour**, the partition columns can only be of the DATETIME type. They cannot be of the DATE type.
partition_column	Yes	Partition column. • Only the date type (DATE or DATETIME) is supported. If **date_trunc** is used, the partition column can be of the DATE or DATETIME type. If **time_slice** is used, the partition column can only be of the DATETIME type. The value of the partition column can be **NULL**. • If the partition column is of the DATE type, the value range is from **0000-01-01** to **9999-12-31**. If the partition column is of the DATETIME type, the value range is from **0000-01-01 01:01:01** to **9999-12-31 23:59:59**. • Currently, only one partition column can be specified.

Example: If you often query data by day, you can use the partition expression **date_trunc ()**, set the partition column to **event_day**, and set the partition granularity to **day** during table creation. In this way, data is automatically partitioned based on dates when being imported. Data of the same day is stored in the same partition. Partition pruning can significantly improve queries.

```
CREATE TABLE site_access1 (
    event_day DATETIME NOT NULL,
    site_id INT DEFAULT '10',
    city_code VARCHAR(100),
    user_name VARCHAR(32) DEFAULT '',
    pv BIGINT DEFAULT '0'
)
DUPLICATE KEY(event_day, site_id, city_code, user_name)
PARTITION BY date_trunc('day', event_day)
DISTRIBUTED BY HASH(event_day, site_id);
```


Partitioning based on the column expression: If you often query and manage data based on enumerated values, you only need to specify the column representing the type as the partition column. An HTAP instance automatically divides and creates partitions based on the partition column value of the imported data.

Syntax:

```
PARTITION BY expression
...
[ PROPERTIES( 'partition_live_number' = 'xxx' ) ]

expression ::=
    ( <partition_columns> )

partition_columns ::=
    <column>, [ <column> [,...] ]
```

Table 20-10 Parameter description

Parameter	Mandatory	Description
partition_columns	Yes	Partition columns. • The value can be a Character (BINARY is not supported), Date, Integer, or Boolean value. The value cannot be **NULL**. • After the import, a partition automatically created can contain only one value of each partition column. If multiple values of each partition column need to be contained, use list partitioning.

Example: If you often query the equipment room billing details by date range and city, you can use a partition expression to specify the date and city as the partition columns when creating a table. In this way, data of the same date and city is grouped into the same partition, and partition pruning can be used to significantly accelerate queries.

```
CREATE TABLE t_recharge_detail1 (
    id bigint,
    user_id bigint,
    recharge_money decimal(32,2),
    city varchar(20) not null,
    dt varchar(20) not null
)
DUPLICATE KEY(id)
PARTITION BY (dt,city)
DISTRIBUTED BY HASH(`id`);
```

List partitioning

Data is partitioned based on a list of enumerated values that you explicitly define. You need to explicitly list the enumerated values contained in each list partition, and the values do not need to be consecutive.

List partitioning is suitable for storing columns where there are a small number of enumerated values and querying and managing data based on the enumerated values. For example, a column indicates a geographical location, status, or category. Each value of a column represents an independent category. Data is partitioned based on the enumerated values of columns to improve query performance and data management. List partitioning is especially suitable for scenarios where a partition needs to contain multiple values of each partition column. For example, the **city** column in a table indicates the city that an individual is from, and you often query and manage data by state and city. You can use the **city** column as the partition column for list partitioning when creating a table, and specify that data of multiple cities in the same state is stored in the same partition **PARTITION pCalifornia VALUES IN ("Los Angeles","San Francisco","San Diego")**, this feature accelerates queries and data management.

NOTE

Partitions must be created during table creation. Partitions cannot be automatically created during data import. If the table does not contain the partitions corresponding to the data, an error is reported.

Syntax:

```
PARTITION BY LIST (partition_columns)(
    PARTITION <partition_name> VALUES IN (value_list)
    [, ...]
)

partition_columns::=
    <column> [,<column> [, ...] ]

value_list ::=
    value_item [, value_item [, ...] ]

value_item ::=
    { <value> | ( <value> [, <value>, [, ...] ] ) }
```

Table 20-11 Parameter description

Parameter	Mandatory	Description
partition_columns	Yes	Partition columns. The value can be a Character (except BINARY), Date (DATE and DATETIME), Integer, or Boolean value. The value cannot be **NULL**.
partition_name	Yes	Partition name. You are advised to set proper partition names to distinguish data categories in different partitions.
value_list	Yes	List of enumerated values of partition columns in a partition.

Example 1: If you often query the equipment room billing details by state or city, you can specify the **city** column as the partition column and specify that the cities in each partition belong to the same state. In this way, you can quickly query data of a specific state or city and manage data by state or city.

```
CREATE TABLE t_recharge_detail2 (
    id bigint,
    user_id bigint,
    recharge_money decimal(32,2),
    city varchar(20) not null,
    dt varchar(20) not null
)
DUPLICATE KEY(id)
PARTITION BY LIST (city) (
PARTITION pCalifornia VALUES IN ("Los Angeles","San Francisco","San Diego"), --: These cities belong to the same state.
   PARTITION pTexas VALUES IN ("Houston","Dallas","Austin")
)
DISTRIBUTED BY HASH(`id`);
```

Example 2: If you often query the equipment room billing details by date range and state or city, you can specify the date and city as the partition columns when creating a table. In this way, data of a specific date and a specific state or city is grouped into the same partition, to accelerate queries and data management.

```
CREATE TABLE t_recharge_detail4 (
    id bigint,
    user_id bigint,
    recharge_money decimal(32,2),
    city varchar(20) not null,
    dt varchar(20) not null
) ENGINE=OLAP
DUPLICATE KEY(id)
PARTITION BY LIST (dt,city) (
   PARTITION p202204_California VALUES IN (
      ("2022-04-01", "Los Angeles"),
      ("2022-04-01", "San Francisco"),
      ("2022-04-02", "Los Angeles"),
      ("2022-04-02", "San Francisco")
    ),
   PARTITION p202204_Texas VALUES IN (
      ("2022-04-01", "Houston"),
      ("2022-04-01", "Dallas"),
      ("2022-04-02", "Houston"),
      ("2022-04-02", "Dallas")
   )
)
DISTRIBUTED BY HASH(`id`);
```

6. After the settings are complete, click **Create Synchronization Task**.
7. Click **Back to Synchronization List** to return to the data synchronization page. A synchronization task to be synchronized is generated. The task status is **Synchronization Stage: Waiting for synchronization**. To start the task, click **Synchronize** in the **Operation** column.

   If the task status changes to **Synchronization Stage: Incremental synchronization in progress**, the data synchronization is complete.

Figure 20-6 Viewing the task status

NOTE

During the full synchronization, if some tables fail to be synchronized, an alarm will be generated and those tables will be skipped. The remaining tables will continue to be synchronized. After the full synchronization is complete, the incremental synchronization starts. You can **repair tables that failed to be synchronized** during the incremental synchronization.

During service tests, if you want to suspend a synchronization task, click **Stop**. The suspension duration cannot exceed the **binlog retention period** set for the source primary TaurusDB instance. If the suspension duration exceeds the binlog retention period, the synchronization task cannot continue. You need to delete the task and create a new one. Do not suspend synchronization tasks in the production environment to prevent data inconsistency between OLTP and OLAP.

Step 3: Connect to the HTAP Instance for OLAP Queries

For details about how to connect to a standard HTAP instance and perform OLAP queries, see **Connecting to a Standard HTAP Instance Through JDBC**.

20.3 Connecting to a Standard HTAP Instance

20.3.1 Connecting to a Standard HTAP Instance Through DAS

Data Admin Service (DAS) enables you to manage DB instances on a web-based console, simplifying database management and improving working efficiency.

By default, you have remote login permissions. It is recommended that you use DAS to connect to HTAP instances because this connection method is more secure and convenient than other methods.

Procedure

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

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

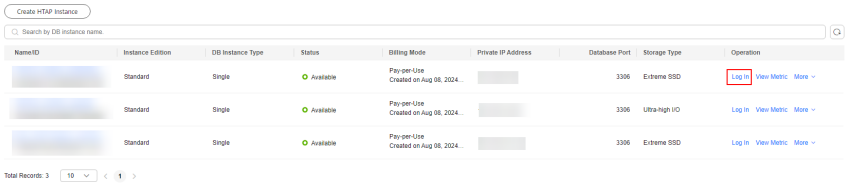
Step 3 Click in the upper left corner of the page, choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 In the instance list, locate an HTAP instance and click **Log In** in the **Operation** column.

Figure 20-7 Logging in to a standard HTAP instance



Step 7 Select the node to be logged in to, enter the database username and password, and click **Test Connection**.

Step 8 After the connection test is successful, click **Log In** to access your database.

----End

20.3.2 Connecting to a Standard HTAP Instance Through JDBC

You can connect to a standard HTAP instance through JDBC.

Precautions

Currently, HTAP instances only support the UTF-8 character set.

Prerequisites

- You are familiar with:
  - Computer basics
  - Java
  - JDBC knowledge
- You have downloaded the official JDBC driver for MySQL or MariaDB.
- You have created a standard HTAP instance.
- The following dependency has been added to the **pom.xml** file.

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.47</version>
</dependency>
```

- You can use the following command to connect to an HTAP instance through JDBC:

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

Parameter	Description
<instance_ip>	IP address of the FE node in the HTAP instance. If a proxy is installed, use the IP address of the proxy.
<instance_port>	HTAP instance port. The default value is 3306.
<database_name>	Database name used for connecting to the instance.

Sample Code

Code example (Java code for connecting to an HTAP database):

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
public class JDBCTest {
    static final String IP = "*.*.*.*"; //IP address of the instance
    static final String USER = "***"; //Username
    static final String PASS = "***"; //Password
    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        String url = "jdbc:mysql://" + IP + ":3306";
        try {

            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(url, USER, PASS);
            stmt = conn.createStatement();
            String sql = "show databases;";
            ResultSet rs = stmt.executeQuery(sql);
            int columns = rs.getMetaData().getColumnCount();
            for (int i = 1; i <= columns; i++) {
                System.out.print(rs.getMetaData().getColumnName(i));
                System.out.print("\t");
            }
            while (rs.next()) {
                System.out.println();
                for (int i = 1; i <= columns; i++) {
                    System.out.print(rs.getObject(i));
                    System.out.print("\t");
                }
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // release resource ....
        }
    }
}
```

20.4 Standard HTAP Instance Management

20.4.1 Rebooting a Standard HTAP Instance

Scenarios

You may need to reboot an HTAP instance for maintenance reasons.

Constraints

- You can reboot an HTAP instance only when it is available or abnormal. When some operations such as creating a task, changing specifications, scaling up storage, and upgrading a minor version, are being performed on an HTAP instance, the instance cannot be rebooted.
- It takes about 1 to 2 minutes to reboot an HTAP instance. During the reboot, the instance is unavailable. Rebooting an HTAP instance will clear its cache. To prevent traffic congestion during peak hours, you are advised to reboot it during off-peak hours.

Procedure

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

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

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

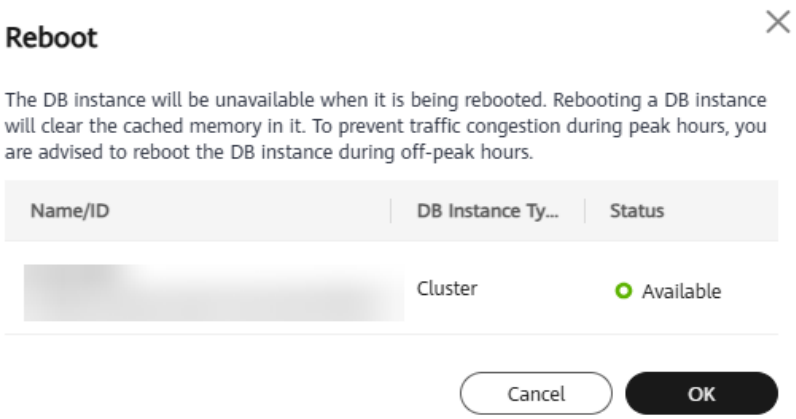
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and choose **More** > **Reboot** in the **Operation** column.

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

Figure 20-8 Rebooting an HTAP instance



Step 8 Check that the standard HTAP instance status changes from **Rebooting** to **Available**.

----End

20.4.2 Rebooting a Node of a Standard HTAP Instance

Scenarios

You may need to reboot a node of an HTAP instance for maintenance reasons.

Constraints

- You can reboot a node only when it is available or abnormal. When some operations such as creating a task, changing specifications, scaling up storage, and upgrading a minor version, are being performed on a node, the node cannot be rebooted.
- It takes about 1 to 2 minutes to reboot a node of an HTAP instance. During the reboot, the instance is unavailable. Rebooting a node will clear its cache. To prevent traffic congestion during peak hours, you are advised to reboot the node during off-peak hours.

Procedure

Step 1 Log in to the management console.

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

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

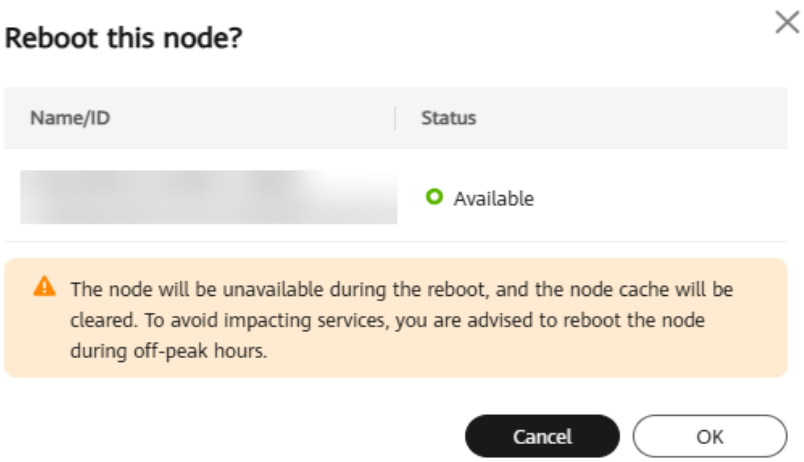
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**. Locate an HTAP instance and click its name to access the **Basic Information** page.

Step 6 Locate an HTAP node and click **Reboot** in the **Operation** column.

Step 7 In the displayed dialog box, click **OK** to reboot the node. It takes about 1 to 2 minutes.

Figure 20-9 Rebooting a node



Step 8 Check that the standard HTAP instance status changes from **Rebooting node** to **Available**.

----End

20.4.3 Deleting a Pay-per-Use Standard HTAP Instance

Scenarios

You can delete any unused pay-per-use HTAP instances on the **HTAP Analysis** page.

To delete a yearly/monthly HTAP instance, you need to unsubscribe the order. For details, see **Unsubscribing from a Yearly/Monthly Standard HTAP Instance**.

Precautions

- HTAP instances cannot be deleted when operations are being performed on them.
- Deleted HTAP instances cannot be recovered. Exercise caution when performing this operation.

Procedure

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

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

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

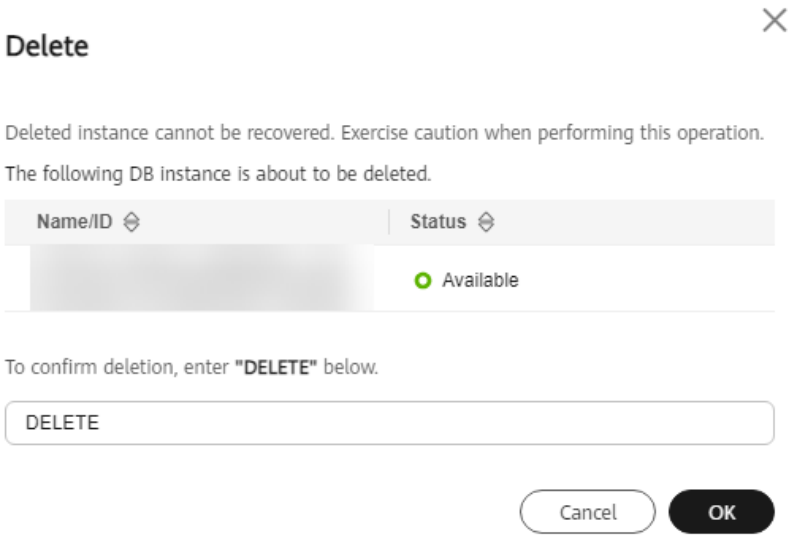
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click **Delete** in the **Operation** column.

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

Figure 20-10 Deleting an HTAP instance



----End

20.4.4 Unsubscribing from a Yearly/Monthly Standard HTAP Instance

Scenarios

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

Constraints

- An instance cannot be unsubscribed when any operations are being performed on it. It can be unsubscribed only after the operations are complete.
- If a backup of an instance is being restored, the instance cannot be unsubscribed.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

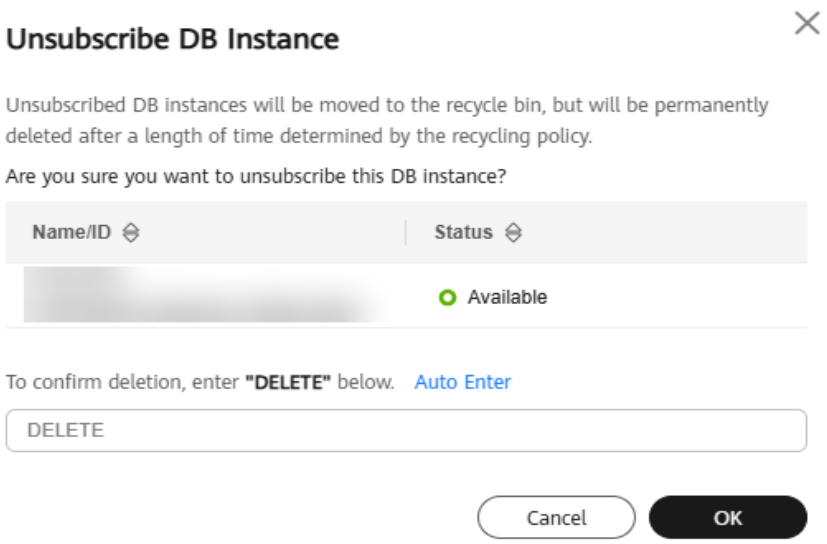
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and choose **More** > **Unsubscribe** in the **Operation** column.

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

Figure 20-11 Unsubscribing from an HTAP instance



Step 8 On the page for unsubscribing from resources, confirm the refund information and click **Confirm**.

Step 9 After you unsubscribe from an instance order, the instance will be deleted. Check that it is no longer displayed in the HTAP instance list.

----End

20.5 Standard HTAP Instance Configuration Changes

20.5.1 Changing the Nodes Specifications of a Standard HTAP Instance

Scenarios

After a standard HTAP instance is created, you can change the specifications of BE and FE nodes separately or simultaneously. The specifications of pay-per-use instances can be downgraded, and those of yearly/monthly instances can only be upgraded.

Constraints

- You cannot reboot or delete the HTAP instance while its specifications are being changed.
- You can change the specifications of your HTAP instance for an unlimited number of times.

Billing

Table 20-12 Billing

Billing Mode	Operation	Impact on Price
Yearly/ Monthly	Specification upgrade	After instance specifications are upgraded, the new instance specifications take effect in the original usage period. You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month single standard HTAP instance with one BE and three FE nodes (specifications: 4 vCPUs | 16 GB; storage: extreme SSD, 50 GB) in CN-Hong Kong on April 1, 2025. The instance price was $624.80 USD per month. On April 15, 2025, you changed the BE node specifications to 8 vCPUs | 32 GB. The instance price was $914.80 USD per month. **Price difference = Price for the new instance specifications x Remaining period – Price for the original instance specifications x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Price difference = $914.80 USD x 0.5 – $624.80 USD x 0.5 = $145 USD
Pay-per-use	Specification upgrade	After instance specifications are changed, the new instance specifications are billed by hour. For details, see **Product Pricing Details**.
	Specification downgrade	

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate the target HTAP instance and choose **More** > **Change Instance Specifications** in the **Operation** column.

Step 7 On the displayed page, change the node specifications.

Table 20-13 Parameter description

Parameter	Description
Node Type	• Yearly/Monthly: You can only select either **BE** or **FE**. • Pay-per-use: You can select either **BE** or **FE**, or select both **BE** and **FE**.
Backend Node Specifications	If **BE** is selected for **Node Type**, the original specifications and type of BE nodes are displayed.
Backend Node Specifications	If **BE** is selected for **Node Type**, set the new specifications of BE nodes.
Frontend Node Specifications	If **FE** is selected for **Node Type**, the original specifications and type of FE nodes are displayed.
Frontend Node Specifications	If **FE** is selected for **Node Type**, set the new specifications of FE nodes.

Step 8 Click **Next**.

Step 9 Confirm settings.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for a pay-per-use or yearly/monthly instance.

Step 10 After the specifications are changed, view and manage them on the **Basic Information** page of the HTAP instance.

----End

20.5.2 Changing Storage Space of a Standard HTAP Instance

Scenarios

After creating a standard HTAP instance, you can change storage space of BE and FE nodes as required.

Constraints

- You cannot reboot or delete the HTAP instance while its storage space is being changed.

- You can change storage space of an HTAP instance multiple times.
- You can change storage space of BE and FE nodes separately or simultaneously.

Billing

Table 20-14 Billing

Billing Mode	Operation	Impact on Price
Yearly/ Monthly	Storage scale-up	You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month single standard HTAP instance with one BE and three FE nodes (specifications: 4 vCPUs | 16 GB; storage: extreme SSD, 50 GB) in CN-Hong Kong on April 1, 2025. The unit price of storage space is $0.448 USD/GB per month. On April 15, 2024, you scaled up the storage space of BE nodes by 20 GB. The total storage space after the scale-up was 70 GB. **Price difference = Scale-up volume x Unit price x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Price difference = 20 GB x $0.448 USD/GB x 0.5 = $4.48 USD
Pay-per-use	Storage scale-up	The new storage space is billed by hour. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and choose **More** > **Change Storage Space** in the **Operation** column.

Step 7 On the displayed page, set required parameters.

Figure 20-12 Changing storage space

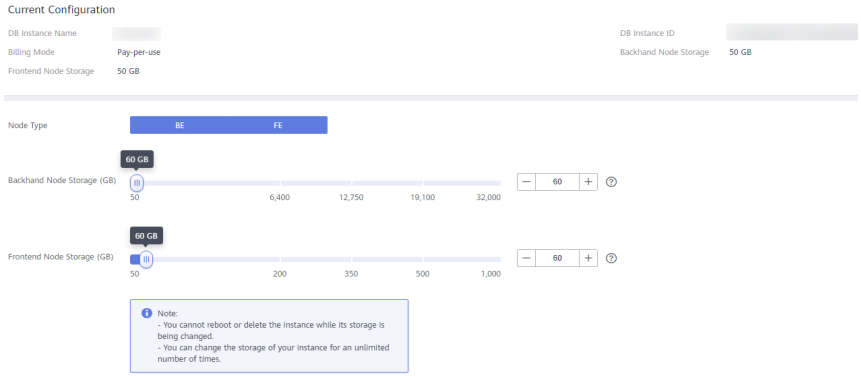


Table 20-15 Parameter description

Parameter	Description
Node Type	You can select either BE nodes or FE nodes, or select both BE nodes and FE nodes.
Backend Node Storage (GB)	You need to set this parameter if **BE** is selected for **Node Type**. You can change storage space to up to 32,000 GB only by a multiple of 10 GB.
Frontend Node Storage (GB)	You need to set this parameter if **FE** is selected for **Node Type**. You can change storage space to up to 1,000 GB only by a multiple of 10 GB.

Step 8 Click **Next**.

Step 9 Confirm the information and click **Submit**.

Step 10 After the storage space is changed, view and manage it on the **Basic Information** page of the HTAP instance.

----End

20.5.3 Adding Read Replicas to a Standard HTAP Instance

Scenarios

In read-intensive scenarios, the primary instance may be unable to handle the read pressure and services 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.

After creating a standard HTAP instance, you can add read replicas to it as required.

Deployment Relationships

New read replicas and existing nodes are deployed in the same AZ.

Constraints

- A single cluster instance supports a maximum of 10 FE nodes and 10 BE nodes.
- You cannot add read replicas to a single-node instance.
- Deleted read replicas cannot be recovered. Exercise caution when performing this operation.
- If another operation is being performed on a DB instance, the read replicas of the instance cannot be manually deleted.
- In a single cluster instance, you can only delete a fe-follower node when there is one fe-leader node and two or more available fe-follower nodes.

  For details about the fe-leader and fe-follower nodes, see **What Is HTAP of Standard Edition?**
- Read replicas of a single-node instance cannot be deleted.
- If a standard HTAP instance has been associated with a proxy instance, you need to manually configure read weights for new read replicas.

Billing

Table 20-16 Billing for new read replicas

Billing Mode of New Read Replicas	Impact on Price
Yearly/ Monthly	You will be billed for the new read replicas based on the time remaining in the requested period of your instance. You need to pay the price difference. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month cluster standard HTAP instance with three BE and three FE nodes (specifications: 4 vCPUs | 16 GB; storage: extreme SSD, 50 GB) in CN-Hong Kong on April 1, 2025. The instance price was $1874.40 USD per month. On April 15, 2025, you added one BE node and one FE node. The instance price was $2499.20 USD per month. **Price difference = Price for the new instance configuration x Remaining period – Price for the original instance configuration x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in April)/30 (Maximum number of days in April) = 0.5. Price difference = $2499.20 USD x 0.5 – $1874.40 USD x 0.5 = $312.4 USD
Pay-per-use	New read replicas are billed by hour. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and choose **More** > **Create Read Replica** in the **Operation** column.

Step 7 On the displayed page, set required parameters.

Figure 20-13 Creating read replicas

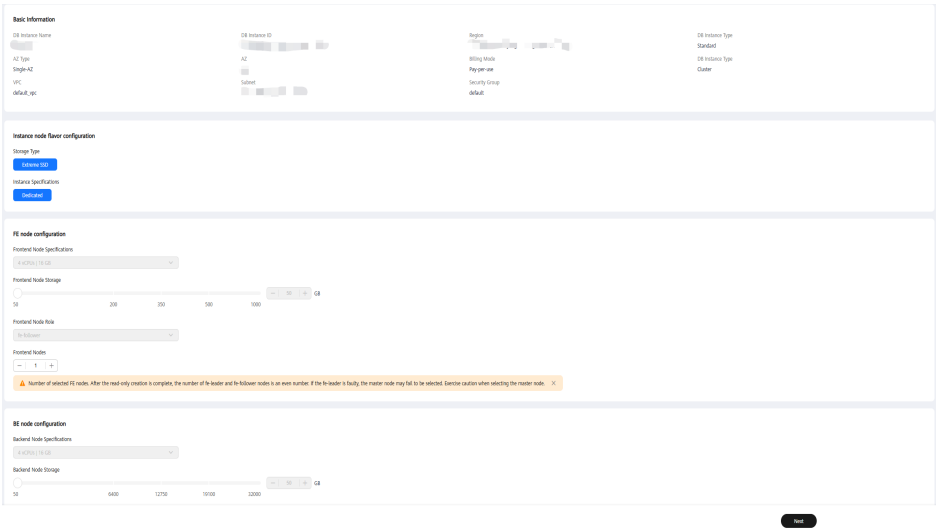


Table 20-17 Parameter description

Parameter	Description
Instance node flavor configuration	• **Storage Type**: By default, the value is the same as that of the current HTAP instance and cannot be changed. • **Instance Specifications**: By default, the value is the same as that of the current HTAP instance and cannot be changed.
FE node configuration	• **Frontend Node Specifications**: By default, the value is the same as that of the current HTAP instance and cannot be changed. • **Frontend Node Storage**: By default, the value is the same as that of the current HTAP instance and cannot be changed. • **Frontend Node Role**: By default, the value is the same as that of the current HTAP instance and cannot be changed. • **Frontend Nodes**: The value ranges from 0 to 7. You can create up to 7 nodes at a time.
BE node configuration	• **Backend Node Specifications**: By default, the value is the same as that of the current HTAP instance and cannot be changed. • **Backend Node Storage**: By default, the value is the same as that of the current HTAP instance and cannot be changed. • **Backend Nodes**: The value ranges from 0 to 7. You can create up to 7 nodes at a time.

Step 8 Click **Next**.

Step 9 Confirm the information and click **Submit**.

Step 10 After the read replicas are created, view and manage them on the **Basic Information** page of the HTAP instance.

- Pay-per-use instance: To delete a read replica, locate the read replica in the node list and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.
- Yearly/Monthly instance: To delete a read replica, locate the read replica in the node list and click **Unsubscribe** in the **Operation** column. On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**. In the displayed dialog box, click **Yes**. After the order is successfully unsubscribed, the read replica will be deleted.

----End

20.5.4 Setting a Repair Mode for Abnormal Tables

Scenarios

An exception may occur when an HTAP synchronization task processes incremental DDL statements. As a result, AP data may be inconsistent with TP data. In this case, an abnormal table is displayed. TaurusDB provides automatic repair for abnormal tables. It periodically checks abnormal data tables and creates temporary tasks to repair them based on the repair mode.

Constraints

- A synchronization task can have only one repair task. A maximum of five repair tasks can run on an instance at the same time. If there are too many repair tasks, they will be executed in subsequent periods.
- If a repair task fails multiple times, the repair task automatically stops.
- Abnormal tables caused by RENAME DDL statements on a database need to be manually repaired.

Setting a Repair Mode for Abnormal Tables

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, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

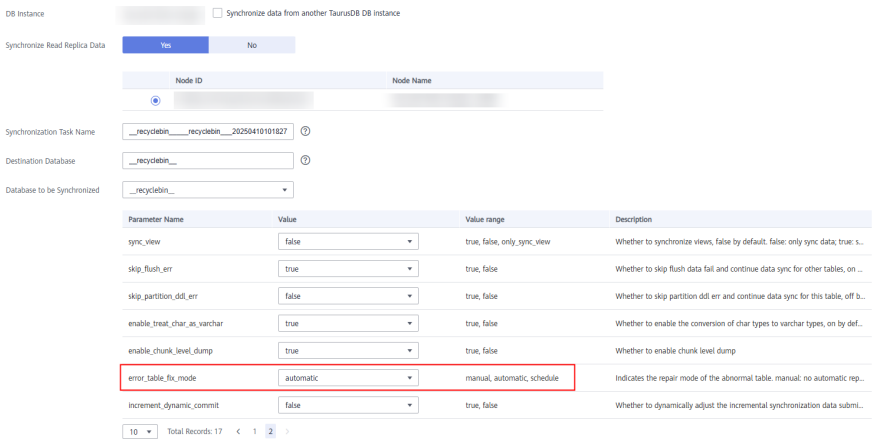
Step 6 Locate an HTAP instance and click its name to access the **Basic Information** page.

Step 7 In the navigation pane, choose **Data Synchronization**. Click **Create Synchronization Task**. On the displayed page, set **error_table_fix_mode**.

The following three repair modes are supported. The default value is **automatic**.

- **manual**: Abnormal tables will be manually repaired.
- **automatic**: Abnormal tables will be automatically repaired.
- **schedule**: Abnormal tables will be repaired in the maintenance time window (02:00–06:00).

Figure 20-14 Setting a repair mode



----End

20.5.5 Adjusting Blacklisted or Whitelisted Tables of a Standard HTAP Instance and Repairing Tables

Scenarios

After creating a synchronization task for a TaurusDB database, you may need to perform the following operations on the task:

- **Remove** tables that are no longer required for analysis from the synchronization task.
- **Add** unsynchronized tables to the synchronization task.
- **Repair tables** that have encountered synchronization exceptions due to certain reasons (for example, DDL statements that are not supported by the HTAP database are executed).

Constraints

- If tables fail to be renamed, you cannot synchronize them by repairing the tables. Instead, you need to add the tables.
- You are advised to modify the blacklist or whitelist during off-peak hours.
- A maximum of 50 blacklisted or whitelisted tables can be adjusted at a time.
- You cannot switch between the blacklist and whitelist.
- You can only add tables to the blacklist and remove tables from the whitelist.
- A main task can have only one temporary task at a time. You can add or repair other tables only after the temporary task is merged or deleted.

- DDL operations are not supported during the table addition or repair process. If a DDL operation is performed, the temporary task will fail to merge. In this case, you need to delete the temporary task and then add or repair the tables again.
- You can create multiple synchronization tasks for a given database. When adding tables, ensure that they are not already in other synchronization tasks. Otherwise, the tables will fail to be added.

Deleting Tables

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**.

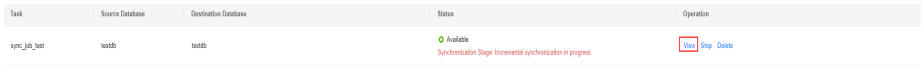
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Click the name of an HTAP instance to access the **Basic Information** page.

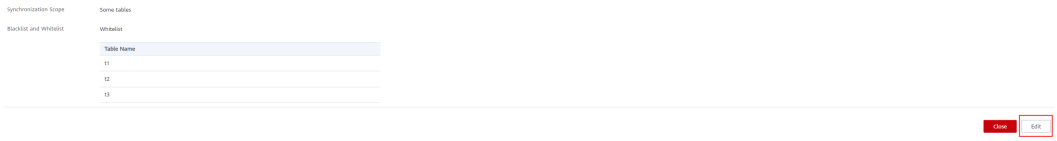
Step 7 In the navigation pane, choose **Data Synchronization**. On the displayed page, locate a synchronization task and click **View** in the **Operation** column.

Figure 20-15 Entry for viewing a synchronization task



Step 8 On the displayed page, click **Edit** in the lower right corner.

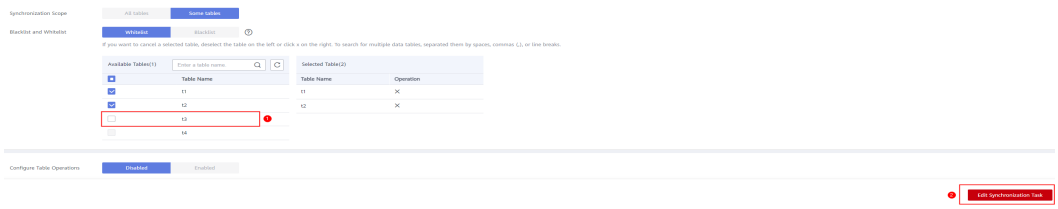
Figure 20-16 Entry for editing a synchronization task



Step 9 Deselect tables to be deleted in the whitelist or select tables to be deleted in the blacklist, click **Edit Synchronization Task** in the lower right corner, and return to the synchronization list.

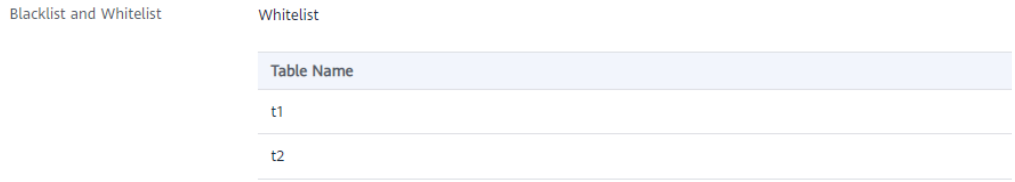
You can quickly locate desired tables by entering their names at once in the search box. Separate the names by commas (,), spaces, or line breaks. The matched tables will be displayed in the search results.

Figure 20-17 Deleting a table from a synchronization task



Step 10 Click **View** again and check that the deleted table is not in the whitelist.

Figure 20-18 Checking that the deleted table is not in the whitelist



----End

Add Tables

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, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Click the name of an HTAP instance to access the **Basic Information** page.

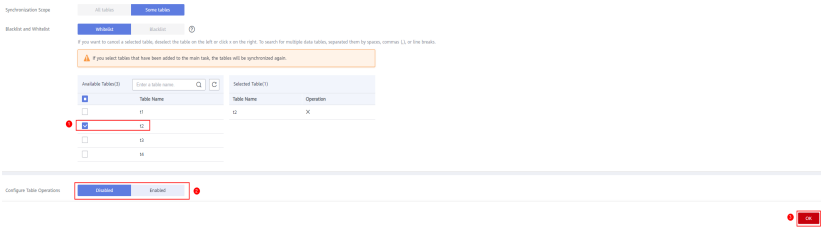
Step 7 In the navigation pane, choose **Data Synchronization**. On the displayed page, locate a task and choose **More** > **Add Tables** in the **Operation** column.

Step 8 Select a table to be added, set **Configure Table Operations** as required, and click **OK**.

If you select a table that has been synchronized, the table will be synchronized again based on the latest **Configure Table Operations** setting.

Tables that are being synchronized in other tasks cannot be added.

Figure 20-19 Selecting a table to be added and setting **Configure Table Operations**

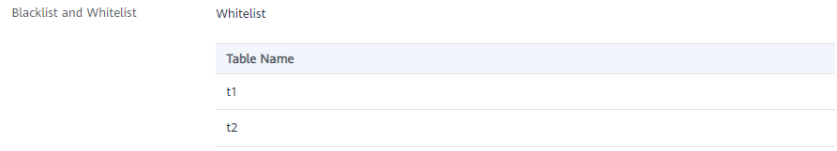


Step 9 Return to the **Data Synchronization** page and click **Synchronize** in the **Operation** column of the new temporary task.

Figure 20-20 Synchronizing a temporary task

Step 10 Once the temporary task disappears from the list, the table has been synchronized and added to the main task. Click **View** in the **Operation** column of the main task to check whether the table is successfully added.

Figure 20-21 Table added to the main task



----End

Repairing Tables

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**.

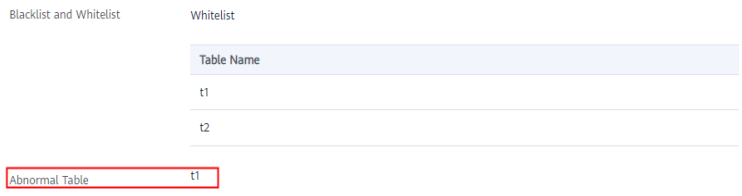
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Click the name of an HTAP instance to access the **Basic Information** page.

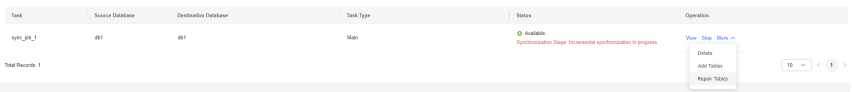
Step 7 In the navigation pane, choose **Data Synchronization**. On the displayed page, view a task. If a table in the task encounters synchronization exceptions, repair it.

Figure 20-22 Abnormal table t1 found



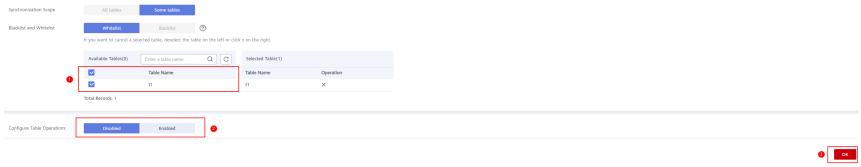
Step 8 Return to the **Data Synchronization** page and choose **More** > **Repair Tables** in the **Operation** column of the task.

Figure 20-23 Repairing tables



Step 9 On the displayed page, select the table to be repaired, set **Configure Table Operations** as required, and click **OK**.

Figure 20-24 Selecting the table to be repaired



Step 10 Return to the **Data Synchronization** page and click **Synchronize** in the **Operation** column of the new temporary task. After the abnormal table is repaired, the temporary task is automatically merged into the main task.

Figure 20-25 Synchronizing the new temporary task

Step 11 Once the temporary task disappears, the abnormal table has been repaired. View the main task information and check that there is no abnormal table.

Figure 20-26 Abnormal table repaired

----End

20.5.6 Upgrading the Minor Version of a Standard HTAP Instance (OBT)

Scenarios

You can upgrade the minor version of your standard HTAP instance to improve performance, optimize functions, and fix bugs.

Upgrade Methods

A minor version can be upgraded in either of the following ways:

- Upon submission: The system upgrades the minor version upon your manual submission of the upgrade request.
- In maintenance window: The system upgrades the minor version during the maintenance window you have specified.

Constraints

- To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- When any new minor version is released, upgrade the minor version of your standard HTAP instance immediately or during the maintenance window.
- The upgrade will cause the standard HTAP instance to reboot and will interrupt services intermittently. To minimize the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.
- A minor version upgrade cannot be rolled back after the upgrade is complete.
- Before upgrading a minor version, check whether there are temporary data synchronization tasks. If there are, wait until the tasks are complete or manually delete the temporary tasks.

Upgrading the Minor Version of a Single Standard HTAP Instance

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

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

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

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

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Click the name of an HTAP instance to access the **Basic Information** page.

Step 7 In the **DB Instance Information** area, click **Upgrade** next to **Kernel Version**.

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

- **Upon submission**: The system upgrades the minor version immediately after your submission of the upgrade request.
- **In maintenance window**: The system upgrades the minor version during the maintenance window you have specified. After the operation is complete, on the **Task Center** page, click **Scheduled Tasks** and view the information about the upgrade task.

----End

20.6 Data Synchronization Using Standard HTAP Instances

20.6.1 Replicating and Rebuilding a Synchronization Task (OBT)

Scenarios

You can rebuild a synchronization task for a standard HTAP instance. You can also replicate an existing synchronization task from another standard HTAP instance.

Constraints

- To use these functions, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- Rebuilding a task will delete it. Exercise caution when performing this operation.
- If a task to be rebuilt requires table synchronization configuration verification, the task will be deleted before the verification is complete.
- Synchronization tasks can only be replicated between instances in the same region.

Replicating a Synchronization 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 and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click its name to access the **Basic Information** page.

Step 7 In the navigation pane, choose **Data Synchronization**. On the displayed page, click **Replicate Synchronization Task**.

Step 8 Set **Source Instance** and **Synchronization Task**.

Step 9 Click **OK**. On the displayed **Replicate Synchronization Task** page, edit task details.

Step 10 Click **Create Synchronization Task**.

----End

Rebuilding a Synchronization 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 and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

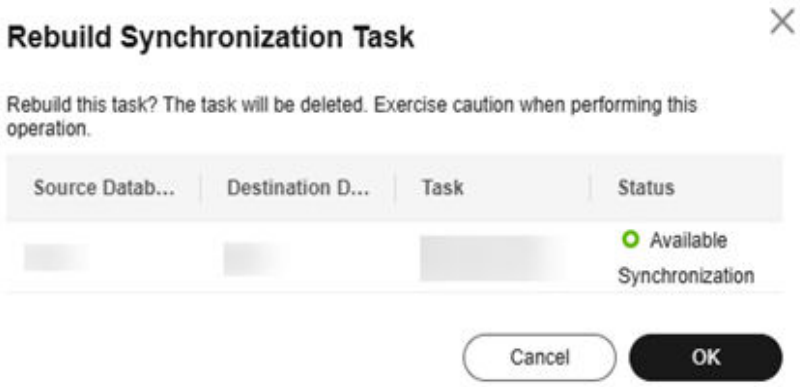
Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click its name to access the **Basic Information** page.

Step 7 In the navigation pane, choose **Data Synchronization**. On the displayed page, locate the task to be rebuilt and choose **More** > **Rebuild** in the **Operation** column.

Step 8 In the displayed dialog box, confirm the information and click **OK**. On the displayed **Rebuild Synchronization Task** page, edit task details.

Figure 20-27 Rebuilding a synchronization task



Step 9 Click **Rebuild Now**.

----End

20.7 Monitoring Metrics and Event Alarms

20.7.1 Viewing Metrics of a Standard HTAP Instance or Nodes

Scenarios

Cloud Eye monitors operating statuses of standard HTAP instances. You can view the metrics of standard HTAP instances on the management console.

Prerequisites

- Standard HTAP instances are running properly.

  Cloud Eye does not display the metrics of faulty or deleted HTAP instances. When the status of an HTAP instance becomes **Available**, you can view its metrics.

  NOTE

  If an HTAP instance has been faulty for 24 hours, Cloud Eye assumes that the instance no longer exists and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the instance.

- Standard HTAP instances have kept running properly for about 10 minutes.

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

Viewing Metrics of a Standard HTAP Instance

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

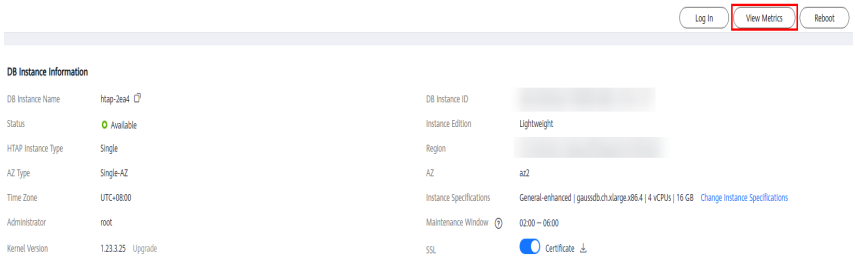
Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click **View Metrics** in the **Operation** column.

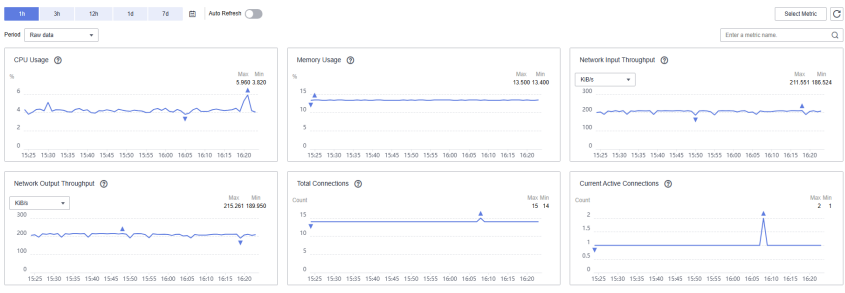
Alternatively, click the HTAP instance name. On the displayed **Basic Information** page, click **View Metrics** in the upper right corner.

Figure 20-28 Entry for viewing metrics



Step 7 On the displayed **Cloud Eye** page, view metrics of the HTAP instance.

Figure 20-29 Viewing metrics of an HTAP instance



----End

Viewing Metrics of a FE or BE Node

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

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

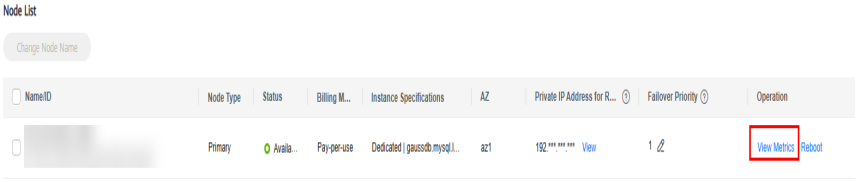
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Locate an HTAP instance and click its name to access the **Basic Information** page.

Step 7 In the **Node List** area, locate a node and click **View Metrics** in the **Operation** column.



Step 8 View metrics of a FE or BE node.

----End

20.7.2 Event Monitoring for Standard HTAP Instances

Scenarios

Event monitoring provides reporting, query, and alarm functions for event data. You can create alarm rules for both system and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on TaurusDB HTAP instances that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by services to Cloud Eye.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events.

Events for Standard HTAP Instances

HTAP (Standard Edition) is a sub-feature of TaurusDB instances. The alarm notifications of event monitoring for standard HTAP instances are the same as those for TaurusDB instances.

Table 20-18 HTAP (Standard Edition)

Event Source	Event Name	Event ID	Event Severity	Description	Handling Suggestion	Impact
TaurusDB	Faulty DB instance	TaurusInstanceRunningStatusAbnormal	Major	The instance process may be faulty or there may be abnormal tables during instance data synchronization.	Check whether there are abnormal tables. If there are, **repair the abnormal tables**. If the fault persists, **submit a service ticket**.	Workloads may be affected, or instance data cannot be synchronized.
	DB instance recovered	TaurusInstanceRunningStatusRecovered	Major	The instance is recovered.	Observe the service running status.	None
	Faulty node	TaurusNodeRunningStatusAbnormal	Major	The node process may be faulty.	Observe the instance and service running statuses.	Workloads may be affected.

Event Source	Event Name	Event ID	Event Severity	Description	Handling Suggestion	Impact
	Node recovered	TaurusNodeRunningStatusRecovered	Major	The node is recovered.	Observe the service running status.	None

Viewing Event Monitoring Data

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**. Locate an HTAP instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the HTAP instance name. On the displayed **Basic Information** page, click **View Metrics** in the upper right corner.

Step 6 Click to return to the main page of Cloud Eye.

Step 7 In the navigation pane, choose **Event Monitoring**.

Step 8 On the displayed **Event Monitoring** page, all system events of the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view events generated in different periods.

Step 9 Locate an event and click **View Event** in the **Operation** column to view details about a specific event.

----End

Creating Alarm Rules for Event Monitoring

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

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

Step 3 In the navigation pane, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the displayed page, configure parameters as needed.

Table 20-19 Parameter description

Parameter	Description
Name	Specifies the name of the alarm rule. The system generates a random name, and you can change it if needed.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create an enterprise project.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Trace Type	Specifies the event type of the metric corresponding to the alarm rule.
Event Source	Specifies the service the event is generated for. Example value: **TaurusDB**
Monitoring Scope	Specifies the monitoring scope for event monitoring.
Method	Specifies the event creation method.
Alarm Policy	**Event Name** indicates the instantaneous operations users performed on system resources, such as login and logout. For details about events supported by Event Monitoring, see **Table 20-18**. You can select a trigger mode and alarm severity as needed.

Click  to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 20-20 Alarm notification parameters

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text message, or through HTTP/HTTPS request to servers.

Parameter	Description
Notification Object	Specifies the object an alarm notification is to be sent to. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Validity Period	Cloud Eye sends notifications only within the validity period specified in the alarm rule. If you set **Validity Period** to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification.

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

----End

20.8 Standard HTAP Account Management

Standard HTAP instances use the following methods to manage accounts:

- After TaurusDB data is synchronized to a standard HTAP instance, accounts cannot be synchronized. You need to manually create database accounts on the HTAP instance.
- You can create databases, tables, and accounts for your HTAP instances as needed.

This section describes how to create an account, reset the password, modify account permissions, and delete an account on the TaurusDB console.

System Accounts

To provide O&M services, the system automatically creates system accounts when you create HTAP instances, but these system accounts are not available to you.

- **rdsAdmin**: a management account with superuser permissions, which is used to query and modify instance information, rectify faults, migrate data, and restore data.
- **rdsMetric**: an account used for metric monitoring. This account is used by watchdog to collect database status data.

> **WARNING**
>
> Attempting to delete, rename, or change passwords or permissions for those accounts will result in an error. Exercise caution when performing these operations.

Creating a Database Account

Step 1 Log in to the management console.

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

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

Step 4 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 5 In the navigation pane, choose **HTAP Analysis**.

Step 6 Click the name of an HTAP instance to access the **Basic Information** page.

Step 7 In the navigation pane, choose **Accounts**. On the displayed page, click **Create Account**.

Step 8 In the displayed dialog box, set the required parameters.

Figure 20-30 Creating a database account

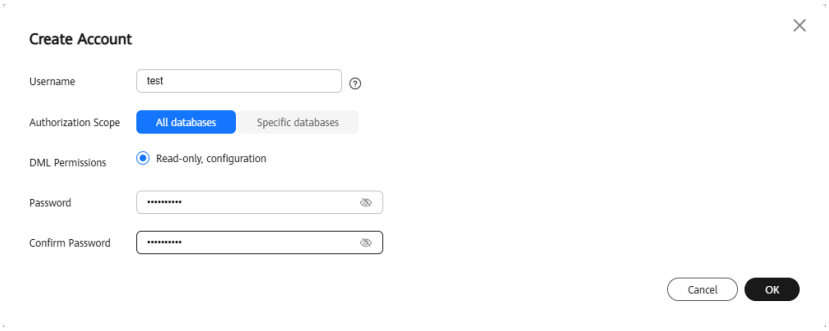


Table 20-21 Parameter description

Parameter	Description
Username	Contains 2 to 32 characters. It must start with a lowercase letter and end with a lowercase letter or digit. Only lowercase letters, digits, and underscores (_) are allowed.

Parameter	Description
Authorization Scope	• **All databases** • **Specific databases** **Database Not Authorized**: When creating an account, do not select any database in this area. The created account cannot perform operations on any database. To learn how to grant the required permissions for a particular database, see **Modifying Account Permissions**. **Database Authorized**: The databases selected in the **Database Not Authorized** area are displayed.
DML Permissions	The permissions include read-only, read/write, read and configuration, and read/write and configuration. Currently, only **Read-only, configuration** is available on the console.
Password	• Contains 8 to 32 characters. • Contains at least three of the following types of characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*-_=+?,). • Cannot be the username or the username backwards.
Confirm Password	Must be the same as the new password.

Step 9 Click **OK**.

Step 10 In the account list, view the account information, including the username, authorized databases, and DML permissions.

You can **reset account passwords**, **change account permissions**, or **delete accounts**.

----End

Resetting a Password

Step 1 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 2 In the navigation pane, choose **HTAP Analysis**.

Step 3 Click the name of an HTAP instance to access the **Basic Information** page.

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

Step 5 In the displayed dialog box, enter a new password, confirm the password, and click **OK**.

----End

Modifying Account Permissions

> **CAUTION**
>
> If you delete a database somewhere other than on the HTAP console, permissions granted specifically for the database are not automatically deleted. They must be deleted manually.

Step 1 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

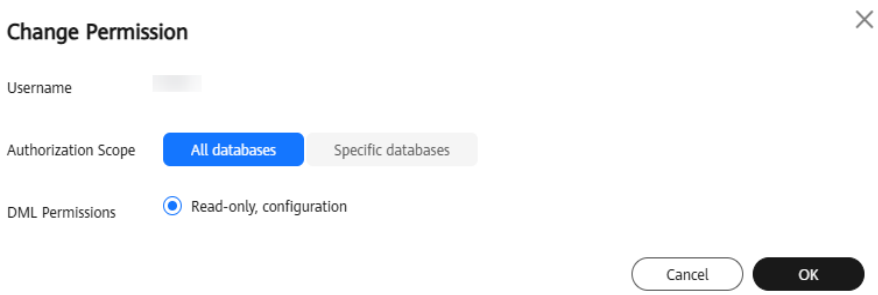
Step 2 In the navigation pane, choose **HTAP Analysis**.

Step 3 Click the name of an HTAP instance to access the **Basic Information** page.

Step 4 In the navigation pane, choose **Accounts**. On the displayed page, locate an account and click **Change Permission** in the **Operation** column.

Step 5 In the displayed dialog box, modify permissions as required and click **OK**.

Figure 20-31 Changing permissions



----End

Deleting an Account

Step 1 On the **Instances** page, locate a TaurusDB instance and click its name to access the **Basic Information** page.

Step 2 In the navigation pane, choose **HTAP Analysis**.

Step 3 Click the name of an HTAP instance to access the **Basic Information** page.

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

Step 5 In the displayed dialog box, confirm the information and click **OK**.

----End

20.9 Syntax and Data Type Mappings Between HTAP and TaurusDB Instances

Standard HTAP instances support the native syntax of StarRocks. Character string comparison and the names of databases, tables, views, users, and roles are case sensitive, but the names of columns and partitions are case insensitive. For details, see **StarRocks' official documentation**.

When data of TaurusDB instances is synchronized to HTAP instances, the data types will be converted. For details, see **Table 20-22**.

Table 20-22 Data type conversion

Data Type	TaurusDB Instance Data Type	HTAP Instance Data Type
NUMERTIC	TINYINT	TINYINT
	TINYINT UNSIGNED	SMALLINT **CAUTION** Unsigned integers in TaurusDB are converted to a larger integer type for storage. When an integer is converted to an unsigned integer, ensure that no negative number exists in historical data.
	SMALLINT	SMALLINT
	SMALLINT UNSIGNED	INT
	MEDIUMINT	INT
	INTEGER/INT	INT
	INTEGER/INT UNSIGNED	BIGINT
	BIGINT	BIGINT
	BIGINT UNSIGNED	LARGEINT
	DECIMAL/NEMERIC	DECIMAL **NOTE** If the precision of DECIMAL is greater than 64, the DECIMAL data type of TaurusDB instances will be converted to VARCHAR when being synchronized to HTAP instances.
	FLOAT	FLOAT
	DOUBLE/REAL	DOUBLE
	BIT	BIT

Data Type	TaurusDB Instance Data Type	HTAP Instance Data Type
DATE TIME	DATE	DATE
	DATETIME	DATETIME
	TIMESTAMP	DATETIME
	TIME	VARCHAR
	YEAR	INT
STRING	CHAR	CHAR/VARCHAR
	VARCHAR	VARCHAR
	BINARY	VARBINARY
	VARBINARY	VARBINARY
	BLOB	VARBINARY
	TEXT	TEXT
	ENUM	VARCHAR **CAUTION** 1. When the ENUM and SET types are used, set the **binlog_row_metadata** parameter of TaurusDB instances to **FULL**. 2. Use the case-sensitive character string format during query.
	SET	VARCHAR
SPATIAL	GEOMETRY	Not supported
	POINT	Not supported
	LINESTRING	Not supported
	POLYGON	Not supported
	MULTIPOINT	Not supported
	MULTILINESTRING	Not supported
	MULTIPOLYGON	Not supported
	GEOMETRYCOLLECTION	Not supported
JSON	JSON	JSON

20.10 Performance Tuning

Setting Synchronization Task Parameters

- **max_full_sync_task_threads_num**: number of full synchronization threads. By default, it is set to half of the vCPUs on FE nodes. Increasing this value will make full synchronization faster, but more vCPUs and memory of OLTP and OLAP will be consumed. Set an appropriate value for this parameter based on the system load when executing a full synchronization task. If multiple full synchronization tasks are executed at the same time, decrease this parameter value.
- **max_incremental_sync_task_threads_num**: number of incremental synchronization threads. By default, it is set to half of the vCPUs on FE nodes. A larger value indicates that more threads are used for incremental synchronization, more resources are consumed, and the synchronization latency is shorter. If there are more than five synchronization tasks on an instance, reduce the number of synchronization threads for each task.
- **expect_tablet_size**: expected size of source data stored in each bucket, in GB. The default value is **3**. If most tables in a database have less than 3 GB of data and there are only a few tables with more data, decrease this value.
- **expect_tablet_num_for_one_partition**: expected default number of buckets in each partition. If this parameter is set to **0**, the number of buckets is calculated based on the data size. The default value is **2**. If there is no data in a table, this default value will be used. If there is data in a table, the number of buckets is calculated as follows: Data size/Value of **expect_tablet_size**. If a partition key is specified for table synchronization, you need to evaluate the number of buckets required for data in each partition. The number of buckets for a table is calculated as follows: Number of partitions x Number of buckets in each partition.

Improving Query Performance

- SQL tuning

  Do not use SELECT *. Remove redundant columns and functions.

- Query cache

  The query cache is suitable for scenarios where data can be categorized as hot data and cold data and aggregate queries are frequently executed. To enable the query cache, use DAS to connect to the OLAP database and execute **SET GLOBAL enable_query_cache=true;**

- Sorting keys

  Add commonly used filter criteria to sorting keys. The order of the sorting keys should be determined based on their usage frequency and data cardinality, with priority given to high cardinality. You are advised to set no more than five sorting keys. Sorting keys are widely used for tuning OLAP performance. When creating a synchronization task, you can use table synchronization to set sorting keys.

- Partitions

A time column whose value does not change is often used for the WHERE filtering. Use the column to create partitions. When creating a synchronization task, you can use table synchronization to set partitions.

- Indexes

  Add indexes to the columns used for filtering. Bitmap indexes are suitable for columns with a cardinality of around 10,000 to 100,000. Bloom filter indexes are suitable for columns with a cardinality of more than 100,000. After data is synchronized, you can connect to the OLAP database through DAS and run SQL commands to create indexes.

- Materialized views

  If there are multiple frequently used queries with different filter criteria and sorting keys can only adapt to one query, create a materialized view to adapt to other queries. After data is synchronized, you can connect to the OLAP database through DAS and run SQL commands to create materialized views.

21 Application Lossless and Transparent (ALT)

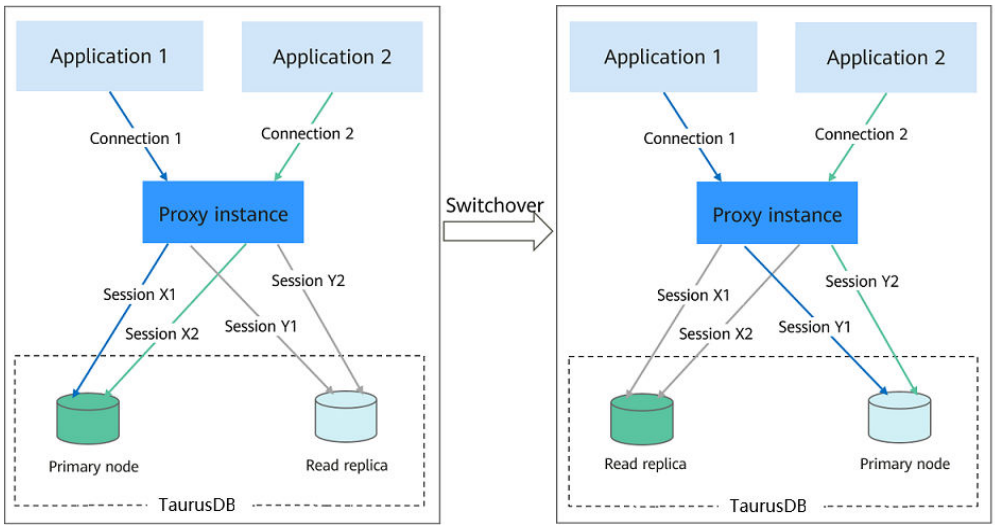
21.1 What Is ALT?

Database sessions may be interrupted when a read replica is promoted to primary, a minor version is upgraded, or specifications are changed. Applications need to check session statuses and react to changes by determining: whether a database connection or transaction has been interrupted, how to compensate for transactions, and how to rebuild session contexts.

To address these issues, TaurusDB provides ALT, which prevents database connection and transaction interruptions during database system switchover. There is no need to compensate for transactions or rebuild session contexts, ensuring application continuity.

Architecture

Figure 21-1 Architecture



ALT can be enabled for your application connections. When you connect to a proxy instance and then promote a read replica to primary, change specifications, or upgrade the minor version, the system can replicate your backend sessions. Once a secure transaction boundary is reached, backend sessions will be fully cloned to the destination node, and workloads do not even notice.

NOTE

A secure transaction boundary refers to the status that a transaction in the current session has been committed but the next transaction is not started. A secure transaction boundary can be reached in any of the following situations:

- Each statement in a transaction block with autocommit enabled is executed.
  start transaction;
  DML;
  commit;
- The commit operation is complete with autocommit disabled.
- A single DML or DDL statement is executed.
- The lock is released when a table lock, backup lock, or user-defined lock is used.

Precautions

Table 21-1 Precautions

Category	Precaution
Version constraints	• The kernel version of the TaurusDB instance must be 2.0.54.240600 or later. • The kernel version of the proxy instance must be 2.24.06.000 or later. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**

Category	Precaution
Usage constraints	• To use ALT, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. • The TaurusDB instance has at least one read replica. A proxy instance has been created and the TaurusDB instance must be connected through the proxy address. • Proxy instances in read-only mode do not support ALT. • Proxy instances in primary/standby mode do not support ALT. • Single-node or multi-primary TaurusDB instances do not support ALT. • When you enable ALT for the first time, the TaurusDB instance will reboot. Enabling or disabling ALT will cause a proxy instance to reboot. Once ALT is disabled for all proxy instances, the TaurusDB instance will also reboot. • ALT requires no active transactions on each connection. When promoting a read replica to primary, you need to wait for ongoing transactions to end. The interval for waiting for the transactions to end is called the transaction draining timeout interval, which is controlled by **rds_tac_drain_timeout**. This parameter defaults to 5s and ranges from 1s to 60s. – Increase this interval for heavy workloads, numerous prepared statements, or time-consuming transactions. – Decreasing this interval is not recommended. If there are connections that do not drain transactions within the configured transaction draining timeout interval, ALT does not take effect for these connections. • During an ALT switchover, standby connections will be established on the new host for a brief period, equal in number to those on the original primary node. Ensure that the maximum number of connections of the TaurusDB instance is at least twice the current number of connections. To change the maximum number of connections, you need to evaluate the instance specifications and memory usage. For details, see **What Is the Maximum Number of Connections to a TaurusDB Instance?** • ALT supports prepared statements. During a switchover, the contexts of prepared statements are rebuilt. If there are a large number of prepared statements, the switchover success rate may be affected. • You are advised to perform an ALT switchover during off-peak hours. If the primary node and read replicas are overloaded, the switchover success rate may be affected. • For details about syntax and function constraints of proxy instances, see **Precautions for Proxy Instances**.

Category	Precaution
Unsupported functions	• Enabling ALT makes your instance lose support for some system variable values. – *innodb_ft_user_stopword_table*: It can only be set to **NULL**. – *transaction_write_set_extraction*: It can only be set to **OFF**. – *profiling*: It cannot be set to **1** or **ON**. • ALT does not support Transparent Data Encryption (TDE). • ALT is unavailable when any of the following proxy capabilities is enabled: – Session-level connection pool – Any column containing more than 16 MB of data – Any query result set containing more than 16 MB of data – Prepared statement cache • ALT does not support temporary tables created by users. • ALT is not supported in the following scenarios where a secure transaction boundary cannot be reached: – InnoDB transaction blocks are not committed in a timely manner. – There are unreleased table locks, user locks, backup locks, and binlog locks. – XA transactions are not committed or rolled back. • ALT will be likely to fail if a switchover, minor version upgrade, or specification change occurs frequently within a short period of time. • If ALT is enabled, prepared statements cannot be transferred in the following scenarios: – The cursor is opened and not closed in a prepared statement. – The variable of a prepared statement has saved the LONG_DATA type.

21.2 Enabling ALT

This section describes how to enable ALT.

Constraints

For details, see **Precautions**.

Procedure

Step 1 Log in to the management console.

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

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

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

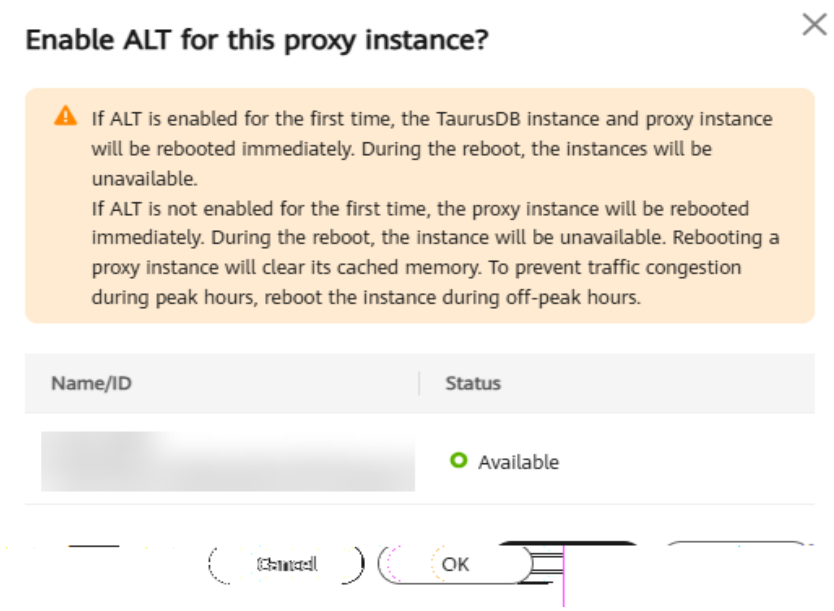
Step 5 In the navigation pane, choose **Database Proxy**.

Step 6 Click a proxy instance to go to the **Basic Information** page.

Step 7 In the **Proxy Instance Information** area, click under **ALT**.

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

Figure 21-2 Enabling ALT



If ALT is enabled for the first time, the TaurusDB instance and proxy instance will be rebooted immediately. During the reboot, the instances will be unavailable.

Rebooting an instance will clear the cached memory in it. You are advised to reboot it during off-peak hours.

To disable ALT, click . Disabling ALT will cause the proxy instance to reboot. Once ALT is disabled for all proxy instances, the TaurusDB instance will also reboot.

Step 9 On the **Basic Information** page of the proxy instance, check that the proxy instance status changes from **ALT is being configured** to **Available**.

----End

21.3 Example: Using ALT to Promote a Read Replica to Primary

This section describes how to use ALT to promote a read replica to primary. The process for minor version upgrades and specification changes is similar.

The process for using ALT to promote a read replica to primary is as follows:

Step 1: Buy a DB Instance

Step 2: Create a Proxy Instance

Step 3: Enable ALT

Step 4: Connect Your Application to the Proxy Instance

Step 5: Promote a Read Replica to Primary

Step 6: Test the ALT Effect

Constraints

If workloads are interrupted, see **Precautions**.

Step 1: Buy a DB Instance

For details, see **Buying a DB Instance**.

Step 2: Create a Proxy Instance

For details, see **Step 1: Create a Proxy Instance**.

Step 3: Enable ALT

For details, see **Enabling ALT**.

Step 4: Connect Your Application to the Proxy Instance

For details, see **Step 4: Use the Proxy Address to Connect to Your TaurusDB Instance**.

Step 5: Promote a Read Replica to Primary

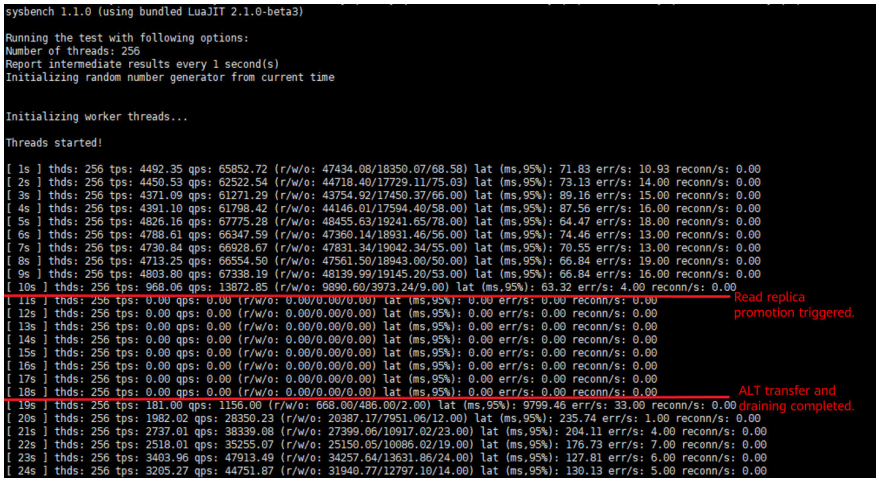
For details, see **Promoting a Read Replica to Primary**.

Step 6: Test the ALT Effect

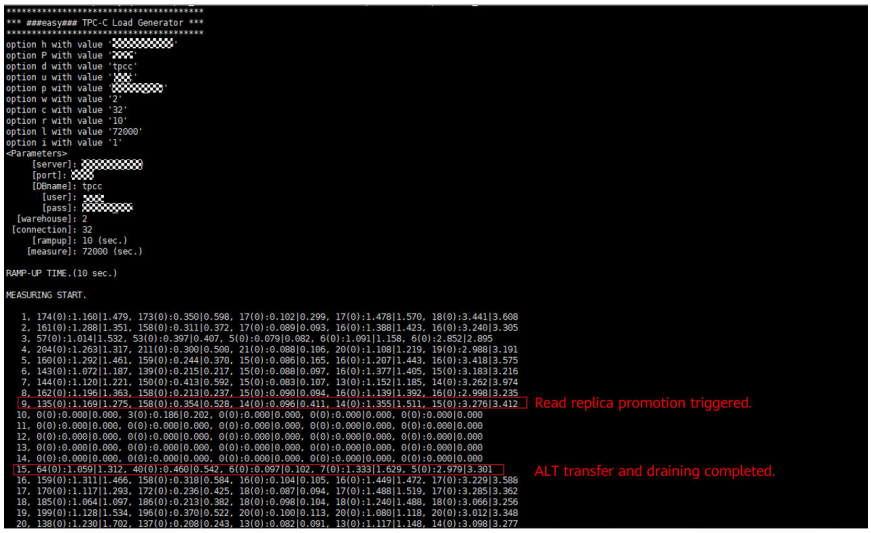
If ALT is enabled and you promote a read replica to primary using sysbench, tpcc-mysql, or the mysql client that is connected to the proxy address, your database only freezes briefly.

The following figures show you what effect ALT has when you promote a read replica to primary using sysbench, tpcc-mysql, and the mysql client.

- **Promoting a read replica to primary using sysbench**

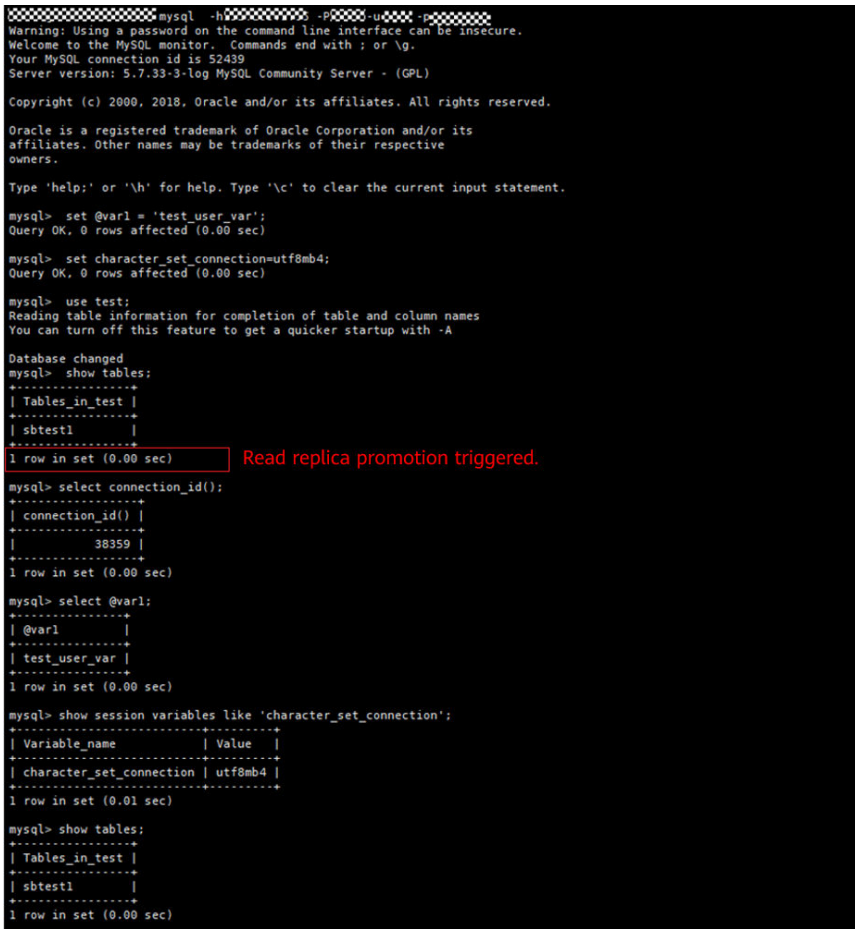


- **Promoting a read replica to primary using tpcc-mysql**

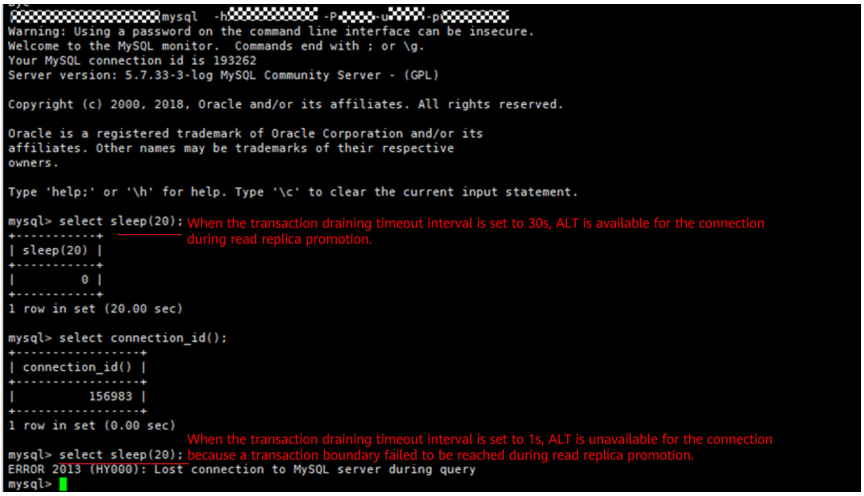


- **Promoting a read replica to primary using the MySQL CLI**

  As shown in the following figure, user-defined variables, session variables, and databases remain unchanged before and after you promote a read replica to primary.



The following figure shows how the transaction draining timeout interval determines whether ALT is available for the current session.

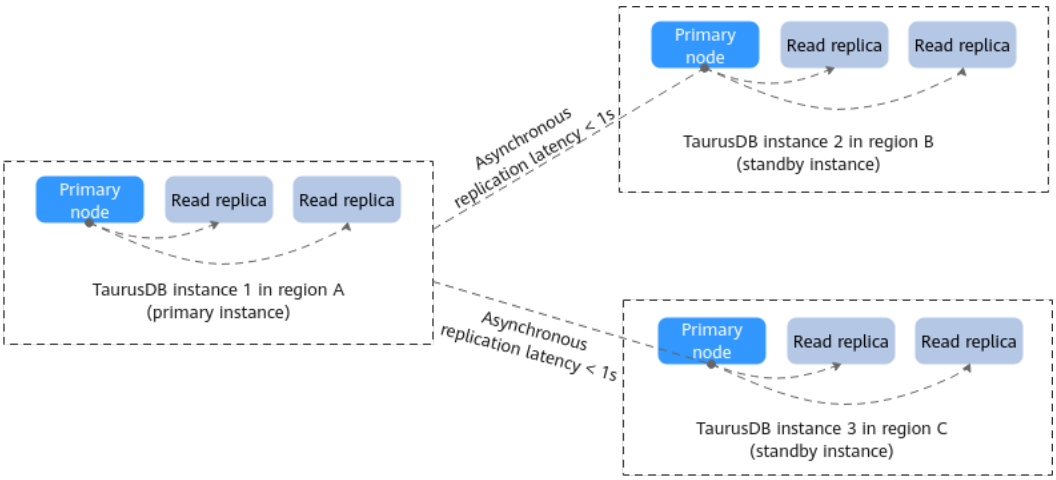


22 RegionlessDB Clusters (OBT)

22.1 What Is a RegionlessDB Cluster?

A RegionlessDB cluster consists of multiple TaurusDB instances in different regions around the world. Currently, a RegionlessDB cluster consists of one primary instance (in the primary region) and up to five standby instances (in standby regions). Data is synchronized between primary and standby instances, providing nearby access and regional DR capabilities.

Figure 22-1 RegionlessDB cluster principle



Basic Concepts

- Recovery Point Objective (RPO)

  The maximum data loss amount tolerated by the system
- Recovery Time Objective (RTO)

  The maximum tolerable service downtime, from the time when a disaster occurred to the time when services were recovered

Scenarios

- Remote multi-active deployment

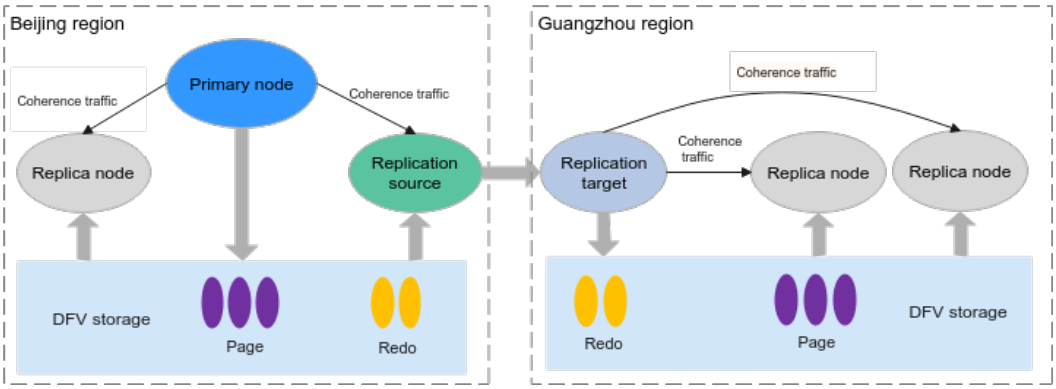
  Data is synchronized among instances in a RegionlessDB cluster. For lower network latency and quicker resource access, you can select the instance nearest to your workloads.

- Remote disaster recovery

  If there is a region-level fault on the primary instance, workloads can be switched to a standby instance for remote DR.

Architecture

Figure 22-2 Architecture



- Cross-region deployment is supported. Redo logs generated in the primary instance are synchronized to a standby instance and written to DFV storage. Pages required for database access are replayed. For details, see **Figure 22-2**. (Data is synchronized based on the replication node **Source** of the primary instance and the replication node **Target** of the standby instance.)
- In the primary instance, the read replica obtains required redo logs and pages from DFV storage through the primary node. In the standby instance, the read replica obtains required redo logs and pages from DFV storage through the replication node **Target**.

Advantages

- Global deployment and nearby data access

  Instances in a RegionlessDB cluster are from different regions around the world. Data generated by the primary instance can be directly read from the nearest standby instance.

- Low latency of cross-region replication

  Redo logs are directly and uninterruptedly read from the DFV storage for asynchronous replication. The replication latency is less than 1 second thanks to high-throughput parallel data synchronization.

- No downtime for the primary node during data synchronization

  The replication node of the primary instance reads data from different nodes in the DFV storage in parallel for synchronization. This means that the primary node does not need to directly synchronize data to the standby

instances. Instead, it only needs to update the location information of redo logs in the storage to the replication node of the primary instance. In this way, workloads on the primary node are not affected.

- Too many read replicas

  There are up to five standby instances in a cluster, and each standby instance supports up to 15 read replicas.

  When creating a DB instance, you can create up to 10 read replicas at a time.

- Region-level disaster recovery

  If there is a region-level fault on the primary instance, workloads can be quickly switched to a standby instance for remote DR, achieving an RPO in minutes and an RTO in seconds.

  NOTE

  To use region-level disaster recovery, **submit a service ticket**.

Constraints

- Before using this feature, you need to obtain the data security compliance requirements of the local region and evaluate the compliance with related laws and regulations.
- RegionlessDB is now in the OBT phase. To use this function, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- To enable communication between regions, you need to create a Virtual Private Network (VPN) in advance. For details about how to create a VPN, see **Configuring Enterprise Edition S2C VPN to Connect an On-premises Data Center to a VPC**.
- Only pay-per-use instances can be created.
- The kernel version must be 2.0.46.231000 or later, and the primary instance must be a new instance. For details about how to check the kernel version, see **How Can I Check the Version of a TaurusDB Instance?**
- The instances in a RegionlessDB cluster cannot use 192.168.0.0/16 as their subnet CIDR block.
- The subnet CIDR blocks of the primary and standby instances in different regions must be different.
- When a standby instance is created, data needs to be synchronized from the primary instance. The time required depends on how much data there is.
- The primary instance in a RegionlessDB cluster cannot be restored to the original instance, and other instances cannot be restored to any instance in a RegionlessDB cluster.
- If you have created proxy instances or HTAP instances for a TaurusDB instance, the TaurusDB instance cannot be used as an instance in a RegionlessDB cluster. To use it, delete the proxy instances or HTAP instances first.
- The primary instance does not support the following operations:
  - Changing a database port
  - Changing a private IP address

  - Creating an HTAP instance
  - Creating a proxy instance
- The standby instance does not support the following operations:
  - Resetting a password
  - Creating and restoring a backup
  - Creating an account
  - Authorizing an account
  - Creating a proxy instance
  - Creating an HTAP instance
  - Promoting a read replica to the primary node
  - Changing a database port
  - Changing a private IP address
  - Modifying auto scaling policies
- Data across regions is synchronized through a network. The VPN bandwidth must be greater than the write bandwidth of the primary instance in a RegionlessDB cluster.
- In large-scale DDL scenarios, the replication latency may fluctuate for more than 1 second.
- RegionlessDB clusters do not support OpenAPIs.
- A RegionlessDB cluster consists of one primary instance (in the primary region) and up to five standby instances (in standby regions). The primary instance processes both read and write requests, while the standby instances only process read requests. **Table 22-1** lists the maximum specifications supported by a RegionlessDB cluster.

Table 22-1 Specifications

Description	Primary Instance	Standby Instance
Max. Instances	1	5
Max. Read/Write Nodes per Instance	1	0
Max. Read-only Nodes per Instance	15	15

NOTE

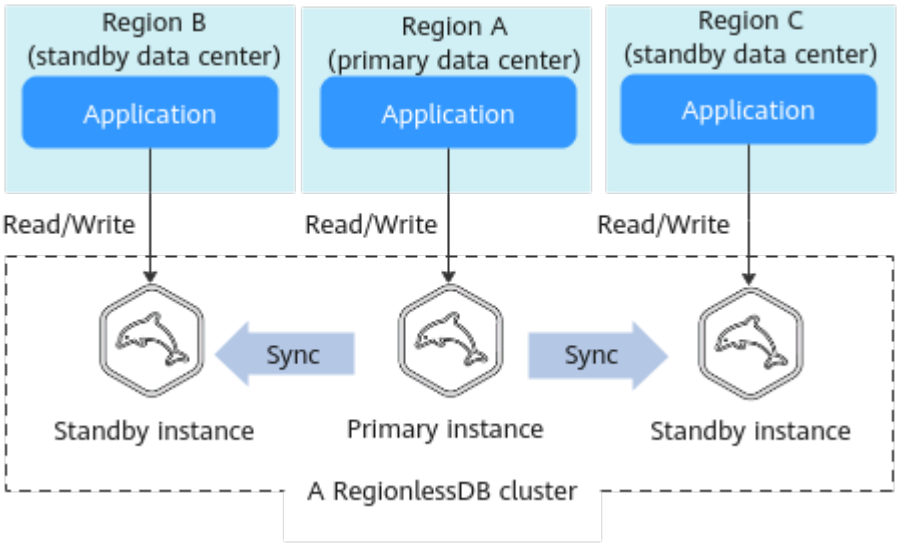
When you are creating a DB instance, a maximum of 10 read replicas can be created at a time.

22.2 Using a RegionlessDB Cluster for Remote Multi-Active DR

Scenarios

If your workloads are deployed in multiple regions, you can create a RegionlessDB cluster to access databases from the nearest region. As shown in **Figure 22-3**, a RegionlessDB cluster contains a primary instance and two standby instances. Read requests are sent to a standby instance in the nearest region, and write requests are automatically forwarded from the nearest region to the primary instance. After data is written to the primary instance, the data is synchronized to all standby instances, reducing the cross-region network latency.

Figure 22-3 Remote multi-active principle



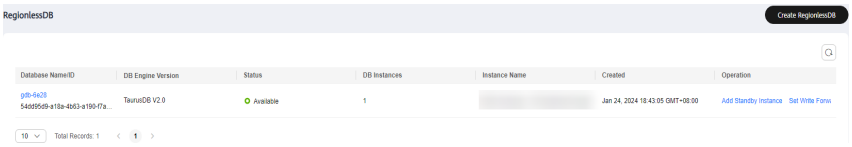
Constraints

For details, see **Constraints**.

Step 1: Create a RegionlessDB Cluster

1. **Log in to the management console.**
2. Click in the upper left corner and select a region and project.
3. Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.
4. On the **RegionlessDB** page, click **Create RegionlessDB** in the upper right corner.

Figure 22-4 Creating a RegionlessDB cluster



5. In the **Create RegionlessDB** dialog box, configure **RegionlessDB Name**, **Primary Instance Region**, and **Primary Instance**.

Figure 22-5 Configuring the RegionlessDB cluster information

Table 22-2 Parameter description

Parameter	Description
RegionlessDB Name	The name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
Primary Instance Region	Select a region where the primary instance is located.
Primary Instance	Select an existing DB instance as the primary instance of the RegionlessDB cluster.

6. Click **OK**.
7. After the primary instance is created, view and manage it.

   During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the primary instance is **Available**, you can use the instance.

Step 2: Add a Standby Instance

1. On the **RegionlessDB** page, locate the RegionlessDB cluster.

2. Click **Add Standby Instance** in the **Operation** column.

Figure 22-6 Adding a standby instance

3. On the displayed page, configure related parameters.

Table 22-3 Basic information

Parameter	Description
Region	Region where the standby instance is deployed. Products in different regions cannot communicate with each other through a private network. After a DB instance is purchased, the region cannot be changed.
Creation Method	Create new
DB Instance Name	The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	TaurusDB
Kernel Version	Kernel version of the standby instance. The kernel version must be 2.0.46.231000 or later. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** To specify the kernel version, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
DB Instance Type	Only **Cluster** can be selected. There are 2 to 10 read replicas in a cluster instance in the RegionlessDB cluster.
Storage Type	Shared
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. • **Single AZ**: The primary node and read replicas are deployed in the same AZ. • **Multi-AZ**: The primary node and read replicas are deployed in different AZs to ensure high reliability.

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.
Instance Specifications	For details about the specifications supported by TaurusDB, see **Instance Specifications**. TaurusDB is a cloud-native database that uses the shared storage. To ensure workload stability in high read/write pressure, the system controls the read/write peaks of DB instances based on instance specifications. For details about how to select instance specifications, see **Performance White Paper**.
CPU Architecture	The CPU architecture can be x86 or Kunpeng. Under a CPU architecture, you need to select vCPUs and memory of the instance.
Nodes	All nodes of the standby instance are read replicas. You can apply for a maximum of 10 read replicas at a time for a pay-per-use instance. After an instance is created, you can add read replicas as required. Up to 15 read replicas can be created for a standby instance in a cluster.
Storage	Storage will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.

VPC	• A dedicated virtual network in which your TaurusDB instance is located. It isolates networks for different workloads. You can select an existing VPC or create a VPC. For details about how to create a VPC, see **Creating a VPC**.
	If no VPC is available, TaurusDB allocates a VPC to you by default.
	CAUTION
	  • Ensure that the VPC selected for the standby instance is connected to the VPC selected for the primary instance through a VPN.
	  • After a TaurusDB instance is created, the VPC cannot be changed.
	• A subnet provides dedicated network resources that are logically isolated from other networks for network security. A private IP address is automatically assigned when you create a DB instance. You can also enter an idle private IP address in the subnet CIDR block.

Parameter	Description
Security Group	It can enhance security by controlling access to 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 you by default. To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. Configure private network security group rules to ensure that the primary and standby instances in a cluster can communicate with each other.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. You can modify the instance parameters as required after the instance is created. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters in a Parameter Template**.
Enterprise Project	This parameter is for enterprise users. To specify it, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

NOTE

The instance password and table name case sensitivity are the same as those of the primary instance. You do not need to set them separately.

4. Click **Next**.
5. Confirm the information and click **Submit**.
6. Go to the **Instances** page to view and manage the instance.

During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the instance is **Available**, you can use the instance.

If there is a large amount of data in the primary instance, it may take a long time to complete a full backup during standby instance creation.

Step 3: Enable Write Forwarding

In normal cases, after a RegionlessDB cluster is created, the primary instance receives and processes read and write requests, and standby instances receive only read requests. After write forwarding is enabled, standby instances can receive write requests and then forward them to the primary instance for processing. After data is written to the primary instance, the data is synchronized to all standby instances. Write forwarding simplifies the data write process. You can directly connect a database service through a standby instance's IP address to perform read and write operations. In addition, consistency is ensured and the nearby read is not affected.

Constraints:

- When write forwarding is enabled, user **_@gdb_WriteForward@_** is created. Do not modify or delete the user, or write forwarding cannot run properly.
- For commands that can be implicitly committed, if write forwarding is not supported, the transactions corresponding to the current node and primary node are automatically committed.
- For the global consistency level, before accessing data for the first time, each transaction needs to use a connection in the session pool to obtain a data point (LSN) from the primary node. If no sessions are available, the command for reading data may fail.
- If there is a connection error when a user uses a session for write forwarding and the user is in a multi-statement transaction, the server proactively closes the connections to the client and the primary node, ensuring that the client can detect the error.
- The versions of the primary and standby instances must be the latest.
- Write operations are finally forwarded to and processed by the primary node. If a temporary table with the same name exists in the given database of the primary and read replicas, the data on the primary node is used.
- If there is a failover for a standby instance in a RegionlessDB cluster, the write forwarding parameters (**rds_open_write_forwarding** and **rds_write_forward_read_consistency**) are restored to the default values.
- **Table 22-4** lists all supported and unsupported scenarios.

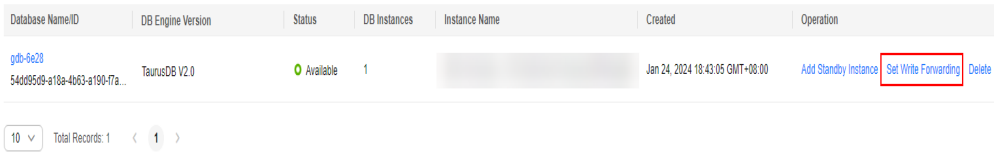
Table 22-4 Scenarios supported and not supported by write forwarding

Constraints	Description
Supported scenarios	• Write forwarding is only available when the transaction isolation level of the standby instances is RR. • Write forwarding supports the following commands: – SQLCOM_UPDATE – SQLCOM_INSERT – SQLCOM_DELETE – SQLCOM_INSERT_SELECT – SQLCOM_REPLACE – SQLCOM_REPLACE_SELECT – SQLCOM_DELETE_MULTI – SQLCOM_UPDATE_MULTI – SQLCOM_ROLLBACK If an unsupported command is executed, the following error information is displayed. `ERROR xxx (yyy): This version of MySQL doesn't yet support 'operation with write forwarding'.` **operation** indicates the operation type that is not supported.
Unsupported scenarios	• In the current version, WARNING and RECORD information cannot be displayed when a standby instance forwards write requests. • In the current version, SQL requests that are being executed cannot be interrupted when a standby instance forwards write requests. • SELECT FOR UPDATE statements are not supported. • EXPLAIN write forwarding statements are not supported. • The statements for write forwarding cannot contain SET VARIABLE. • SAVEPOINT is not supported when write forwarding is enabled. • Write forwarding is not supported in XA transactions. • Currently, START TRANSACTION READ WRITE is not supported. You can directly use START TRANSACTION to test write forwarding. • Write forwarding is not supported in stored procedures. • When write forwarding is enabled, temporary tables cannot be created. To create temporary tables, disable write forwarding temporarily.

Step 1 On the **RegionlessDB** page, locate the RegionlessDB cluster.

Step 2 Click **Set Write Forwarding** in the **Operation** column to create a write forwarding account.

Figure 22-7 Creating a write forwarding account

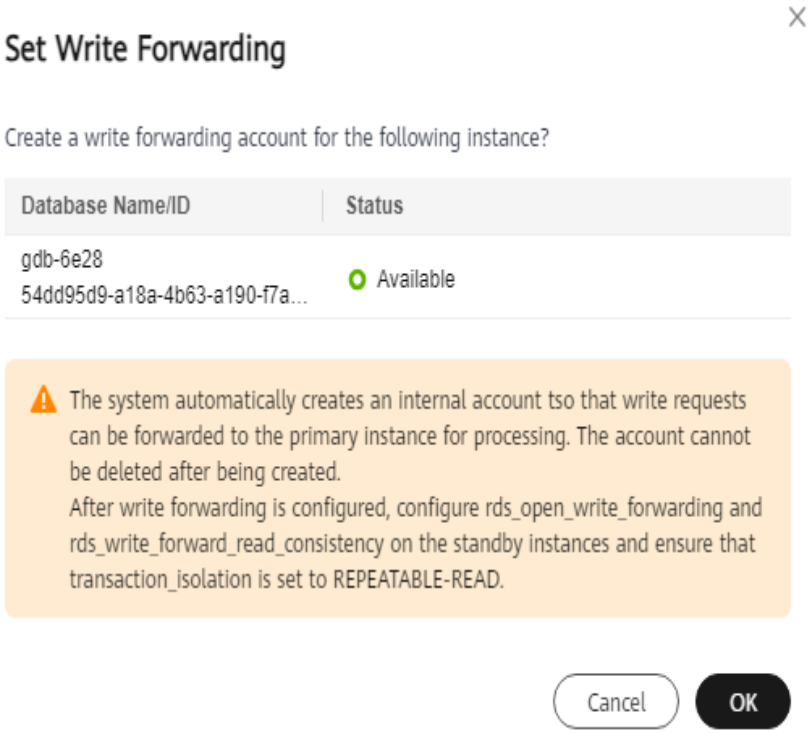


NOTE

The system automatically creates an internal account (**_@gdb_WriteForward@_**) so that write requests can be forwarded to the primary instance for processing. You cannot modify or delete the internal account, or write forwarding will be affected.

Step 3 In the **Set Write Forwarding** dialog box, confirm the information and click **OK**.

Figure 22-8 Setting write forwarding



Step 4 On the **Instances** page, click the name of the standby instance in the RegionlessDB cluster.

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

Step 6 Search for **rds_open_write_forwarding** in the upper right corner of the **Parameters** page and change its value to **ON**.

Step 7 Click **Save** in the upper left corner to enable write forwarding.

Step 8 Search for **rds_write_forward_read_consistency** in the upper right corner of the **Parameters** page and change the read consistency level of write forwarding.

You can modify the parameters to set the read consistency range. For details, see **Table 22-5**.

- If read consistency is required, you are advised to set the consistency level to SESSION. The consistency level GLOBAL will cause a large extra cost for all read requests. For example, if any client is used to connect to TaurusDB and the GLOBAL level is used, the time for accessing the MySQL command line is prolonged.
- The read consistency level in write forwarding cannot be changed to SESSION in a transaction.
- Before enabling write forwarding, ensure that the transaction isolation levels of standby instances are RR.
- When write forwarding is enabled, the transaction isolation level of the current session cannot be changed.
- The read consistency level cannot be changed in a transaction.

Table 22-5 Parameter description

Parameter	Description
NONE	Write forwarding is disabled.
EVENTUAL	Results of write operations are not visible until the write operations are performed on the primary instance. The query does not wait for data synchronization between primary and standby instances to complete, so data that is not updated may be read.
SESSION	All queries executed by a standby instance with write forwarding enabled see the results of all data writes performed in this session. The queries wait for the results of forwarded write operations to be replicated.
GLOBAL	A session can view all committed changes of all sessions and instances in a RegionlessDB cluster. The query may wait for a certain period, which is related to the replication latency.

Step 9 Click **Save** in the upper left corner.

----End

Step 4: Connect to the RegionlessDB Cluster for Service Management

After a RegionlessDB cluster is created, no unified connection address is provided. The primary and standby instances in the RegionlessDB cluster provide independent connection addresses. You can use the nearest primary or standby

instance based on the service access region to connect to the RegionlessDB cluster. The RegionlessDB cluster automatically forwards write requests to the primary instance for processing and read requests to the instance of the nearest region for processing.

Example:

1. Connect to the primary instance and write data to the database.
```
mysql> CREATE DATABASE mydatabase;
mysql> CREATE TABLE orders (order_id INT PRIMARY KEY, customer_name VARCHAR(255),
order_date DATE);
mysql> INSERT INTO orders (order_id, customer_name, order_date) VALUES (1, 'UserA', '2023-12-18'),
(2, 'UserB', '2023-12-17'), (3, 'UserC', '2023-12-16');
```
2. Use the nearest standby instance to access the database and query the data written in **1**.
```
mysql> select * from mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|        1 | UserA         | 2023-12-18 |
|        2 | UserB         | 2023-12-17 |
|        3 | UserC         | 2023-12-16 |
+----------+---------------+------------+
```
3. Connect to database through the primary instance and run the following SQL statements to create the **mydatabase** database and **orders** table.
```
mysql> CREATE DATABASE mydatabase;
Query OK, 1 row affected (0.00 sec)
mysql> USE mydatabase;
Database changed
mysql> CREATE TABLE orders (order_id INT PRIMARY KEY, customer_name VARCHAR(255),
order_date DATE);
Query OK, 0 rows affected (0.01 sec)
```
4. Connect to the database from a standby instance, run the following SQL statements to write three data records to the **orders** table, and query the written data.
```
mysql> INSERT INTO orders (order_id, customer_name, order_date) VALUES (1, 'UserA', '2023-12-18'),
(2, 'UserB', '2023-12-17'), (3, 'UserC', '2023-12-16');
Query OK, 3 rows affected (0.00 sec)
mysql> SELECT * FROM mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|        1 | UserA         | 2023-12-18 |
|        2 | UserB         | 2023-12-17 |
|        3 | UserC         | 2023-12-16 |
+----------+---------------+------------+
3 rows in set (0.01 sec)
```
5. Connect to the database through the primary instance and run the following SQL statements to query the data inserted by the standby instance in **4**.
```
mysql> SELECT * FROM mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|        1 | UserA         | 2023-12-18 |
|        2 | UserB         | 2023-12-17 |
|        3 | UserC         | 2023-12-16 |
+----------+---------------+------------+
3 rows in set (0.00 sec)
```

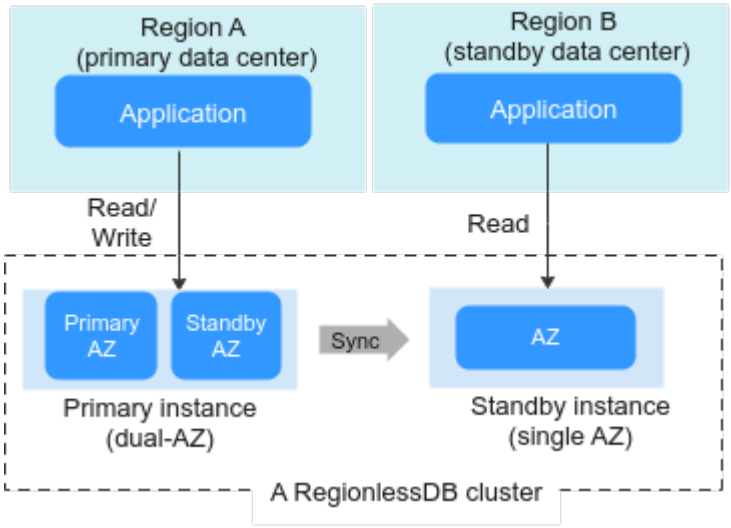

22.3 Using a RegionlessDB Cluster for Remote DR

Scenarios

If there is a region-level fault on the primary instance, workloads can be switched to a standby instance for remote DR.

As shown in **Figure 22-9**, a RegionlessDB cluster contains a primary instance deployed across two AZs and a standby instance deployed in a single AZ. If the primary AZ of the primary instance is faulty, workloads are preferentially switched to the standby AZ. If both the primary and standby AZs of the primary instance are faulty, workloads are switched to the standby instance.

Figure 22-9 Remote DR principle



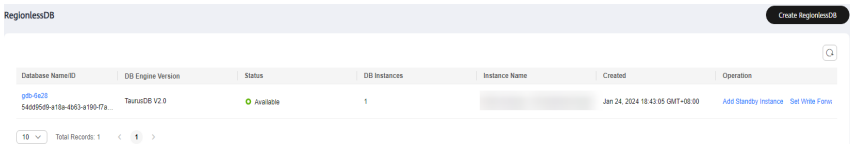
Constraints

For details, see **Constraints**.

Step 1: Create a RegionlessDB Cluster

1. **Log in to the management console.**
2. Click in the upper left corner and select a region and project.
3. Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.
4. On the **RegionlessDB** page, click **Create RegionlessDB** in the upper right corner.

Figure 22-10 Creating a RegionlessDB cluster



5. In the **Create RegionlessDB** dialog box, configure **RegionlessDB Name**, **Primary Instance Region**, and **Primary Instance**.

Figure 22-11 Configuring the RegionlessDB cluster information

Table 22-6 Parameter description

Parameter	Description
RegionlessDB Name	The name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
Primary Instance Region	Select a region where the primary instance is located.
Primary Instance	Select an existing DB instance as the primary instance of the RegionlessDB cluster.

6. Click **OK**.
7. After the primary instance is created, view and manage it.

   During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the primary instance is **Available**, you can use the instance.

Step 2: Add a Standby Instance

1. On the **RegionlessDB** page, locate the RegionlessDB cluster.
2. Click **Add Standby Instance** in the **Operation** column.

Figure 22-12 Adding a standby instance

3. On the displayed page, configure related parameters.

Table 22-7 Basic information

Parameter	Description
Region	Region where the standby instance is deployed. Products in different regions cannot communicate with each other through a private network. After a DB instance is purchased, the region cannot be changed.
Creation Method	Create new
DB Instance Name	The name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	TaurusDB
Kernel Version	Kernel version of the standby instance. The kernel version must be 2.0.46.231000 or later. For details about the updates in each kernel version, see **TaurusDB Kernel Version Release History**. **NOTE** To specify the kernel version, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
DB Instance Type	Only **Cluster** can be selected. There are 2 to 10 read replicas in a cluster instance in the RegionlessDB cluster.
Storage Type	Shared
AZ Type	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Some regions support both single-AZ and multi-AZ deployment and some only support single-AZ deployment. • **Single AZ**: The primary node and read replicas are deployed in the same AZ. • **Multi-AZ**: The primary node and read replicas are deployed in different AZs to ensure high reliability.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. The time zone is selected during instance creation and cannot be changed after the instance is created.

Parameter	Description
Instance Specifications	For details about the specifications supported by TaurusDB, see **Instance Specifications**. TaurusDB is a cloud-native database that uses the shared storage. To ensure workload stability in high read/write pressure, the system controls the read/write peaks of DB instances based on instance specifications. For details about how to select instance specifications, see **Performance White Paper**.
CPU Architecture	The CPU architecture can be x86 or Kunpeng. Under a CPU architecture, you need to select vCPUs and memory of the instance.
Nodes	All nodes of the standby instance are read replicas. You can apply for a maximum of 10 read replicas at a time for a pay-per-use instance. After an instance is created, you can add read replicas as required. Up to 15 read replicas can be created for a standby instance in a cluster.
Storage	Storage will be scaled up dynamically based on the amount of data that needs to be stored, and is billed hourly on a pay-per-use basis.

VPC	• A dedicated virtual network in which your TaurusDB instance is located. It isolates networks for different workloads. You can select an existing VPC or create a VPC. For details about how to create a VPC, see **Creating a VPC**.
	If no VPC is available, TaurusDB allocates a VPC to you by default.
	CAUTION
	  • Ensure that the VPC selected for the standby instance is connected to the VPC selected for the primary instance through a VPN.
	  • After a TaurusDB instance is created, the VPC cannot be changed.
	• A subnet provides dedicated network resources that are logically isolated from other networks for network security. A private IP address is automatically assigned when you create a DB instance. You can also enter an idle private IP address in the subnet CIDR block.

Parameter	Description
Security Group	It can enhance security by controlling access to 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 you by default. To ensure subsequent database connection and access, you need to allow all IP addresses to access your DB instance through port 3306 and over ICMP. Configure private network security group rules to ensure that the primary and standby instances in a cluster can communicate with each other.
Parameter Template	Contains engine configuration values that can be applied to one or more instances. You can modify the instance parameters as required after the instance is created. If you use a custom parameter template when creating a DB instance, the specification-related parameters in the custom template will not be applied. Instead, the default values are used. For details, see **What Parameters Should I Pay Attention to When Creating a DB Instance?** After a DB instance is created, you can adjust its parameters as needed. For details, see **Modifying Parameters in a Parameter Template**.
Enterprise Project	This parameter is for enterprise users. To specify it, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. An enterprise project provides a way to manage cloud resources and enterprise members on a project-by-project basis. You can select an enterprise project from the drop-down list. The default project is **default**.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each DB instance can have up to 20 tags. After a DB instance is created, you can view its tag details on the **Tags** tab. For details, see **Tag Management**.

NOTE

The instance password and table name case sensitivity are the same as those of the primary instance. You do not need to set them separately.

4. Click **Next**.
5. Confirm the information and click **Submit**.
6. Go to the **Instances** page to view and manage the instance.

During the creation process, the instance status is **Creating**. To view the detailed progress and result of the creation, go to the **Task Center** page. After the status of the instance is **Available**, you can use the instance.

If there is a large amount of data in the primary instance, it may take a long time to complete a full backup during standby instance creation.

Step 3: Connect to a Standby Instance for Workload Management

Select the nearest standby instance to access the database based on the workload access area.

For example, use a standby instance to access the database and query data.

```
mysql> select * from mydatabase.orders;
+----------+---------------+------------+
| order_id | customer_name | order_date |
+----------+---------------+------------+
|        1 | UserA         | 2023-12-18 |
|        2 | UserB         | 2023-12-17 |
|        3 | UserC         | 2023-12-16 |
+----------+---------------+------------+
```

22.4 Performing a Failover in a RegionlessDB Cluster

A RegionlessDB cluster consists of multiple TaurusDB instances in different regions around the world. The cluster in each region inherits the original same-region HA feature. A RegionlessDB cluster provides cross-region failover capabilities.

Failovers

If the primary instance in a RegionlessDB cluster fails and cannot be restored, usually due to a regional outage, a failover is triggered to promote the standby instance with the latest data from all available standby instances to the primary instance.

A failover may result in some data loss, depending on the replication latency between the primary and standby instances during the failover.

Generally, a failover can be complete within several minutes. However, after a failover is performed, the original primary instance needs to be rebuilt as a standby instance before being added back to the entire cluster. The rebuilding process may take dozens of minutes to several hours, depending on the data volume and network conditions between regions.

If a failover is required, **submit a service ticket**.

Other Operations and Checks

The primary and standby instances are independent TaurusDB instances. During a failover, the configurations between the primary and standby instances are not exchanged. To prevent performance and compatibility issues caused by different configuration parameters, you are advised to check for any differences in configuration items between the primary and standby instances after a failover.

- Check the read/write addresses of your application.

During a failover, the read/write addresses of instances are not exchanged. You need to check whether the read/write address of your application is as expected.

During a failover, you can configure an application to use the read/write address of the new primary instance. After the faulty instance is rebuilt, reconfigure the read/write address of the application.

- Check the write forwarding configurations of standby instances. For details, see **Step 3: Enable Write Forwarding**.
- Check the configurations of a parameter template. For details, see **Modifying Parameters of a DB Instance**.
- Configure monitoring alarms. For details, see **Metrics and Alarms**.

22.5 Removing a Standby Instance from a RegionlessDB Cluster

Only standby instances can be removed from a RegionlessDB cluster. After a standby instance is removed from a RegionlessDB cluster, data of the primary instance will not be synchronized to the standby instance.

CAUTION

After a standby instance is removed from a RegionlessDB cluster, the standby instance will be permanently deleted. Exercise caution when performing this operation.

You can remove a standby instance from a RegionlessDB cluster.

Constraints

For details, see **Constraints**.

Procedure

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

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

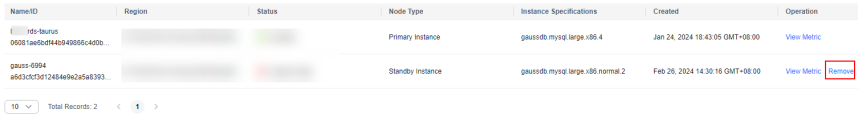
Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **RegionlessDB** page, locate the RegionlessDB cluster.

Step 5 Click the name of the cluster to view its details.

Step 6 In the instance list area, locate a standby instance and click **Remove** in the **Operation** column.

Figure 22-13 Accessing the instance list page

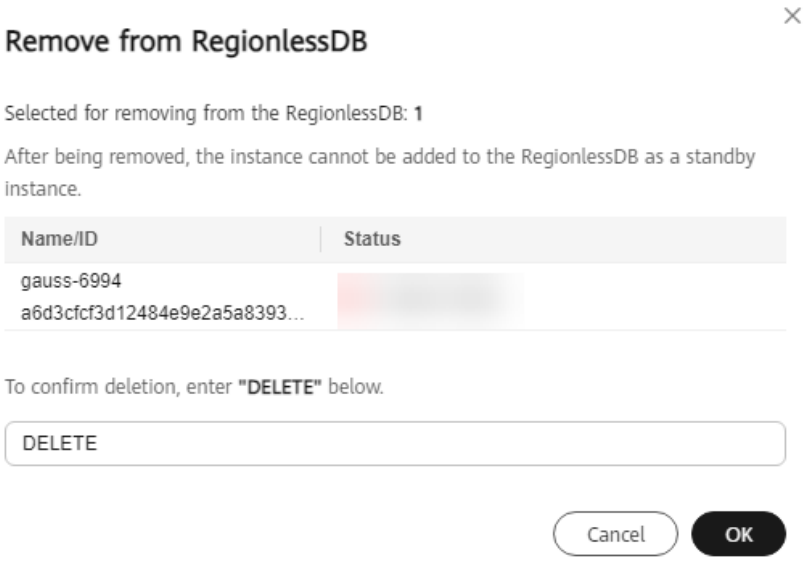


Step 7 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

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

Step 8 In the displayed dialog box, enter **DELETE** in the text box and click **OK**.

Figure 22-14 Removing a standby instance from RegionlessDB



To view the detailed progress and result of the task, go to the **Task Center** page.

----End

22.6 Deleting a RegionlessDB Cluster

You can delete a RegionlessDB cluster.

Constraints

- Before deleting a RegionlessDB cluster, ensure that all standby instances have been removed from it. For details about how to remove a standby instance, see **Removing a Standby Instance from a RegionlessDB Cluster**.
- For more constraints, see **Constraints**.

Procedure

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

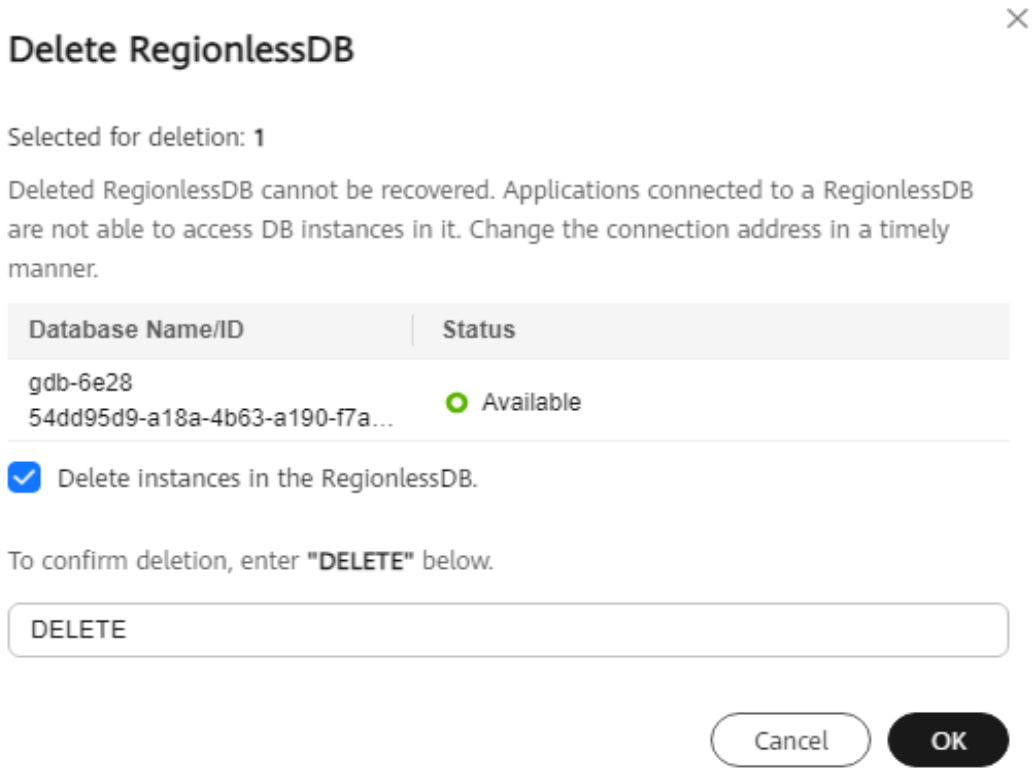
Step 5 Locate a cluster and click **Delete** in the **Operation** column.

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

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

Step 7 In the **Delete RegionlessDB** dialog box, select or deselect **Delete instances in RegionlessDB** as required, enter **DELETE** in the text box, and click **OK**.

Figure 22-15 Deleting a RegionlessDB cluster



Step 8 Refresh the RegionlessDB cluster list later to confirm that the deletion was successful.

To view the detailed progress and result of the task, go to the **Task Center** page.

----End

22.7 Viewing the Replication Latency and Traffic of a RegionlessDB Cluster

After a RegionlessDB cluster is created, you can monitor the database status and performance based on related metrics.

Viewing the Replication Latency and Traffic on the Console

Step 1 Log in to the management console.

Step 2 Click in the upper left corner and select the region and project of the primary instance.

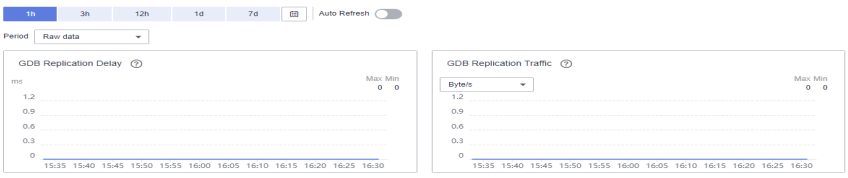
Step 3 Click and select **Cloud Eye** under **Management & Governance**.

Step 4 In the navigation pane, choose **Cloud Service Monitoring** > **TaurusDB**.

Step 5 Click in the front of a RegionlessDB cluster. Locate a standby instance and click **View Metrics** in the **Operation** column.

- You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 1 day, or last 7 days.

Figure 22-16 Viewing RegionlessDB metrics



- For details about metrics supported by RegionlessDB clusters, see **Table 22-8**. For details about the metrics of primary and standby instances, see **TaurusDB Metrics**.

Table 22-8 RegionlessDB cluster metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gdb_replication_latency	GDB Replication Delay	Data replication latency of the measured object	≥0 ms	Standby instances in a RegionlessDB cluster	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
gdb_replication_capacity	GDB Replication Traffic	Data replication traffic of the measured object	≥0 bytes/s	Standby instances in a RegionlessDB cluster	1 minute

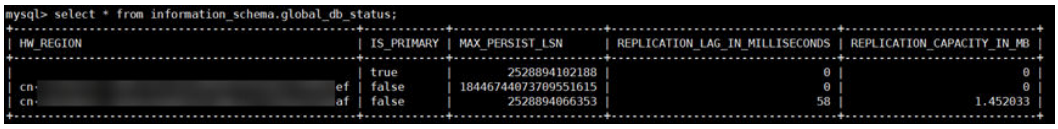
----End

Viewing the Replication Latency and Traffic Using SQL Commands

Use a MySQL client tool to connect to the TaurusDB instance and run the following command to query the RegionlessDB status:

```
mysql> select * from information_schema.global_db_status;
```

Figure 22-17 Querying the RegionlessDB status



In the command output, each row indicates an instance in the RegionlessDB cluster (the first row indicates the primary instance and other rows indicate the standby instances). For details about the parameters contained in each row, see **Table 22-9**.

Table 22-9 Parameter description

Parameter	Description
HW_REGION	Region code of the standby instance. The first row in the table is the primary instance, and the region code of the primary instance is an empty string.
IS_PRIMARY	Whether the instance is the primary instance. **true**: it is the primary instance. **false**: It is the standby instance.
MAX_PERSIST_LSN	Maximum LSN of the current redo logs of the instance that have been persisted to the shared storage.
REPLICATION_LAG_IN_MILLISECONDS	Latency from the time when data is written to the primary instance to the time when data can be read from the standby instance, in ms. The replication latency of the primary instance is 0.

Parameter	Description
REPLICATION_CAPACITY_IN_MB	Throughput of data replication from the primary instance to a standby instance, in MB/s. The replication throughput of the primary instance is 0.

23 Metrics and Alarms

23.1 TaurusDB Metrics

Function

You can monitor the status of your instances using Cloud Eye. This section describes the TaurusDB metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions.

The monitoring interval can be 1 minute, 1 second, or 5 seconds. The default monitoring interval is 1 minute. To enable Monitoring by Seconds, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

TaurusDB Instance Metrics

Namespace: SYS.GAUSSDB

Table 23-1 TaurusDB instance metrics

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100	%	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql002_mem_util	Memory Usage	Memory usage of the monitored object	0–100	%	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥0	Bytes/s	1024 (IEC)	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥0	Bytes/s	1024 (IEC)	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql006_conn_count	Total Connections	Total number of connections that attempt to connect to the TaurusDB server	≥0	Count	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql007_conn_active_count	Current Active Connections	Number of active connections	≥0	Count	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql008_qps	QPS	Query times of SQL statements (including DDL, DML, SHOW, SET statements and storage procedures) per second	≥0	Times/s	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql009_tps	TPS	Execution times of submitted and rollback transactions per second	≥0	Times/s	N/A	TaurusDB instance nodes	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	0–100	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0–100	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to all data in the InnoDB buffer	0–100	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥0	Byte/s	1024 (IEC)	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql014_innodb_writes	InnoDB Write Throughput	Bytes written to pages by InnoDB per second. TaurusDB only writes data to temporary tables.	≥0	Bytes/s	1024 (IEC)	TaurusDB instance nodes	1 minute
gaussdb_mysql017_innodb_log_write_req_count	InnoDB Log Write Request Frequency	Number of InnoDB log write requests per second	≥0	Times/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql019_innodb_log_writes	InnoDB Log Writes	Number of physical writes to the InnoDB redo log file	≥0	Count	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql020_temp_tbl_count	Temporary Tables	Number of temporary tables automatically created on disks when TaurusDB statements are executed	≥0	Count	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second
gaussdb_mysql035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–128*1024	GB	1024 (IEC)	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql060_rx_errors	Error Rate of Received Packets	Ratio of the number of error packets to the total number of received packets during the monitoring period	0–100	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql061_rx_dropped	Loss Rate of Received Packets	Ratio of the number of lost packets to the total number of received packets during the monitoring period	0–100%	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql062_tx_errors	Error Rate of Sent Packets	Ratio of the number of error packets to the total number of sent packets during the monitoring period	0–100%	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql063_tx_dropped	Loss Rate of Sent Packets	Ratio of the number of lost packets to the total number of sent packets during the monitoring period	0–100%	%	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql072_conn_usage	Connection Usage	Percent of used TaurusDB connections to the total number of connections	0–100%	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql074_slow_queries	Slow Query Logs	Number of TaurusDB slow query logs generated per minute	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql077_replication_delay	Replication Delay	Delay between the primary node and read replicas **NOTE** This metric is used only for read replicas.	≥0	s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql104_dfv_write_delay	Storage Write Delay	Average delay of writing data to the storage layer in a specified period	≥0	ms	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql105_dfv_read_delay	Storage Read Delay	Average delay of reading data from the storage layer in a specified period	≥0	ms	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql107_comdml_ins_and_ins_sel_count	INSERT and INSERT_SELECT Statements per Second	Number of INSERT and INSERT_SELECT statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql108_com_commit_count	COMMIT Statements per Second	Number of COMMIT statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql109_com_rollback_count	ROLLBACK Statements per Second	Number of ROLLBACK statements executed per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql110_innodb_bufpool_reads	InnoDB Storage Layer Read Requests per Second	Number of times that InnoDB reads data from the storage layer per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql111_innodb_bufpool_read_requests	InnoDB Read Requests per Second	Number of InnoDB read requests per second	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
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	Count	N/A	TaurusDB instance nodes	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	Count	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql116_innodb_bufpool_read_ahead_rnd	InnoDB Bufpool Read Ahead Rnd	Number of random read-aheads initiated by InnoDB	≥0	Count	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql117_innodb_pages_read	InnoDB Pages Read	Number of pages read from the InnoDB buffer pool by operations on InnoDB tables	≥0	Count	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql118_innodb_pages_written	InnoDB Pages Written	Number of pages written by operations on InnoDB tables	≥0	Count	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql119_disk_used_ratio	Disk Usage	Disk usage of the monitored object	0–100	%	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql120_innodb_buffer_pool_bytes_data	Total Bytes of Buffer Pool	Total number of bytes in the InnoDB buffer pool containing data	≥0	Bytes	1024 (IEC)	TaurusDB instance nodes	1 minute
gaussdb_mysql121_innodb_row_lock_time	Row Lock Time	Total time spent in acquiring row locks for InnoDB tables	≥0	ms	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql122_innodb_row_lock_waits	Row Lock Waits	Number of times operations on InnoDB tables had to wait for a row lock	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql123_sort_range	Sorts Using Ranges	Number of sorts that were done using ranges	≥0 n	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql124_sort_rows	Sorted Rows	Number of sorted rows	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql125_sort_scan	Sorts by Scanning Tables	Number of sorts that were done by scanning tables.	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql126_table_open_cache_hits	Hits for Open Tables Cache Lookups	Number of hits for open tables cache lookups	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql127_table_open_cache_misses	Misses for Open Tables Cache Lookups	Number of misses for open tables cache lookups	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql128_long_trx_count	Long-Running Transactions	Number of long transactions that are not closed	≥0	Count	N/A	TaurusDB instance nodes	150s
gaussdb_mysql342_iostat_iops_write	I/O Write IOPS	I/O write IOPS	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql344_iostat_iops_read	I/O Read IOPS	I/O read IOPS	≥0	Count/s	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql346_iostat_throughput_write	I/O Write Bandwidth	Disk write bandwidth per second	≥0	Bytes/s	1024 (IEC)	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql348_iostat_throughput_read	I/O Read Bandwidth	Disk read bandwidth per second	≥0	Bytes/s	1024 (IEC)	TaurusDB instance nodes	1 minute
gaussdb_mysql371_taurus_binlog_total_file_counts	Binlog Files	Number of TaurusDB binlog files	≥0	Count	N/A	TaurusDB instance nodes	5 minutes
gaussdb_mysql378_create_temp_tbl_per_min	Temporary Tables Created per Minute	Number of temporary tables automatically created on disks per minute when TaurusDB statements are executed	≥0	Count/min	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql386_undo_spaces_trx_count	Existing Transactions in Undo Space	Number of transactions that are not cleared in the undo space	≥0	Count	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql348_taurus_throttle_slice_num	Write Traffic Control	Whether write traffic control is triggered for a DB instance. If its value is greater than 0, write traffic control is triggered. Its value indicates the number of slices whose traffic is limited.	≥0	Count	N/A	TaurusDB instance nodes	1 minute
gaussdb_mysql339_taurus_sal_flow_control_instance_read_page_throttle	Read Traffic Control	Whether read traffic control is triggered for a DB instance. If its value is greater than 0, read traffic control is triggered. Its value indicates the number of read pages whose traffic is limited.	≥0	Count	N/A	TaurusDB instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
gaussdb_mysql430_taurus_innodb_rpl_millisecond_behind_master	Replication Delay Millisecond-level	Millisecond-level replication delay in an instance. It is only used for read replicas. **NOTE** This metric is only available to instances whose kernel version is 2.0.54.240600 or later.	≥0	Count	N/A	TaurusDB instance nodes	1 minute 5 seconds 1 second

Proxy Instance Metrics

Namespace: SYS.DBPROXY

Table 23-2 Proxy instance metrics

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_proxy_frontend_connections	Frontend Connections	Number of connections between applications and the proxy	≥0	Count	N/A	Proxy instance nodes	1 minute
rds_proxy_backend_connections	Backend Connections	Number of connections between the proxy and TaurusDB databases	≥0	Count	N/A	Proxy instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_proxy_average_response_time	Average Response Time	Average response time	≥0	ms	N/A	Proxy instance nodes	1 minute
rds_proxy_query_per_seconds	QPS	Query times of SQL statements	≥0	Count	N/A	Proxy instance nodes	1 minute
rds_proxy_read_query_proportions	Read Proportion	Proportion of read requests to total requests	0–100	%	N/A	Proxy instance nodes	1 minute
rds_proxy_write_query_proportions	Write Proportion	Proportion of write requests to total requests	0–100	%	N/A	Proxy instance nodes	1 minute
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100	%	N/A	Proxy instance nodes	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0–100	%	N/A	Proxy instance nodes	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥0	byte/s	1024 (IEC)	Proxy instance nodes	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥0	byte/s	1024 (IEC)	Proxy instance nodes	1 minute

Metric ID	Metric	Metric Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_proxy_frontend_connection_creation	Front-End Connections Created per Second	Number of connections created per second between the database proxy and applications	≥0	Counts/sec	N/A	Proxy instance nodes	1 minute
rds_proxy_multi_statement_query	Multi-Statement Queries per Second	Number of multi-statements executed in transactions per second	≥0	Counts/sec	N/A	Proxy instance nodes	1 minute
rds_proxy_transaction_query	Transaction Queries per Second	Number of SELECT statements executed in transactions per second	≥0	Counts/sec	N/A	Proxy instance nodes	1 minute

Dimension

Table 23-3 Metric dimension

Key	Value
gaussdb_mysql_instance_id	TaurusDB instance ID
gaussdb_mysql_node_id	TaurusDB instance node ID
dbproxy_instance_id	Proxy instance ID
dbproxy_node_id	Proxy node ID

23.2 Viewing TaurusDB Metrics

Scenarios

Cloud Eye monitors TaurusDB running statuses. You can view TaurusDB metrics 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 for problematic SQL statements to optimize them.

Prerequisites

- DB instances are running properly.

  Metrics of the DB instances that are faulty or have been deleted cannot be displayed on the Cloud Eye console, but you can view them after the DB instances are rebooted or become available.

  CAUTION

  If a DB instance has been faulty for 24 hours, Cloud Eye assumes that the instance no longer exists and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the instance.

- DB instances have kept running properly for about 10 minutes.

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

- To view metrics of a proxy instance, ensure that read/write splitting has been enabled for the DB instance. For details, see **Creating a Proxy Instance for Read/Write Splitting**.

Viewing DB Instance Metrics

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

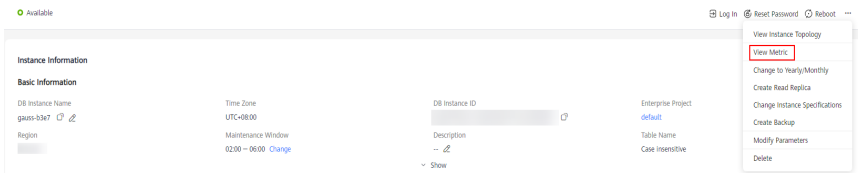
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Alternatively, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click and choose **View Metric**.

Figure 23-1 Viewing metrics on the **Basic Information** page



To view metrics of a node, locate the node in the **Node List** area and click **View Metrics** in the **Operation** column.

Step 5 On the displayed Cloud Eye page, view metrics.

You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 1 day, or last 7 days.

----End

Viewing Proxy Instance Metrics

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

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

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

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

Step 5 Choose **Database Proxy** in the navigation pane, locate a proxy instance, and click **View Metrics** in the **Operation** column.

You can view the performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 1 day, or last 7 days.

----End

Viewing Real-Time Metrics

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

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

Step 3 Click  in the upper left corner of the page, choose **Databases** > **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

23.3 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.

Billing

TaurusDB provides monitoring every 60 seconds for free, but you are billed for Monitoring by Seconds. Its pricing is listed on a per-hour basis, but bills are calculated based on actual usage.

Table 23-4 Price details

Region	Monitoring Interval	Pay-per-Use (USD/Hour)
CN East-Shanghai1, CN North-Beijing4, CN South-Guangzhou, CN Southwest-Guiyang1, CN North-Ulanqab1, and CN South-Guangzhou-InvitationOnly	1s	0.024
	5s	0.012
AP-Singapore, AP-Jakarta, RU-Moscow2, CN-Hong Kong, AP-Bangkok, and TR-Istanbul	1s	0.032
	5s	0.016
LA-Sao Paulo1	1s	0.054
	5s	0.027

Enabling Monitoring by Seconds

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

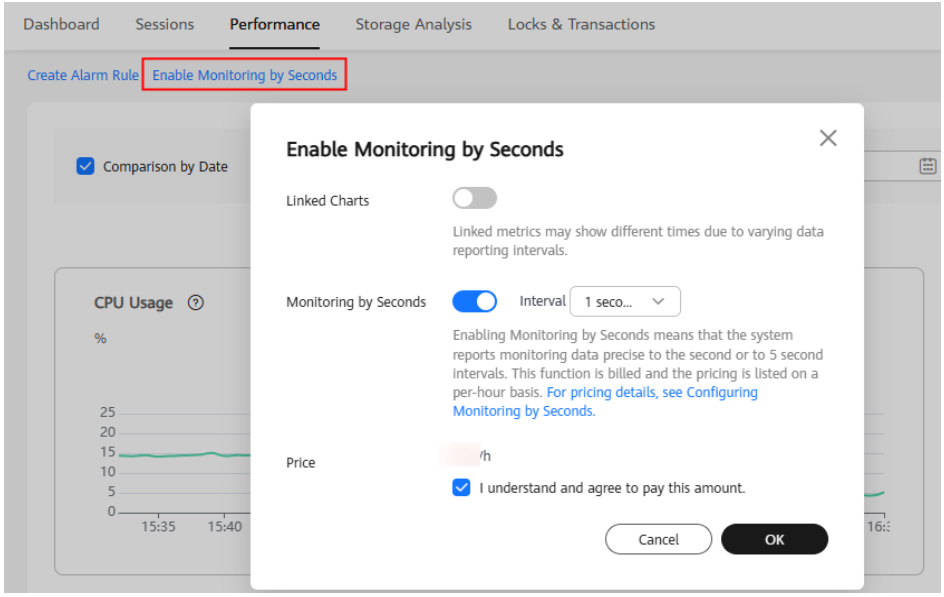
Step 6 Click **Performance**.

Step 7 In the upper part of the page, click **Enable Monitoring by Seconds**.

Step 8 In the displayed dialog box, click next to **Monitoring by Seconds**, select a collection interval, and click **OK**.

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

Figure 23-2 Enabling Monitoring by Seconds



Step 9 In the navigation pane, click **Advanced O&M** > **Real-Time Monitoring** to view metric data.

- View the current data collection period in the upper part of the page.
- Monitoring by Seconds supports the following metrics: CPU usage, memory usage, SELECT statements per second, DELETE statements per second, and INSERT statements per second. You can click **View details** to view more metrics. For details about the metrics, see **TaurusDB Metrics**.
- If you need to change the collection period, see **Modifying Collection Interval**.

----End

Disabling Monitoring by Seconds

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

Step 2 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 3 Click **Performance**.

Step 4 In the upper part of the page, click **Enable Monitoring by Seconds**.

Step 5 In the displayed dialog box, click  next to **Monitoring by Seconds** and click **OK**.

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

----End

Modifying Collection Interval

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

Step 2 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

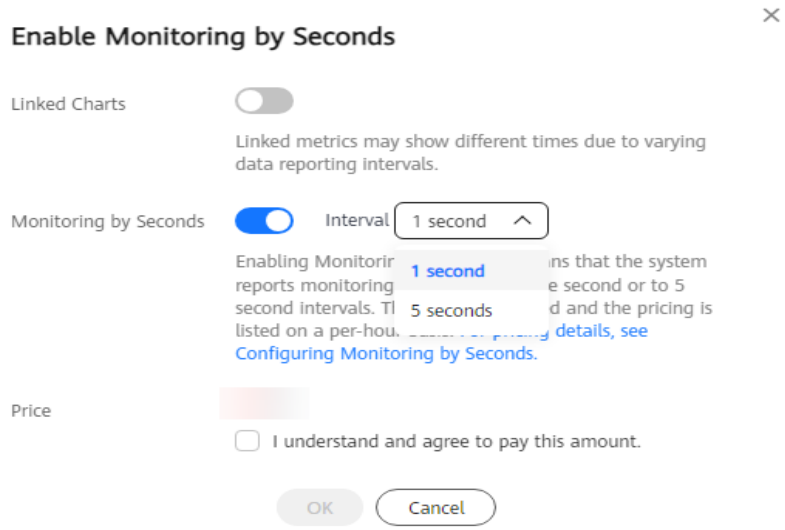
Step 3 Click **Performance**.

Step 4 In the upper part of the page, click **Enable Monitoring by Seconds**.

Step 5 Select the monitoring interval and click **OK**.

Monitoring data will be reported based on the new collection interval about 5 minutes later.

Figure 23-3 Modifying the collection interval



----End

APIs

- **Configuring Monitoring by Seconds**
- **Querying the Configuration of Monitoring by Seconds**

23.4 Configuring TaurusDB Alarm Rules

Scenarios

You can create alarm rules for an instance to configure the monitored objects and notification policies and then stay aware of the instance status.

The following parameters can be configured: alarm rule names, services, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Constraints

A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm will be triggered.

Creating a Metric Alarm Rule for 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 **Service List**. Under **Management & Governance**, click **Cloud Eye**.

Alternatively, go to the Cloud Eye console using any of the following methods:

- On the **Instances** page, locate a DB instance and click **View Metrics** in the **Operation** column.
- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click and choose **View Metric**.
- In the **Node List** area of the **Basic Information** page, locate a node and click **View Metrics** in the **Operation** column.

Step 4 In the navigation pane, choose **Alarm Management** > **Alarm Rules**. On the displayed page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the displayed page, set parameters as prompted.

Figure 23-4 Setting alarm rule parameters

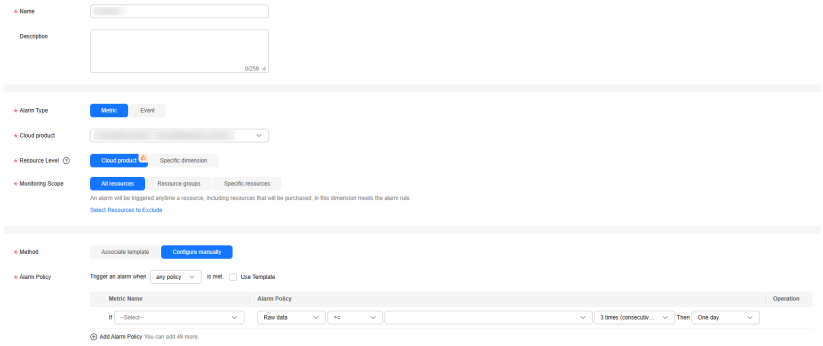


Table 23-5 Alarm rule parameters

Parameter	Description
Name	Name of the alarm rule. The system generates a random name, but you can change it if needed.
Description	Description of the alarm rule.
Alarm Type	Select **Metric**.

Parameter	Description
Cloud product	Select **TaurusDB**.
Resource Level	**Cloud product** is recommended.
Monitoring Scope	• **All resources**: An alarm will be triggered if any resource of the current cloud product meets the alarm policy. To exclude resources that do not require monitoring, click **Select Resources to Exclude**. • **Resource groups**: An alarm will be triggered if any resource in the selected resource group meets the alarm policy. • **Specific resources**: Click **Select Specific Resources** to select resources.
Method	• **Associate template**: After an associated template is modified, the policies contained in this alarm rule to be created will be modified accordingly. You are advised to select **Use existing template**. The existing templates already contain three common alarm metrics: CPU usage, memory usage, and storage space usage. • **Configure manually**: Configure alarm policies manually.
Template	If you select **Associate template** for **Method**, you need to select a template. You can select a default alarm template or create a custom template.
Alarm Policy	If you select **Configure manually** for **Method**, you need to configure alarm policies. An alarm is triggered when the metric configured for this alarm reaches the preset threshold in consecutive periods. For example, Cloud Eye triggers an alarm every 5 minutes if the average CPU usage of the monitored object is 80% or more for three consecutive 5-minute periods.
Alarm Severity	The alarm severity can be **Critical**, **Major**, **Minor**, or **Informational**.

Figure 23-5 Setting alarm notification parameters

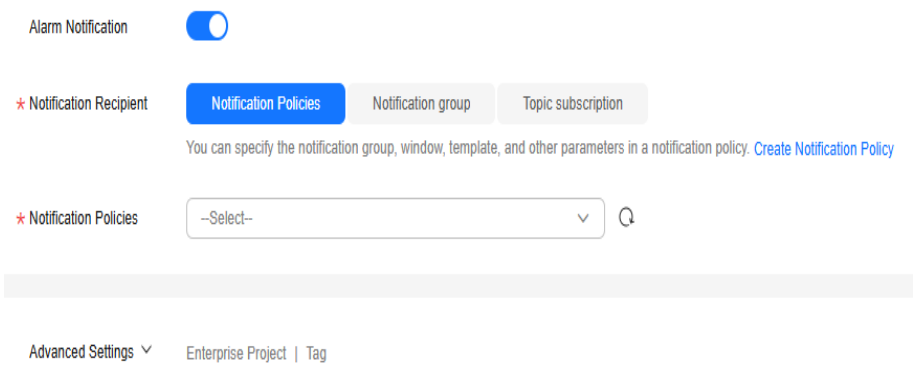


Table 23-6 Alarm notification parameters

Parameter	Description
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Recipient	You can select a notification group or topic subscription as required.
Notification Group	Notification group the alarm notification is to be sent to.
Notification Object	Object the alarm notification is to be sent to. You can select the account contact or a topic. This parameter is only available if you select **Topic subscription** for **Notification Recipient**. • The account contact is the mobile phone number and email address of the registered account. • A topic is used to publish messages and subscribe to notifications.
Notification Window	Time window during which Cloud Eye sends notifications. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within this window.
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.
Tag	Key-value pairs that you can use to easily categorize and search for cloud resources.

Step 6 Click **Create**.

For details about how to create alarm rules, see **Creating an Alarm Rule** in *Cloud Eye User Guide*.

----End

Creating a Metric Alarm Rule for a Proxy Instance

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

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

Alternatively, go to the TaurusDB console. On the **Instances** page, click the instance name. In the navigation pane, choose **Database Proxy**. On the displayed page, locate a proxy instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

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

Step 4 Click **Database Proxy Service DBPROXY**.

Step 5 On the **Instances** page, locate a proxy instance and choose **More** > **Create Alarm Rule** in the **Operation** column.

Step 6 On the displayed page, set parameters as needed. For details, see **Creating an Alarm Rule**.

1. Set the alarm rule name and description.

Figure 23-6 Setting the alarm rule name and description

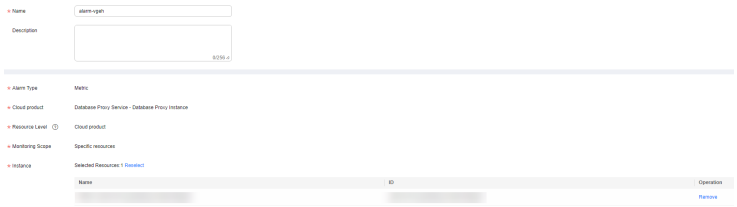


Table 23-7 Name and **Description**

Parameter	Description
Name	Name of the alarm rule. The system generates a random name, but you can change it if needed. Example value: **alarm-b6al**
Description	(Optional) Supplementary information about the alarm rule.

2. Set alarm rule parameters.

Figure 23-7 Setting alarm rule parameters

Table 23-8 Alarm rule parameters

Parameter	Description
Method	Select an associated template, use an existing template or create a custom template as required. – Modifying the template will also modify its associated alarm rules. – If you select **Configure manually**, you can configure **Alarm Policy** and **Alarm Severity** as required.
Template	Select the template to be used. You can select a default alarm template or create a custom template. For details about how to create a custom template, see **Creating a Custom Template**
Alarm Policy	Specifies the policy for triggering an alarm. A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm will be triggered.

3. Set alarm notification parameters.

Figure 23-8 Setting alarm notification parameters

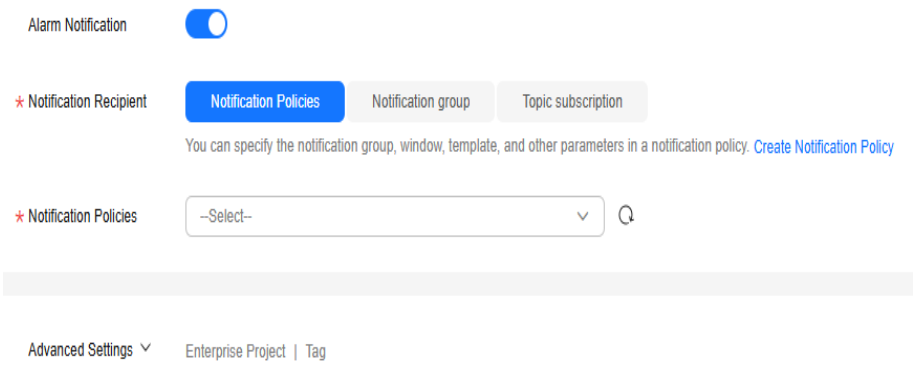


Table 23-9 Alarm notification parameters

Parameter	Description
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Recipient	You can select a notification group or topic subscription as required.

Parameter	Description
Notification Group	Notification group the alarm notification is to be sent to.
Notification Object	Object the alarm notification is to be sent to. You can select the account contact or a topic. This parameter is only available if you select **Topic subscription** for **Notification Recipient**. – The account contact is the mobile phone number and email address of the registered account. – A topic is used to publish messages and subscribe to notifications.
Notification Window	Time window during which Cloud Eye sends notifications. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within this window.
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.
Tag	Key-value pairs that you can use to easily categorize and search for cloud resources.

Step 7 Click **Create**.

----End

23.5 Event Monitoring

23.5.1 Introducing Event Monitoring

Event monitoring provides reporting, query, and alarm functions for event data. You can create alarm rules for both system events and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on TaurusDB that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, such as deleting a read replica or changing instance specifications.

Event monitoring provides an API for reporting custom events (abnormal events or important change events) generated by services to Cloud Eye.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see **Events Supported by Event Monitoring**.

23.5.2 Viewing Event Monitoring Data

Scenarios

In event monitoring, you can query system events that are automatically reported to Cloud Eye and custom events reported to Cloud Eye through the API. You can create alarm rules for both system events and custom events. When specific events occur, Cloud Eye generates alarms for you.

Event monitoring is enabled by default.

You can view the event monitoring data.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **TaurusDB**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using either of the following methods:

- On the **Instances** page, click the instance name to go to the **Basic Information** page. In the upper right corner of the page, click and choose **View Metrics**.
- In the **Node List** area of the **Basic Information** page, locate a node and click **View Metrics** in the **Operation** column.

Step 5 Click to return to the Cloud Eye console.

Step 6 In the navigation pane, choose **Event Monitoring**.

On the displayed **Event Monitoring** page, all system events of the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view events generated in different periods.

Step 7 Locate an event and click **View Event** in the **Operation** column to view details about a specific event.

----End

23.5.3 Creating Alarm Rules for Event Monitoring

Scenarios

If you need to focus on core events, you can create alarm rules and alarm notifications for specified events. This way, you get timely alerts and can quickly troubleshoot or switch services. This section describes how to create an alarm rule to monitor an event.

Procedure

Step 1 Log in to the management console.

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

Step 3 In the navigation pane, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the displayed page, set parameters as needed.

Table 23-10 Parameter description

Parameter	Description
Name	Specifies the name of the alarm rule. The system generates a random name, but you can change it if needed.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create an enterprise project.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Event Type	Specifies the event type of the metric corresponding to the alarm rule.
Event Source	Specifies the service the event is generated for. Example value: **TaurusDB**
Monitoring Scope	Specifies the monitoring scope for event monitoring.
Method	Specifies the event creation method.
Alarm Policy	Events indicate the instantaneous operations users performed on system resources, such as login and logout. For details about events supported by Event Monitoring, see **Events Supported by Event Monitoring**. You can select a trigger mode and alarm severity as needed.

Click  to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 23-11 Alarm notification parameters

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text message, or through HTTP/HTTPS request to servers.
Notification Object	Specifies the object an alarm notification is to be sent to. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Validity Period	Cloud Eye sends notifications only within the validity period specified in the alarm rule. If you set **Validity Period** to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification.

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

----End

23.5.4 Events Supported by Event Monitoring

TaurusDB Instance Events

Table 23-12 TaurusDB

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
TaurusDB	Incremental backup failure	TaurusIncrementalBackupInstanceFailed	Major	The network between the instance and the management plane or the OBS is disconnected, or the backup environment created for the instance is abnormal.	**Submit a service ticket**.	Backup jobs fail.
	Read replica creation failure	addReadonlyNodesFailed	Major	The quota is insufficient or underlying resources are exhausted.	Check the read replica quota. Release resources and create read replicas again.	Read replicas fail to be created.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	DB instance creation failure	createInstanceFailed	Major	The quota is insufficient or underlying resources are exhausted.	Check the instance quota. Release resources and create instances again.	Instances fail to be created.
	Read replica promotion failure	activeStandBySwitchFailed	Major	The read replica fails to be promoted to the primary node due to network or server failures. The original primary node takes over services quickly.	**Submit a service ticket**.	The read replica fails to be promoted to the primary node.
	Instance specifications change failure	flavorAlterationFailed	Major	The quota is insufficient or underlying resources are exhausted.	**Submit a service ticket**.	Instance specifications fail to be changed.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Faulty DB instance	TaurusInstanceRunningStatusAbnormal	Major	The instance process is faulty or the network between the instance and the DFV storage is disconnected.	**Submit a service ticket**.	Services may be affected.
	DB instance recovered	TaurusInstanceRunningStatusRecovered	Major	The instance is recovered.	Observe the service running status.	None.
	Faulty node	TaurusNodeRunningStatusAbnormal	Major	The node process is faulty or the network between the node and the DFV storage is disconnected.	Observe the instance and service running statuses.	A read replica may be promoted to the primary node.
	Node recovered	TaurusNodeRunningStatusRecovered	Major	The node is recovered.	View the node running status.	None.
	Read replica deletion failure	TaurusDeleteReadOnlyNodeFailed	Major	The network between the management plane and the read replica is disconnected or the VM fails to be deleted from IaaS.	**Submit a service ticket**.	Read replicas fail to be deleted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Password reset failure	TaurusResetInstancePasswordFailed	Major	The network between the management plane and the instance is disconnected or the instance is abnormal.	Check the instance status and try again. If the fault persists, **submit a service ticket**.	Passwords fail to be reset for instances.
	DB instance reboot failure	TaurusRestartInstanceFailed	Major	The network between the management plane and the instance is disconnected or the instance is abnormal.	Check the instance status and try again. If the fault persists, **submit a service ticket**.	Instances fail to be rebooted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Restoration to new DB instance failure	TaurusRestoreToNewInstanceFailed	Major	The instance quota is insufficient, underlying resources are exhausted, or the data restoration logic is incorrect.	If the new instance fails to be created, check the instance quota, release resources, and try to restore to a new instance again. In other cases, **submit a service ticket**.	Backup data fails to be restored to new instances.
	EIP binding failure	TaurusBindEIPToInstanceFailed	Major	The binding task fails.	**Submit a service ticket**.	EIPs fail to be bound to instances.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	EIP unbinding failure	TaurusUnbindEIPFromInstanceFailed	Major	The unbinding task fails.	**Submit a service ticket**.	EIPs fail to be unbound from instances.
	Parameter modification failure	TaurusUpdateInstanceParameterFailed	Major	The network between the management plane and the instance is disconnected or the instance is abnormal.	Check the instance status and try again. If the fault persists, **submit a service ticket**.	Instance parameters fail to be modified.
	Parameter template application failure	TaurusApplyParameterGroupToInstanceFailed	Major	The network between the management plane and instances is disconnected or the instances are abnormal.	Check the instance status and try again. If the fault persists, **submit a service ticket**.	Parameter templates fail to be applied to instances.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Full backup failure	TaurusBackupInstanceFailed	Major	The network between the instance and the management plane or the OBS is disconnected, or the backup environment created for the instance is abnormal.	**Submit a service ticket**.	Backup jobs fail.
	Read replica promotion	TaurusActiveStandbySwitched	Major	When the primary node is faulty, a read replica is promoted to the primary node.	Check the instance status. If the fault persists, **submit a service ticket**.	Services are intermittently interrupted.
	Instance read-only	NodeReadonlyMode	Major	The instance supports only query operations.	**Submit a service ticket**.	After the instance becomes read-only, write requests cannot be processed.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Instance read/ write	NodeReadWriteMode	Major	The instance can process both write and read requests.	**Submit a service ticket**.	None.
	Instance DR switchover	DisasterSwitchOver	Major	If an instance is faulty and unavailable, a switchover is performed to ensure that the instance continues to provide services.	**Submit a service ticket**.	The database connection is intermittently interrupted. The DR instance is promoted to primary to provide services.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Database process restarted	TaurusDatabaseProcessRestarted	Major	The database process is stopped due to insufficient memory or high load.	Log in to the Cloud Eye console. Check whether the memory usage increases sharply or the CPU usage is too high for a long time. You can increase the specifications or optimize the service logic.	When the database process is suspended, workloads on the node are interrupted. In this case, the HA service automatically restarts the database process and attempts to recover the workloads.

Proxy Instance Events

Table 23-13 Events supported by proxy instances

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
Proxy instance	Connection failure between proxy instance and DB instance	proxy_connection_failure_to_db	Major	The proxy instance failed to establish a new connection with the primary node of a DB instance, and it may fail to establish a new connection with a read replica. The DB instance or proxy instance is overloaded, or the network between the them is abnormal.	Change values of related parameters based on metrics (Connections, Active Connections, and CPU Usage) of the DB instance and proxy instance. If the metrics are normal, **submit a service ticket**.	Service requests routed through the proxy instance are interrupted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Connection failure between database proxy and read replica	proxy_connection_failure_to_replica	General	The proxy instance failed to establish a new connection with a read replica. The read replica is overloaded, or the network between the proxy instance and read replica is abnormal.	Change values of related parameters based on metrics (Connections, Active Connections, and CPU Usage) of the read replica. If the metrics are normal, **submit a service ticket**.	Read requests accessed through the proxy instance are interrupted.

Event Source	Event Name	Event ID	Alarm Severity	Description	Handling Suggestion	Impact
	Proxy instance access to DB instance failure	proxy_connection_failure_cause_security_group	Major	No rules in the security group allow the proxy instance to access the DB instance.	Add the proxy instance address to the rules of the security group.	Service requests routed through the proxy instance are interrupted.

24 Interconnection with CTS

24.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to TaurusDB for further query, audit, and backtracking. **Table 24-1** lists the supported operations.

Table 24-1 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
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
Creating a database	instance	createDatabase

Operation	Resource Type	Trace Name
Creating a database account	instance	createDatabaseUser
Resetting a password	instance	resetPassword
Deleting a database	instance	dropDatabase
Deleting a database account	instance	dropDatabaseUser
Changing the password of a database user	instance	modifyDatabaseUserPwd
Restoring data to a new DB instance	instance	restoreInstance
Enabling read/write splitting	instance	openProxy
Disabling read/write splitting	instance	closeProxy
Assigning read weights	instance	setProxyWeight
Changing the CPU and memory specifications of an instance	instance	resizeFlavorOrVolume
Configuring monitoring by seconds	instance	openSecondExtend
Upgrading a minor version	instance	upgradeVersion
Adding a tag	instance	addInstanceTags
Authorizing database user permissions	instance	grantDatabaseUser
Revoking database user permissions	instance	revokeDatabaseUser
Applying for a private domain name	instance	createDnsName
Modifying a private domain name	instance	modifyDnsName
Changing the routing policy of a proxy instance	instance	modifyProxyRouteMode
Changing the port of a proxy instance	instance	modifyProxyPort

Operation	Resource Type	Trace Name
Applying for a private domain name for a database proxy instance	instance	proxyCreateDns
Changing a private domain name for a database proxy instance	instance	modifyProxyDnsName
Deleting a private domain name for a database proxy instance	instance	deleteProxyDnsName
Deleting database proxy nodes	instance	reduceProxy
Creating a backup	backup	createManualSnapshot
Configuring an automated backup policy	backup	setBackupPolicy
Deleting a backup	backup	deleteManualSnapshot
Creating a parameter template	parameterGroup	createParameterGroup
Modifying parameters in a parameter template	parameterGroup	updateParameterGroup
Deleting a parameter template	parameterGroup	deleteParameterGroup
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Comparing parameter templates	parameterGroup	compareParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup

24.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

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

Prerequisites

Before using CTS, you need to enable it. For details, see **Enabling CTS**.

Procedure

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

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

Step 3 In the upper left corner of the page, click  and choose **Management & Governance** > **Cloud Trace Service**.

Step 4 In the navigation pane, choose **Trace List**.

Step 5 Filter conditions to query traces.

Table 24-2 Filtering criteria

Filtering Criteria	Description
Time Range	In the upper right corner, choose **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.
Trace Type	Select **Management** or **Data** • Management traces record details about creating, configuring, and deleting cloud service resources in your tenant account. • Data traces record operations on data, such as data upload and download. **NOTE** – If you select **Data** for **Trace Type**, you can only filter traces by tracker. – The trace list does not record queries.
Trace Source	Select a trace source as needed.
Resource Type	Select a resource type as needed.
Search By	If you select **Resource ID** for **Search By**, you need to enter a resource ID.
Operator	Select a specific operator from the drop-down list.
Trace Status	Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.

Step 6 View the events that meet the search criteria.

Step 7 Click an event name. Details about the event are displayed in the dialog box on the right.

Step 8 Click **Export** in the upper left corner of the list. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to the traces.

For details about key fields in the trace structure, see **Trace Structure** and **Example Traces** in *Cloud Trace Service User Guide*.

----End

25 Task Center

25.1 Viewing a Task

You can view the progresses and results of instant and scheduled tasks on the **Task Center** page.

Viewing an Instant Task

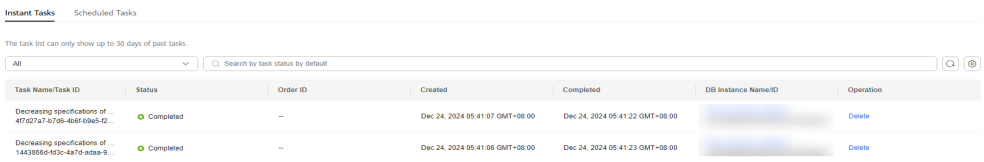
Step 1 Log in to the management console.

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

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

Figure 25-1 Viewing an instant task



- Identify a task based on the task name/task ID, order ID, and instance name/ID, and view the task creation time and end time.
- Click the **All** drop-down list box in the upper part to view the task execution progress and status in a specified period. The default period is all time. The task list shows tasks that have been executed in the past 30 days.
- Click the filter box in the upper part to query the desired instant tasks by task name and task status.
  - Task status: **Running**, **Completed**, and **Failed**
  - Task name:

- Creating a TaurusDB instance
- Creating a TaurusDB read replica
- Rebooting a TaurusDB instance
- Changing a TaurusDB instance port
- Promoting a TaurusDB read replica to the primary node
- Binding an EIP to a TaurusDB instance
- Unbinding an EIP from a TaurusDB instance
- Changing the instance name for a TaurusDB instance
- Changing a security group for a TaurusDB instance
- Deleting a TaurusDB instance
- Upgrading a DB instance version
- Deleting a TaurusDB read replica
- Changing the specifications of a TaurusDB instance
- Restoring to a new TaurusDB instance
- Changing private IP address
- Modifying collection period of Monitoring by Seconds
- Adding database proxy nodes
- Deleting database proxy nodes
- Enabling database proxy
- Disabling database proxy
- Changing IP address of a proxy instance
- Changing proxy instance specifications
- Enabling or disabling SSL
- Changing consistency level of a proxy instance
- Changing read weights of nodes
- Restoring to an existing DB instance
- Restoring tables to a point in time
- Creating a database
- Deleting a database

- Creating a database account
- Deleting a database account
- Changing the password of a database user
- Changing the host IP address of a database user
- Authorizing database user permissions
- Deleting database user permissions
- Rebooting a node
- Changing read/write splitting address
- Changing a node name
- Increasing specifications of a serverless instance
- Decreasing specifications of a serverless instance
- Changing the port of a proxy instance
- Applying for a private domain name for a proxy instance
- Changing the private domain name of a proxy instance
- Deleting the private domain name of a proxy instance
- Changing the routing policy of a proxy instance
- Enabling or disabling SSL for a proxy instance
- Applying for a private domain name for the DB instance
- Changing the private domain name of the DB instance
- Creating the primary instance for a RegionlessDB cluster
- Creating standby instances for a RegionlessDB cluster
- Deleting a RegionlessDB cluster
- Setting write forwarding for a RegionlessDB cluster
- Modifying the remarks of a TaurusDB database
- Modifying the remarks of a TaurusDB database user

----End

Viewing a Scheduled Task

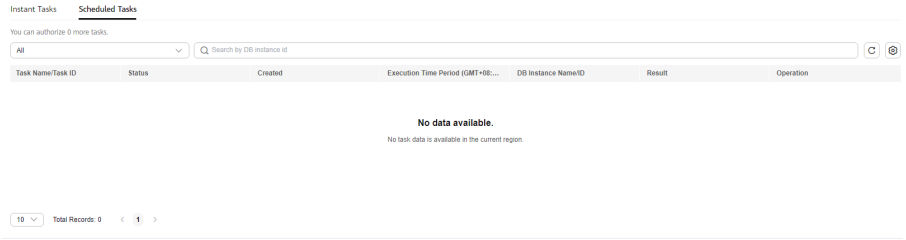
Step 1 **Log in to the management console.**

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 **Databases** > **TaurusDB**.

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

Figure 25-2 Viewing a scheduled task



- To identify the task, you can use the instance name/ID or enter the instance ID in the search box in the upper right corner.
- You can enter the instance ID or task status in the search box to determine the desired task and view the task creation time and execution time.

  Task status: **Running**, **Completed**, **Failed**, **Canceled**, **To be executed**, and **To be authorized**.
- Click the **All** drop-down list box in the upper part to view the task execution progress and status in a specified period. The default period is all time.

----End

APIs

- **Obtaining Information About a Task with a Specified ID**
- **Obtaining Instant Tasks**
- **Obtaining Scheduled Tasks**

25.2 Deleting a Task Record

You can delete the task records that no longer need to be displayed.

Constraints

- Deleted task records cannot be recovered. Exercise caution when performing this operation.
- Deleting task records will not delete instances or terminate tasks in progress.

Deleting an Instant Task Record

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

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

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

Step 4 In the navigation pane, choose **Task Center**. Locate the target task on the displayed **Instant Tasks** tab page.

Step 5 Click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

You can delete instant task records with the following statuses:

- Completed
- Failed

----End

Deleting a Scheduled Task Record

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

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

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

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the task record to be deleted and check whether the task record status is **To be executed** or **To be authorized**.

- If yes, go to **Step 5**.
- If no, go to **Step 6**.

Step 5 Click **Cancel** in the **Operation** column. In the displayed dialog box, click **OK**. Then, click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, enter **DELETE** as prompted and click **OK**.

You can delete scheduled task records with the following statuses:

- Completed
- Failed
- Canceled
- To be authorized

----End

APIs

- **Canceling a Scheduled Task**
- **Deleting a Task Record**

26 Tag Management

Scenarios

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

- You are advised to configure predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Each instance can have up to 20 tags.

Adding a Tag

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

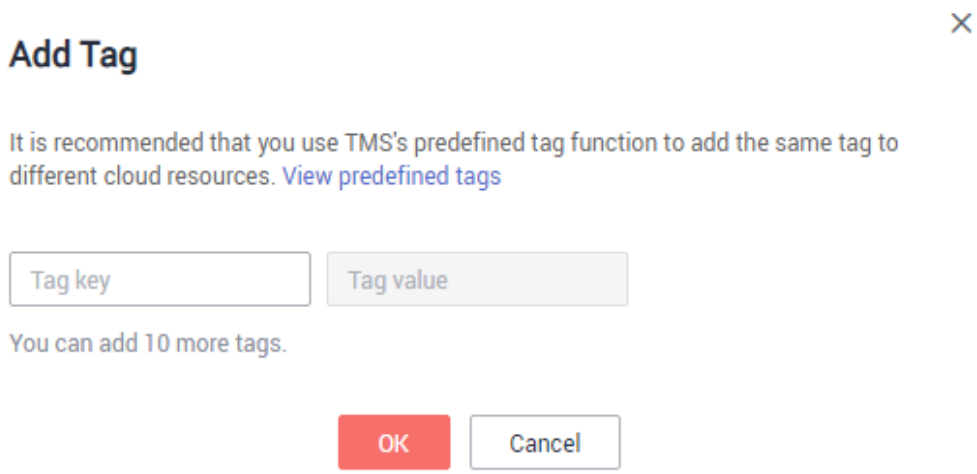
Step 2 Click  in the upper left corner and select a region and project.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **TaurusDB**.

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

Step 5 Choose **Tags** in the navigation pane and click **Add Tag**. In the displayed dialog box, enter a tag key and value, and click **OK**.

Figure 26-1 Adding a tag



- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with all instances except the current one.
- The tag key must be unique and must consist of 1 to 36 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- The tag value can be empty or consist of 1 to 43 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

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

----End

Editing a Tag

Step 1 **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 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 and choose **Databases** > **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

APIs

- Querying Resource Tags
- Querying Project Tags
- Adding or Deleting Tags in Batches

27 Quota Management

Scenarios

Quotas put limits on the quantities and capacities of resources available to users, for example, the maximum number of TaurusDB instances that you can create.

If a quota cannot meet your needs, apply for a higher quota.

Viewing Quotas

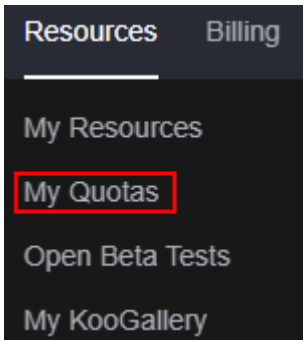
Step 1 **Log in to the management console.**

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

Step 3 Choose **Resources** > **My Quotas** in the upper right corner of the page.

The **Quota** page is displayed.

Figure 27-1 Viewing quotas



Step 4 View the used and total quotas of each type of resources.

----End

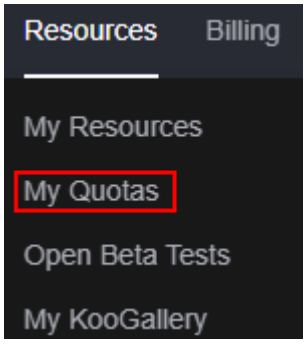
Increasing Quotas

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

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

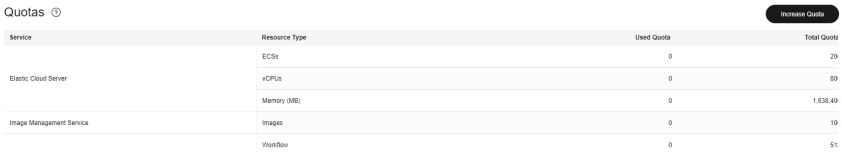
Step 3 In the upper right corner of the console page, choose **Resources** > **My Quotas**.

Figure 27-2 Viewing quotas



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

Figure 27-3 Increasing quotas



Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for quota adjustment.

Step 6 After all necessary parameters are configured, select the agreement and click **Submit**.

----End

APIs

- Querying the Instance Quotas of a Tenant
- Querying the Resource Quotas of a Specified Enterprise Project
- Configuring Resource Quotas for a Specified Enterprise Project
- Modifying the Resource Quotas of a Specified Enterprise Project

