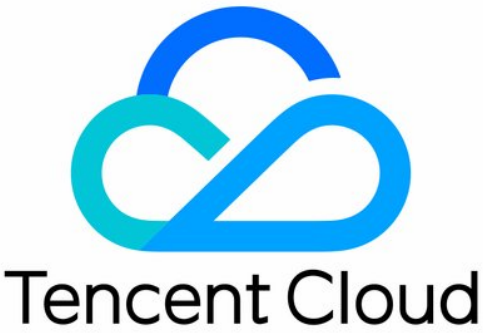
TDSQL-C for MySQL

Operation Guide

Product Documentation



Contents

Operation Guide
Switching Cluster Page View in Console

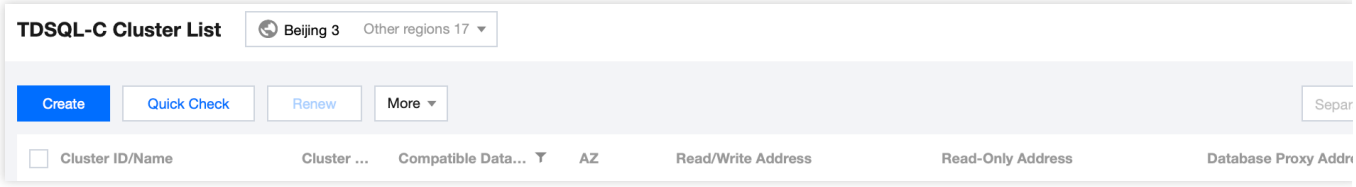
Last updated：2023-11-08 15:49:21

Overview

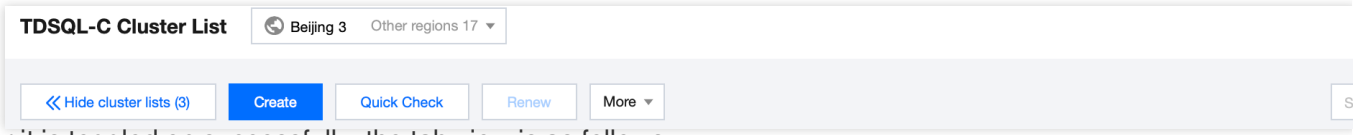
In the TDSQL-C for MySQL console, the cluster list page supports tab and list views. You can switch between them as instructed below.
When there are no more than ten clusters, the tab view displays clearer cluster information and instance data than the list view.
Note
TDSQL-C for MySQL walks you through using the tab view in the console. In addition to the document introduction, you may learn about the tab view by clicking **User Guide** in the the console.

Directions

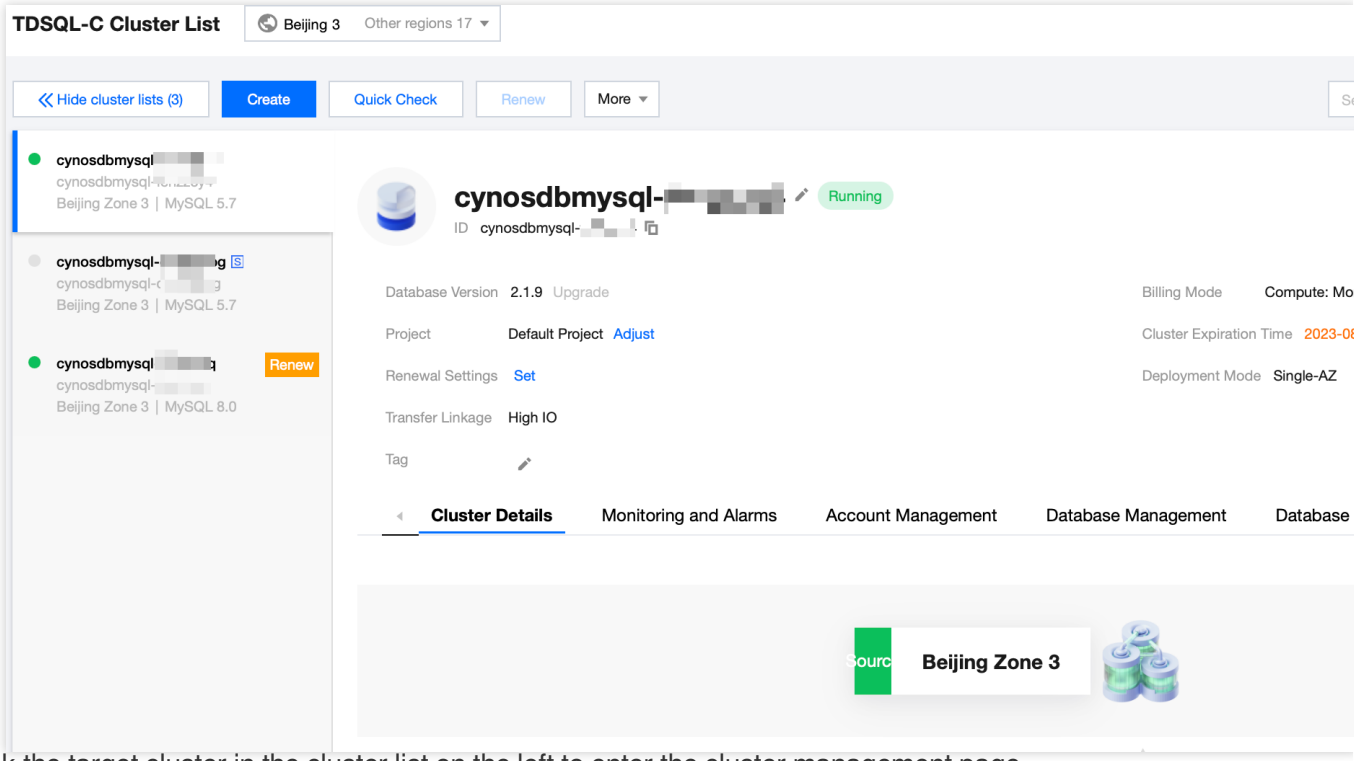
1. Log in to the TDSQL-C for MySQL console, and toggle on the button in front of **List View** in the top right of the cluster list.
Before toggling on:



After toggling on:



2. After it is toggled on successfully, the tab view is as follows.



3. Click the target cluster in the cluster list on the left to enter the cluster management page.

Note

Cluster List provides indicator lights to indicate the status of the current cluster as follows.

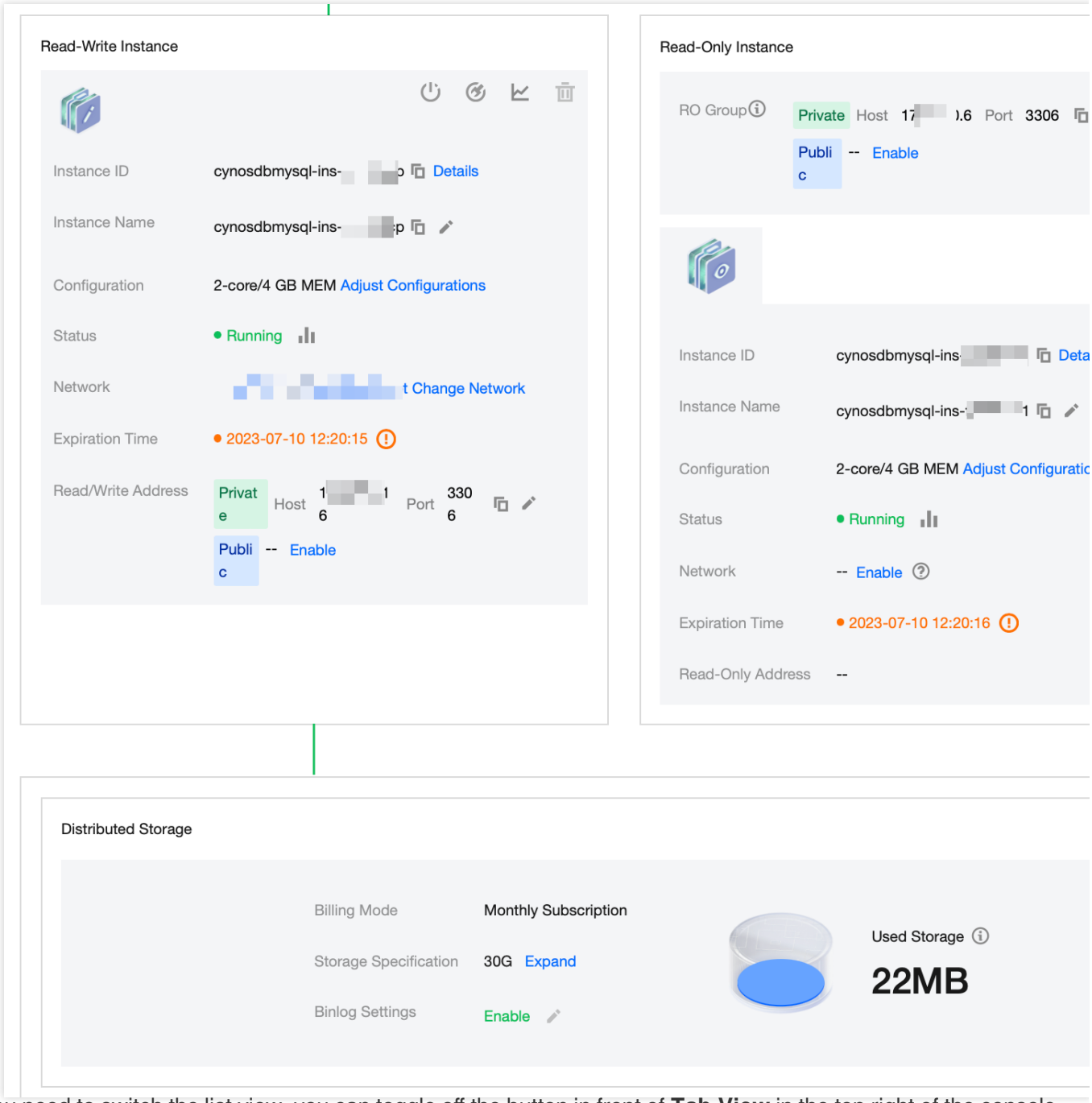
A green light indicates that the cluster status is "Running".

A gray light indicates that the cluster status is "Creating".

A red light indicates that the cluster status is "Isolating".

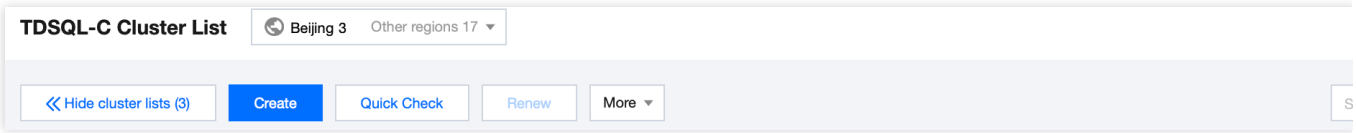
4. On the **Cluster Management** page, you can upgrade the database version of the cluster, log in to the cluster, terminate the cluster, request a refund, adjust project, change network, modify deployment mode, edit tags, set renewal and billing mode.

5. On the **Cluster Details** page, you can learn about the cluster architecture and storage details of TDSQL-C for MySQL and perform related operations on the instance. For more information, see Instance Management Overview.

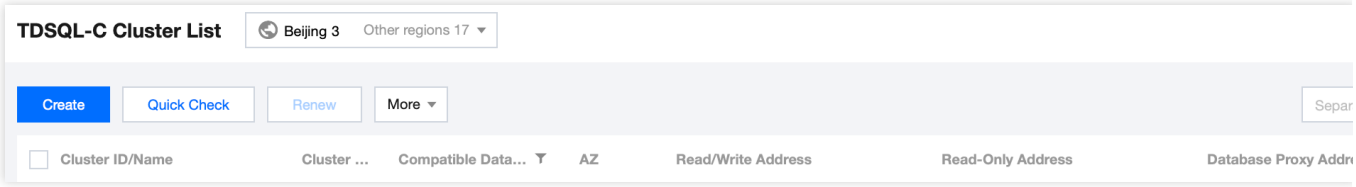


6. If you need to switch the list view, you can toggle off the button in front of **Tab View** in the top right of the console.

Before toggling off:

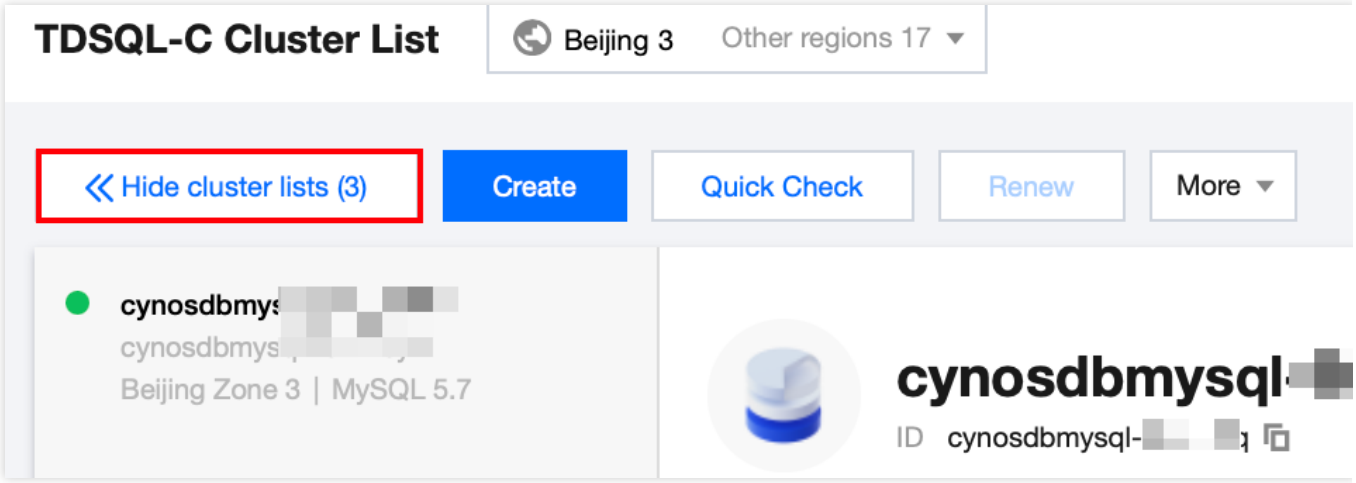


After toggling off:

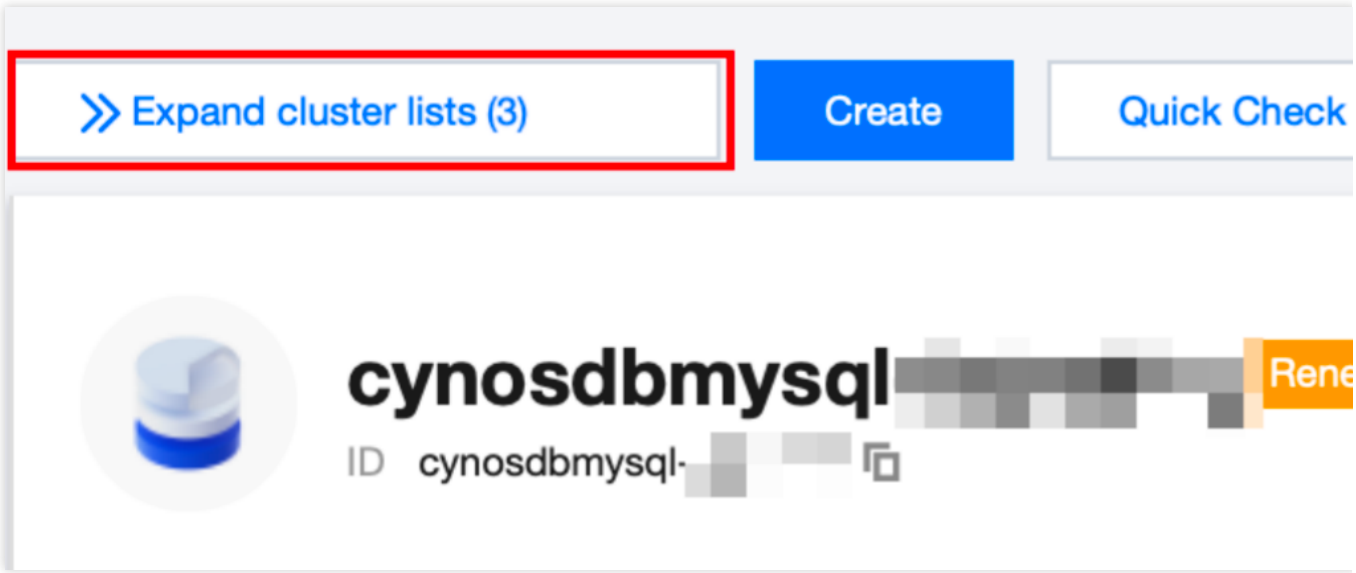


Hiding Cluster List

In the tab view, you can hide **Cluster List** for a clearer preview of the corresponding cluster management page. On the top of the cluster list, click **Hide cluster list**.



To display the cluster list again, click **Expand cluster list**.



Database Connection
Configuring Server
Quickly Configuring Linux CVM Instance

Last updated：2023-11-09 15:24:37

Cloud Virtual Machine (CVM) is a scalable cloud computing service. It frees you from estimation of resource usage and upfront investment. You can start CVM instances and deploy applications immediately.
This document describes how to create a Linux CVM instance from scratch in the simplest way. You can purchase and configure your first CVM instance by following the instructions below. For more information about building a Windows CVM instance, see Quickly Configuring Windows CVM Instance.

Prerequisite

You have registered a Tencent Cloud account. For more information, see Account Registration.

Configuring a Linux CVM Instance

Note
To ensure successful connection to TDSQL-C for MySQL, the purchased CVM and TDSQL-C for MySQL instances must meet the following requirements:
Both are under the same Tencent Cloud root account.
Both are in the same region.
Both are in the same VPC.

Step 1. Purchase a Linux CVM instance

1. Go to the quick purchase page.
2. On the purchase page, select the **Quick Configuration** tab, configure the CVM instance, and click **Buy Now**.
The configuration is as described below:

Configuration Items	Description
Region	Select the same region as your TDSQL-C for MySQL cluster.
Instance	Select the CVM instance specifications as needed. "General (2-core, 4GB)" is selected in this example.

Operating system	Select the CVM operating system as needed. "CentOS 7.9 64-bit" is selected in this example.
Public IP	After this option is selected, a public IP will be assigned to your instance. The default public network bandwidth is "1 Mbps", which can be adjusted as needed.
Login methods	After creating a CVM instance, you can obtain the random password in the Message Center.
Default configuration	Six default configuration items can be expanded, such as AZ and security group.
Auto-renewal	After the checkbox is selected, if your account balance is sufficient, the CVM instance will be automatically renewed by month upon expiration.
Agreements	Read and indicate your consent to the relevant agreements.
Period	Select the purchase period, which is "1 month" by default.
Quantity	Select the quantity, which is "1" by default.

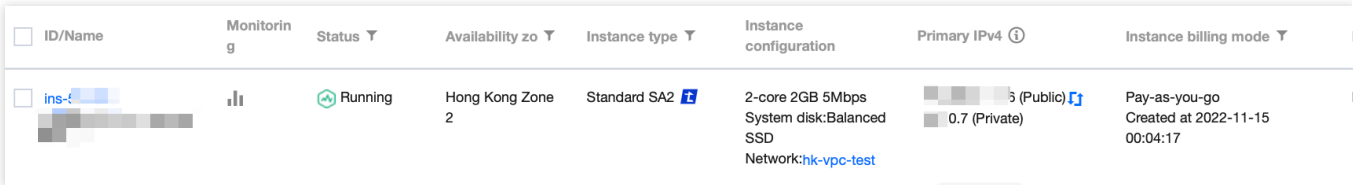
Click **Buy now** and make the payment to purchase the CVM instance. You can use the CVM instance as a personal virtual machine or as a server for your websites. Then, you can log in to the CVM instance

Step 2. Log in to the CVM instance

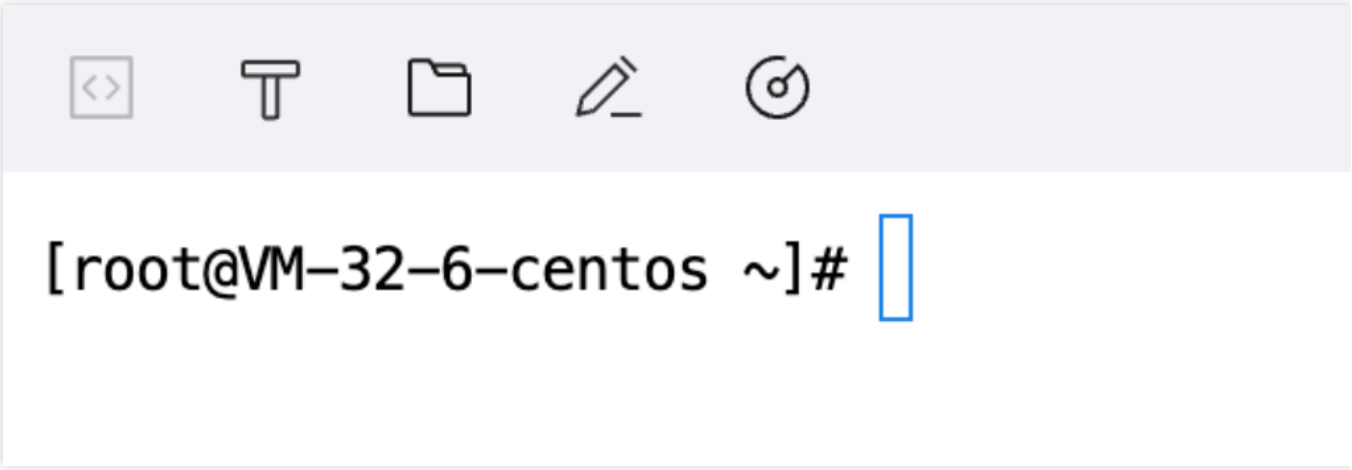
Note

After you purchase a quickly configured CVM instance, the system will automatically generate a login password and send it to your Message Center. This password is the credential for logging in to the CVM instance. You can view the password in the Message Center.

1. Log in to the CVM console and select a region at the top.
2. Find the CVM instance you purchased in the instance list and click **Log In** in the **Operation** column on the right.



3. In the **Standard Login** | **Linux Instance** window, enter the username (which is `root` by default) and password of the **CVM instance** and click **Log In**.
4. A page will be displayed after successful login.



Quickly Configuring Windows CVM Instance

Last updated：2023-02-07 12:04:33

CVM is a scalable cloud computing service that frees you from estimation of resource usage and upfront investment. You can start CVM instances and deploy applications immediately.

This document describes how to purchase and configure a Windows CVM instance from scratch in the simplest way. For more information on how to create a Linux CVM instance, see Quickly Configuring Linux CVM Instance.

Prerequisites

You have registered a Tencent Cloud account as instructed in Account Registration.

Configuring a Windows CVM Instance

Note：
To ensure successful connection to TDSQL-C for MySQL, the purchased CVM and TDSQL-C for MySQL instances must be:

- Under the same Tencent Cloud root account.
- In the same region.
- In the same VPC.

Step 1. Purchase a Windows CVM instance

1. Go to the quick purchase page.
2. On the purchase page, select the **Quick Configuration** tab, configure the CVM instance, and click **Buy Now**.
   The configuration is as described below:

Configuration Item	Description
Region	Select the same region as your TDSQL-C for MySQL cluster.
Instance	Select the CVM model configuration as needed, which is "General (2-core, 4 GB)" here for example.

Configuration Item	Description
Operating System	Select the CVM operating system as needed, which is "Windows Server 2012 R2 Datacenter Edition 64-bit" here for example.
Public IP	After this option is selected, a public IP will be assigned to your instance. The default public network bandwidth is "1 Mbps", which can be adjusted as needed.
Login Methods	After creating a CVM instance, you can obtain the random password in the Message Center.
Default configuration	Six default configuration items can be expanded, such as AZ and security group.
Auto-renewal	After this option is selected, if your account balance is sufficient, the CVM instance will be automatically renewed by month upon expiration.
Agreements	Read and indicate your consent to the relevant agreements.
Period	Select the purchase period, which is "1 month" by default.
Quantity	Select the quantity, which is "1" by default.

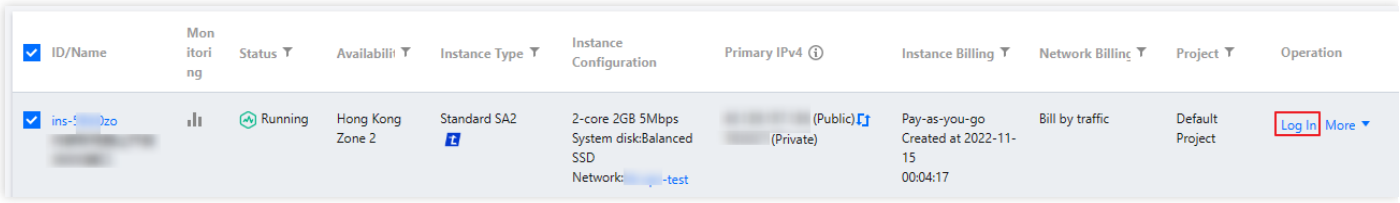
After you make the payment, the CVM instance will be successfully purchased. It can be used as a personal virtual machine or website server. Next, you can log in to it.

Step 2. Log in to the CVM instance

Note：
After you purchase a quickly configured CVM instance, the system will automatically generate a login password and sent it to you in the Message Center. This password is the credential for logging in to the CVM instance.

1. Log in to the CVM console.
2. On the instance management page, proceed according to the actually used view mode:
   - List view
   - Tab view

Locate the Windows CVM instance you want to log in to and click **Log In** on the right as shown below:



3. In the **Standard Login** | **Windows Instance** window that is opened, enter the login information according to the actual situation.

- **Port**: The default port is 3389. Enter a value as needed.
- **Username**: The default username of Windows instances is `Administrator` . Enter a value as needed.
- **Password**: Enter the login password obtained in the Message Center.

5. Click **Log In** to log in to the Windows instance.
   This document uses logging in to a CVM instance on Windows Server 2016 Datacenter Edition 64-bit as an example. If the login is successful, a page similar to the following will appear:

Configuring Network Address
Enabling/Disabling Public Network Address

Last updated：2024-07-05 09:21:32

In addition to the private network access, you can also connect to TDSQL-C for MySQL by using the system-assigned domain name and port after enabling the public network access. It takes about five minutes for the configuration to take effect. Note that the public network access should be used only for database development or management. For business access, you should use the private network access.
This document describes how to enable/disable public read-write and read-only addresses of a cluster in the console.

Overview

A TDSQL-C for MySQL cluster contains read-write and read-only instances. They support both private and public network addresses, with the former enabled by default for you to access your instance over the private network and the latter enabled or disabled as needed. Note that the latter is automatically assigned by the system and cannot be customized currently.

Note:

After TDSQL-C for MySQL public network access is enabled, it will be controlled by the network access policies in the security group. You need to configure the corresponding policies in advance.
To access a cluster over the public network, you need to enable and configure a security group policy and **open the private network access port**; otherwise, public network access will fail.
Currently, the public network access feature is free of charge, but the stability of the public network bandwidth and traffic cannot be guaranteed.
The **Enable** button will be displayed for the public read-only address in the **Connection Info** section on the **Cluster Details** page only if there are read-only instances in your cluster.

CLB Architecture

Currently, after public network address is enabled for TDSQL-C for MySQL, it adopts a Cloud Load Balancer (CLB) architecture. The system will automatically create a basic CLB instance in the same region in the CLB console to provide public network capabilities. The CLB architecture offers detailed resource restriction policies (as shown in the table below). If you have higher performance requirements, you can also directly purchase a CLB instance.

Category	Number of Concurrent Connections	New Connections	Packet Volume	Inbound Bandwidth	Outbound Bandwidth

CLB	2000	200/s	Unlimited	20 Mbps	20 Mbps

Note:

A CLB instance is automatically created due to the enabling of a public network address. You can try it for free.

After the public network address is disabled, the corresponding CLB instance will be automatically deleted.

Supported Regions

Clusters in Guangzhou, Shanghai, Beijing, Chengdu, Chongqing, Nanjing, Hong Kong (China), Singapore, Seoul, Tokyo, Silicon Valley, Frankfurt, and Virginia regions will support the enabling of public network addresses. The availability is progressively being rolled out. Refer to the supported regions already opened in the console.

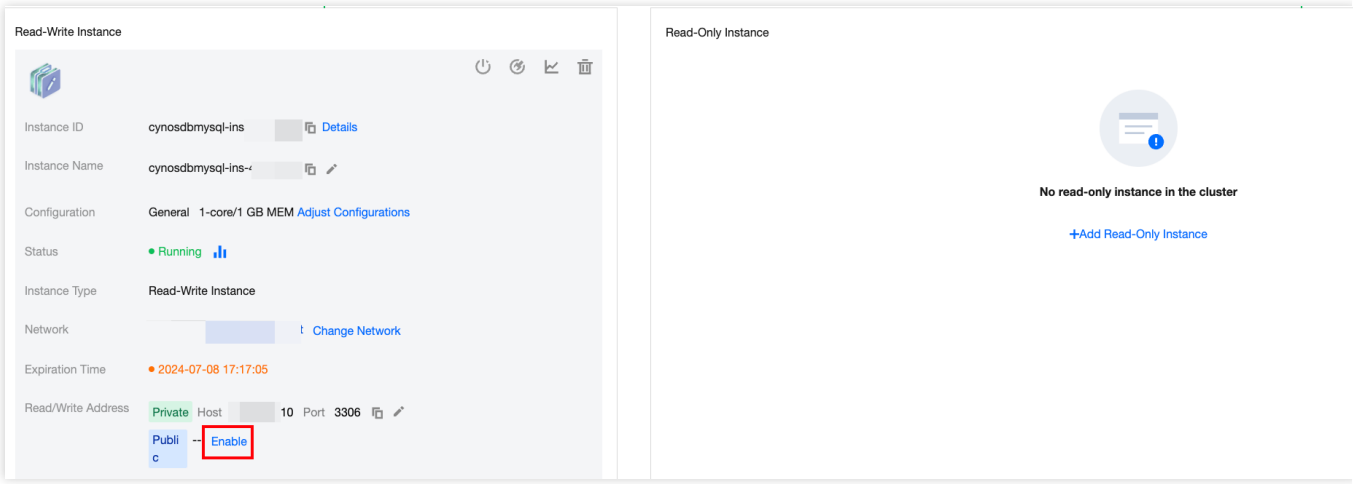
Enabling the public read-write/read-only addresses of a cluster

On the cluster management page, proceed according to the actually used view mode:

Tab View

List view

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. Under **Cluster Details**, locate the instance for which you want to enable the public network and click **Enable** next to the read-write/read-only address.



3. After you click **Enable**, the system will perform operations based on the instance's bound security group situation in several scenarios.

Scenario 1: The port is not opened in the current security group configuration.

To enable public network access, you need to click **Authorize and Create** (the system will request your authorization to automatically bind a security group that allows the internal network port for this cluster (current port: 3306) to facilitate your public network connection. Subsequently, you may access the security group interface for further settings or to bind a new security group).

Note:

Enabling the public network requires the internal network policy of the security group to be opened for access. If your current security group configuration does not allow the port, to ensure connection success, the system will automatically open the port policy for this cluster and allow 0.0.0.0/0 and ::/0 access. For network security, we recommend that you manually modify it to the fixed IP access later.

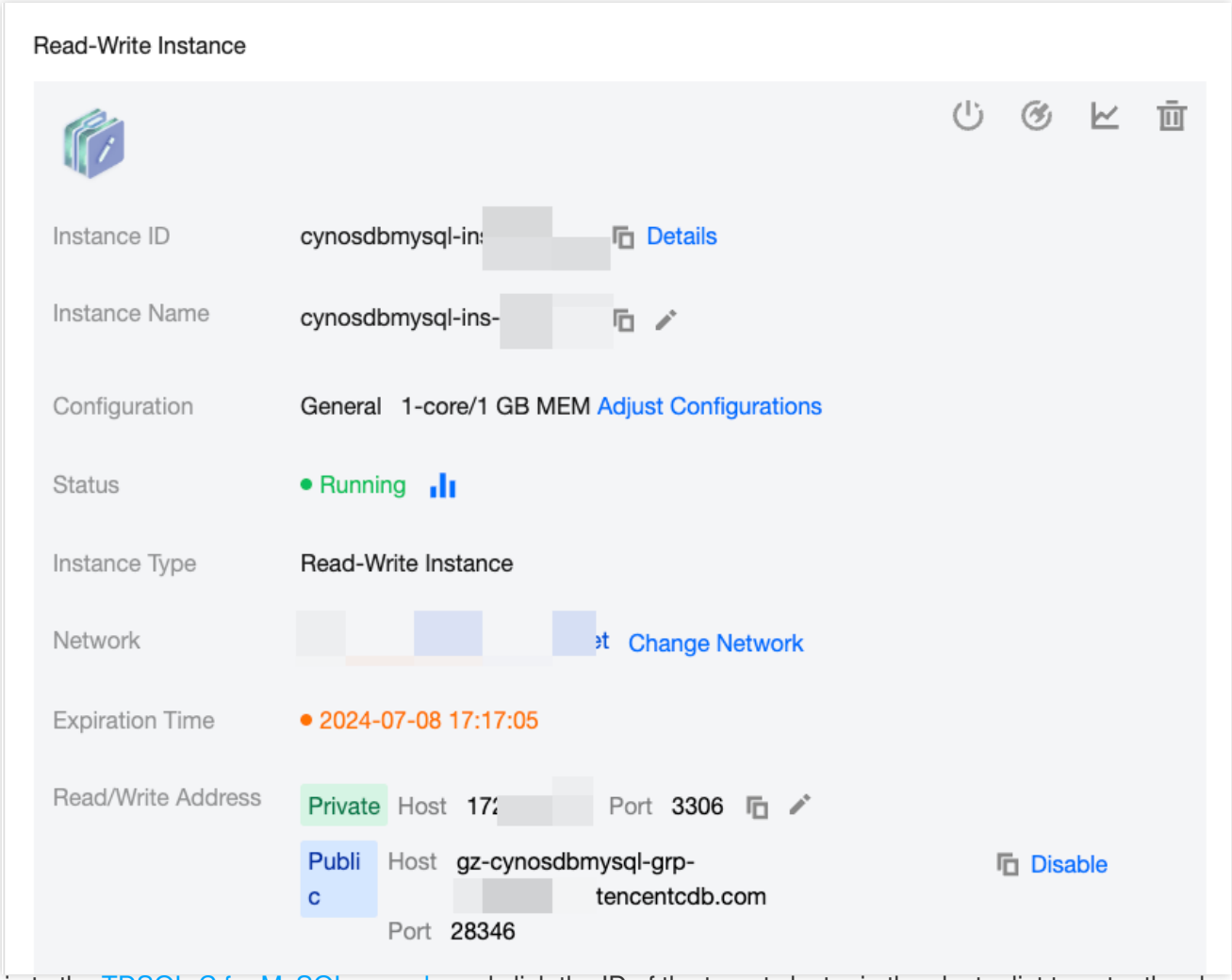
After enabling, you can access TDSQL-C for MySQL through a system-assigned domain name and port in the public network. The effective time is about 5 minutes.

Public network access is solely for developing or assisting in database management. For business access, use internal network access.

Scenario 2: The port is opened in the current security group configuration.

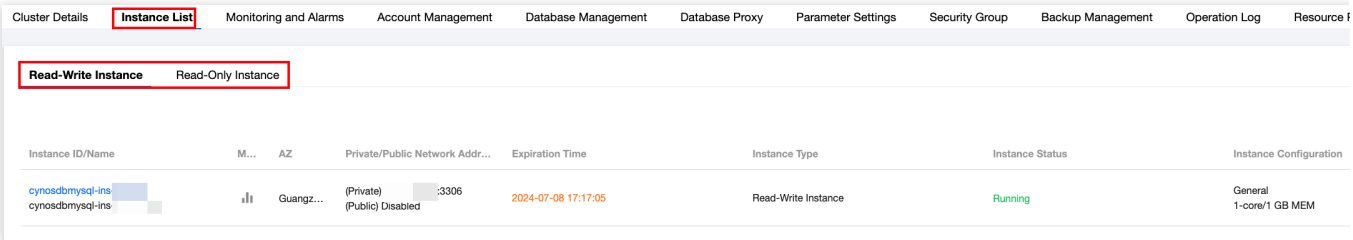
In the pop-up window, read the prompt and click **OK** to enable public network.

4. Once the network is successfully enabled, you can view the public network's host and port at the public network address under connection information. The public network host and port cannot be modified.

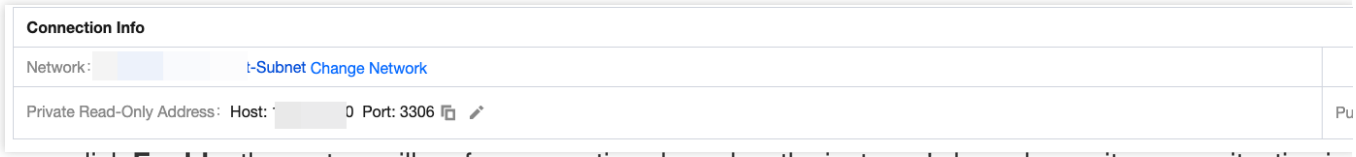


1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list to enter the cluster management page.

2. On the **Instance List** tab page of the cluster management page, select the instance (read/write instance or read-only instance) to be enabled for public network, click its **Instance ID** or **Manage** in the **Operation** column to enter the instance details page.



3. On the instance details page, choose **Connection Information** > **Public Network Address**, and then click **Enable** .



4. After you click **Enable**, the system will perform operations based on the instance's bound security group situation in several scenarios.

Scenario 1: The port is not opened in the current security group configuration.

To enable public network access, you need to click **Authorize and Create** (the system will request your authorization to automatically bind a security group that allows the internal network port for this cluster (current port: 3306) to facilitate your public network connection. Subsequently, you may access the security group interface for further settings or to bind a new security group).

Note:

Enabling the public network requires the internal network policy of the security group to be opened for access. If your current security group configuration does not allow the port, to ensure connection success, the system will automatically open the port policy for this cluster and allow 0.0.0.0/0 and ::/0 access. For network security, we recommend that you manually modify it to the fixed IP access later.

After enabling, you can access TDSQL-C for MySQL through a system-assigned domain name and port in the public network. The effective time is about 5 minutes.

Public network access is solely for developing or assisting in database management. For business access, use internal network access.

Scenario 2: The port is opened in the current security group configuration.

In the pop-up window, read the prompt and click **OK** to enable public network.

5. Once the network is successfully enabled, you can view the public network's host and port at the public network address under connection information. The public network host and port cannot be modified.

Disabling the public read-write/read-only addresses of a cluster

Note:

After the public network address is disabled, you will no longer be able to access the TDSQL-C for MySQL cluster through the public network domain name and port. Ensure your system does not use a public network access address to avoid unnecessary losses.

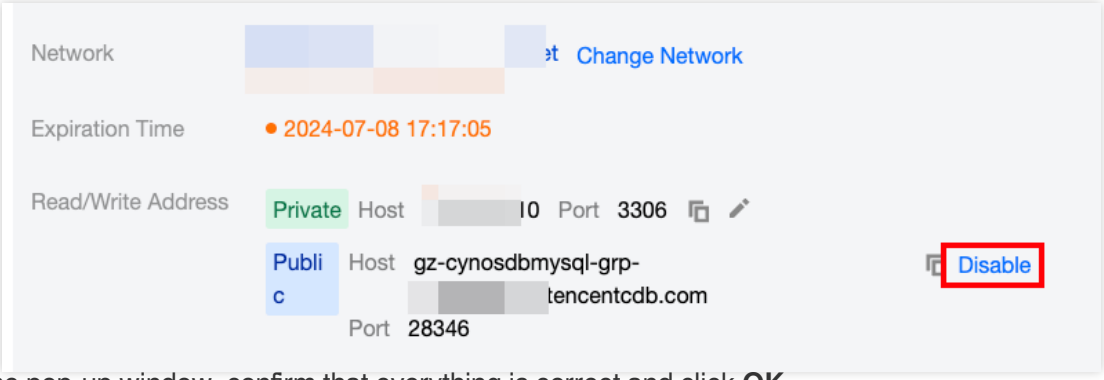
On the cluster management page, proceed according to the actually used view mode:

Tab View

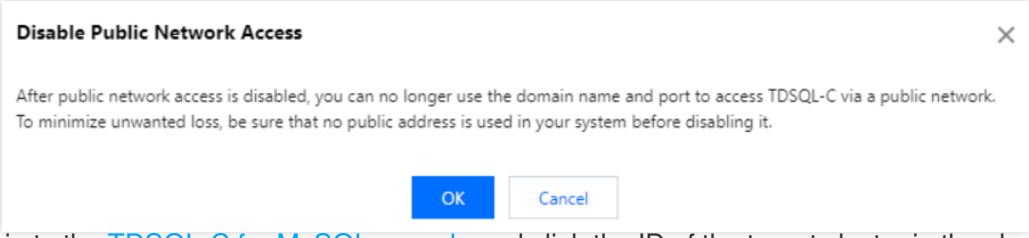
List View

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.

2. Under **Cluster Details**, locate the instance for which you want to enable the public network and click **Disable** next to the read-write/read-only address.

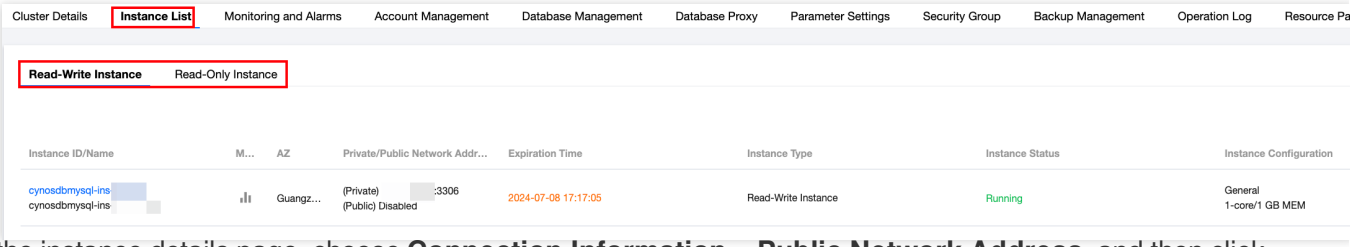


3. In the pop-up window, confirm that everything is correct and click **OK**.



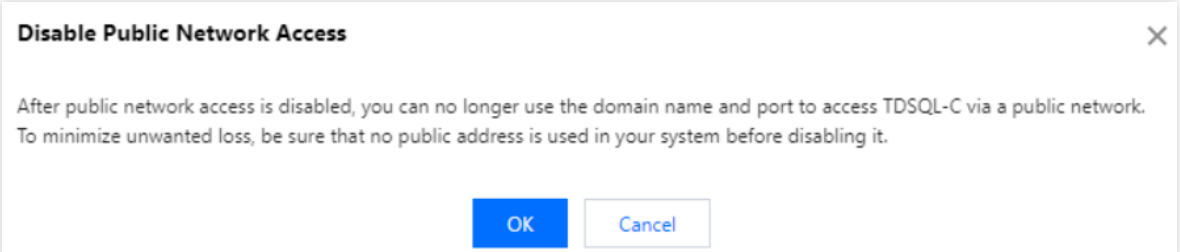
1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list to enter the cluster management page.

2. On the **Instance List** tab page of the cluster management page, select the instance (read/write instance or read-only instance) to be enabled for public network, click its **Instance ID** or **Manage** in the **Operation** column to enter the instance details page.



3. On the instance details page, choose **Connection Information** > **Public Network Address**, and then click **Disable** .

4. In the pop-up window, confirm that everything is correct and click **OK**.



Modifying Private Network Address

Last updated：2023-11-20 10:10:29

This document describes how to modify the private read-write/read-only addresses of TDSQL-C for MySQL in the console.

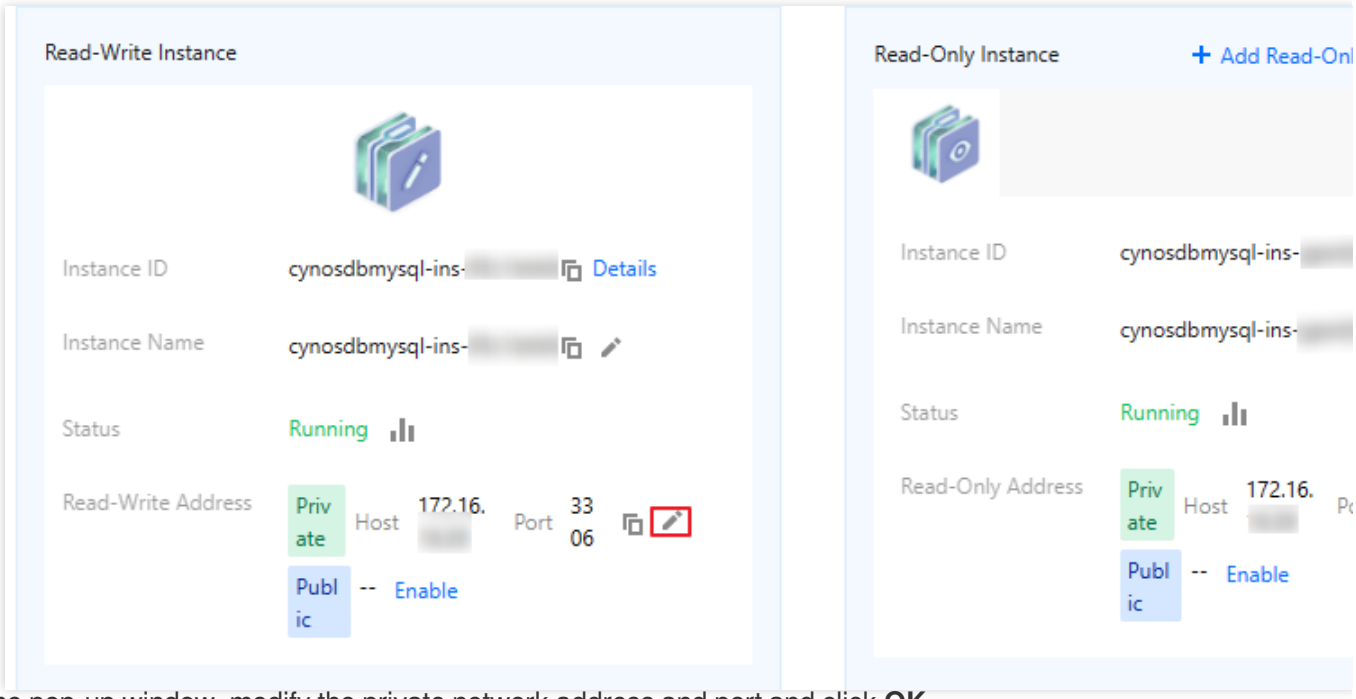
Directions

On the cluster management page, proceed according to the actually used view mode:

Tab view

List view

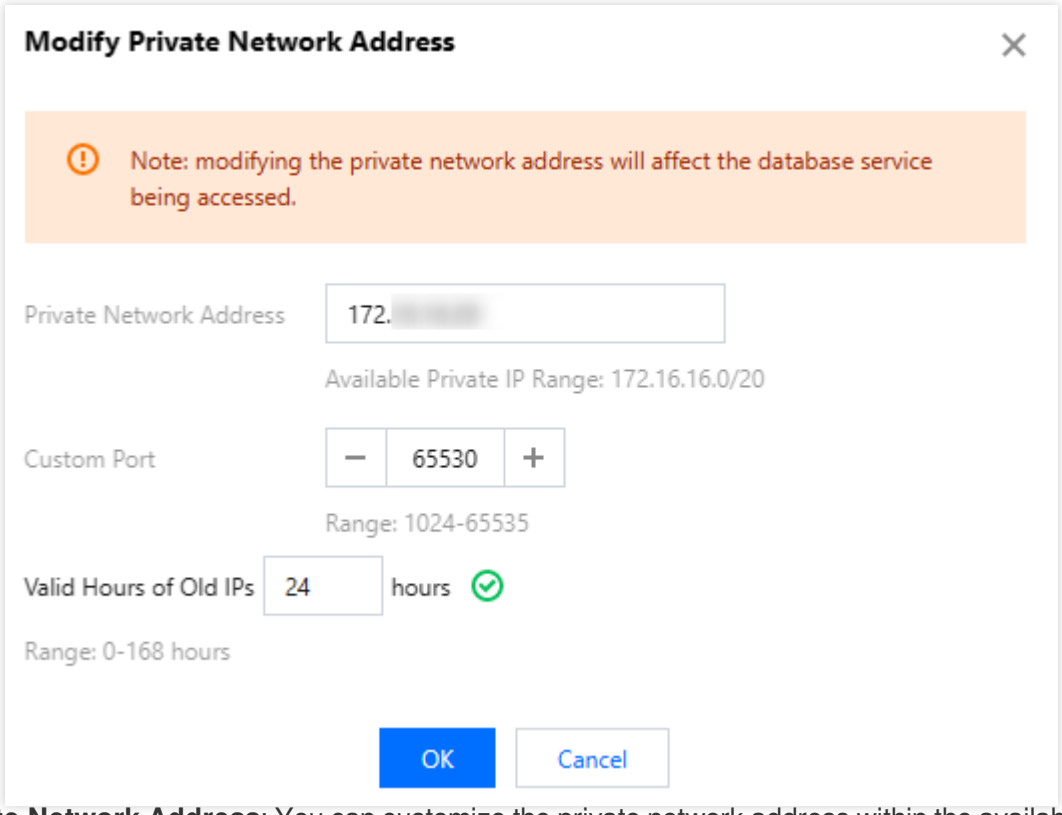
1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. On the **Cluster Details** page, find the target instance and click  after its private network address.



3. In the pop-up window, modify the private network address and port and click **OK**.

Note:

Modifying the private network address affects the database business being accessed.

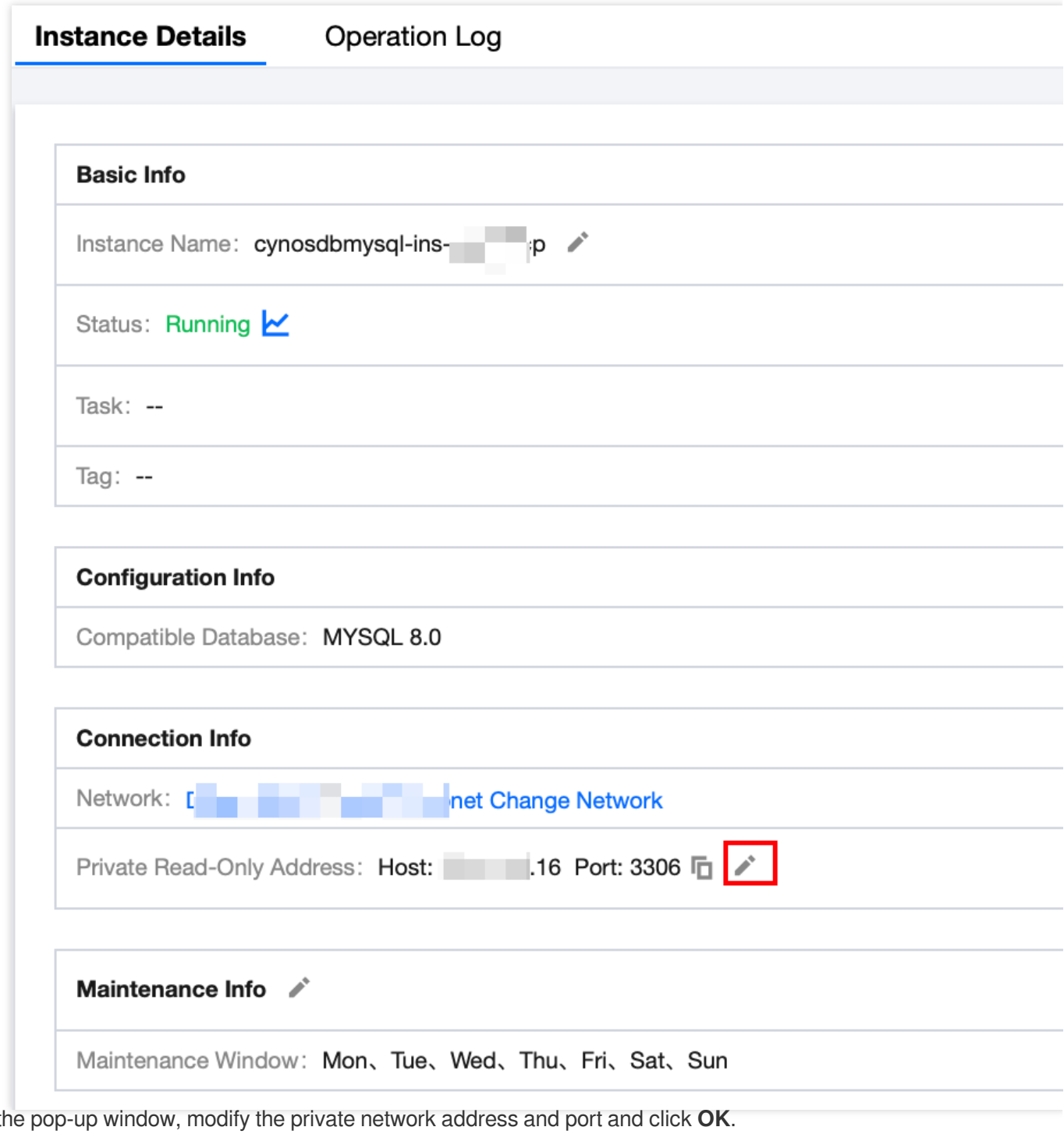


Private Network Address: You can customize the private network address within the available IP range.

Custom Port: You can customize the port in the range of 1024–65535.

Valid Hours of Old IPs: The default value is 24 hours, meaning that you can still access the instance at the old IP within 24 hours. You can customize the value between 0 and 168 hours. If you set the value to 0, the old IP will be repossessed immediately, and then you cannot access the instance at it.

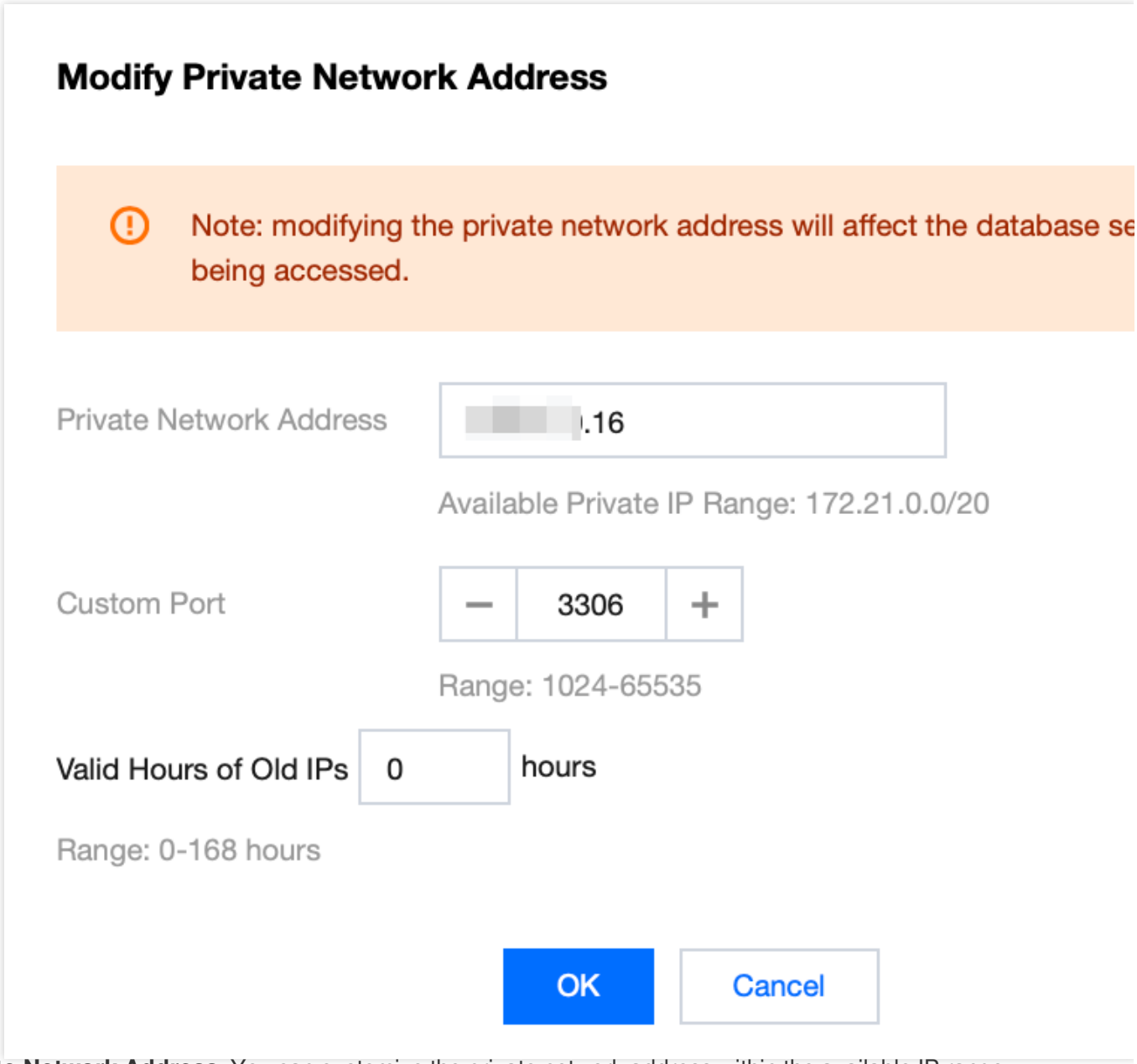
1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list to enter the cluster management page.
2. Select the **Instance List** tab, select the target read-write or read-only instance, and click its ID to enter instance details page.
3. On the **Instance Details** page, click after the private network address in the **Connection Info** section.



4. In the pop-up window, modify the private network address and port and click **OK**.

Note:

Modifying the private network address affects the database business being accessed.



Private Network Address: You can customize the private network address within the available IP range.

Custom Port: You can customize the port in the range of 1024–65535.

Valid Hours of Old IPs: The default value is 24 hours, meaning that you can still access the instance at the old IP within 24 hours. You can customize the value between 0 and 168 hours. If you set the value to 0, the old IP will be repossessed immediately, and then you cannot access the instance at it.

Changing Network

Last updated：2023-07-13 11:05:59

This document describes how to change the network of TDSQL-C for MySQL in the console.

Overview

TDSQL-C for MySQL supports VPC, which is capable of offering a diversity of smooth services. On this basis, it provides the cluster network change feature to help you manage network connectivity with ease.

Notes

- When the network is changed, all of the cluster's private IPs are automatically replaced with new ones. You should modify the client program accordingly in time.
- You can specify the remaining validity period of the old IPs in the console. The period is 24 hours by default. If you set the period to 0 hours, the old IPs will be released immediately after the network is changed.
- The new VPC and subnet should be in the region of the cluster.

Subnet description

A subnet is a logical network space in a VPC. You can create subnets in different AZs in the same VPC, which communicate with each other over the private network by default. Even if you select a subnet IP in another AZ in the region of the cluster, the network latency will not be increased because the actual business connection adopts nearby access.

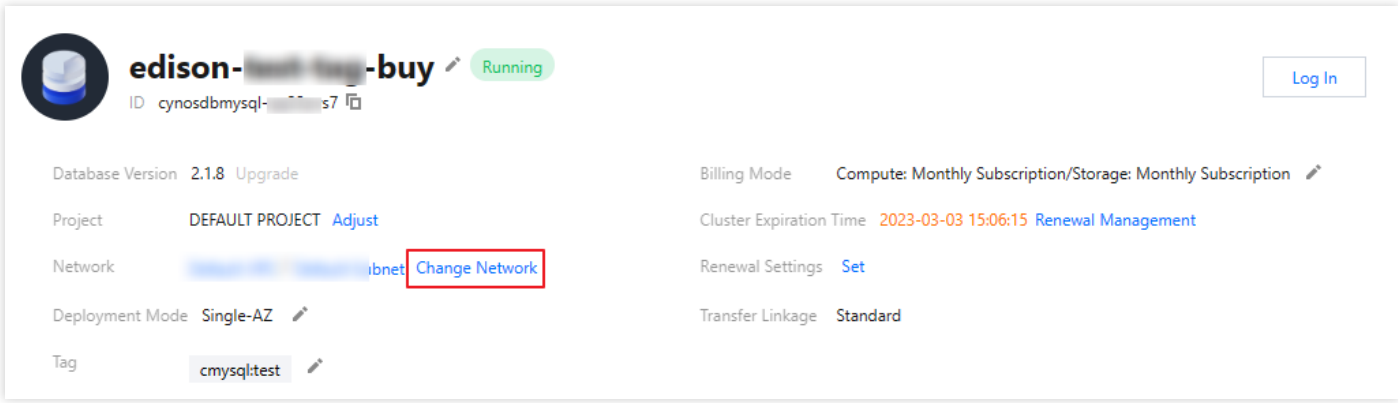
Directions

On the cluster management page, proceed according to the actually used view mode:

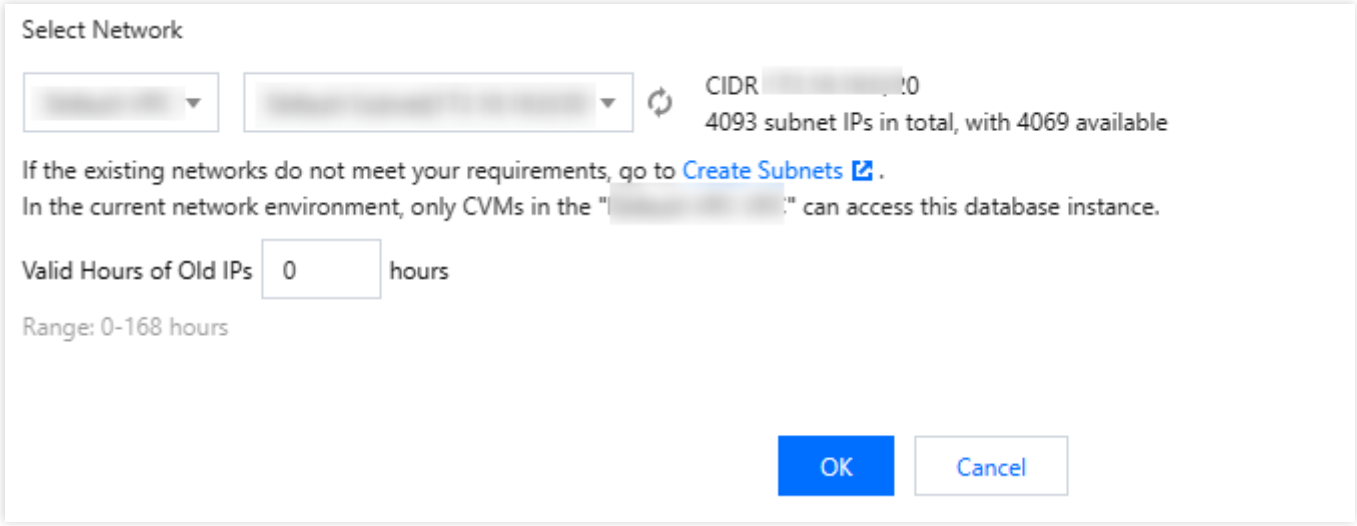
- Tab view
- List view

1. Log in to the TDSQL-C for MySQL console.
2. Click the target cluster in the cluster list on the left to enter the cluster management page.

3. On the cluster management page, click **Change Network** after **Network**.



4. In the **Change Network** pop-up window, select a network, set **Valid Hours of Old IPs**, and click **OK**.



5. After changing the network, you can view the new network in the **Network** section on the cluster management page, and view the new private network address in the **Basic Info** section on the cluster details page.

Connecting to Cluster

Connecting to Cluster via the Console

Last updated：2023-11-09 16:16:31

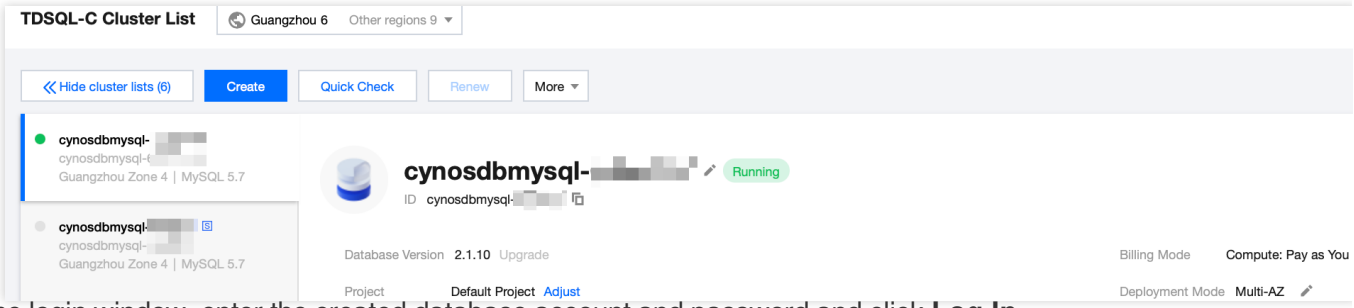
Database Management Center (DMC) is a one-stop Tencent Cloud database management tool. It provides a variety of services, including database/table-level operations, real-time monitoring, instance session management, SQL window, and data management.
This document describes how to connect to a TDSQL-C for MySQL cluster in DMC.

Prerequisite

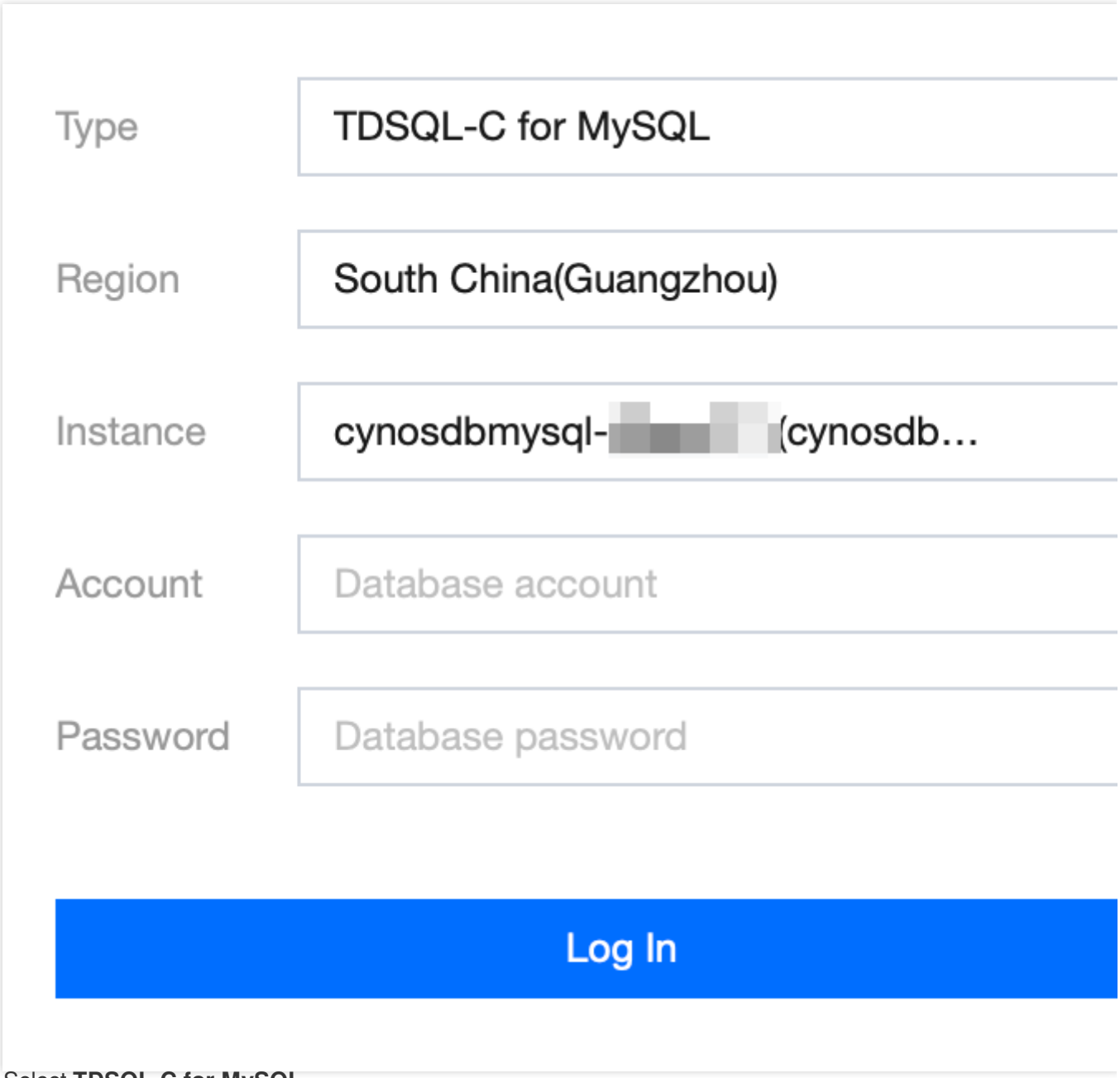
You have created a database cluster account. For more information, see Creating Account.

Directions

On the cluster list page, proceed based on the actually used view mode.
Tab view
List view
1. Log in to the TDSQL-C for MySQL console, and select a region at the top.
2. Click the target cluster in the cluster list on the left to enter the cluster management page.
3. On the cluster management page, click **Log In** on the right.



4. In the login window, enter the created database account and password and click **Log In**.



Type: Select **TDSQL-C for MySQL**.

Region: Select the region of the cluster.

Instance: Select the cluster to be connected to, or search for the cluster by ID in the drop-down list.

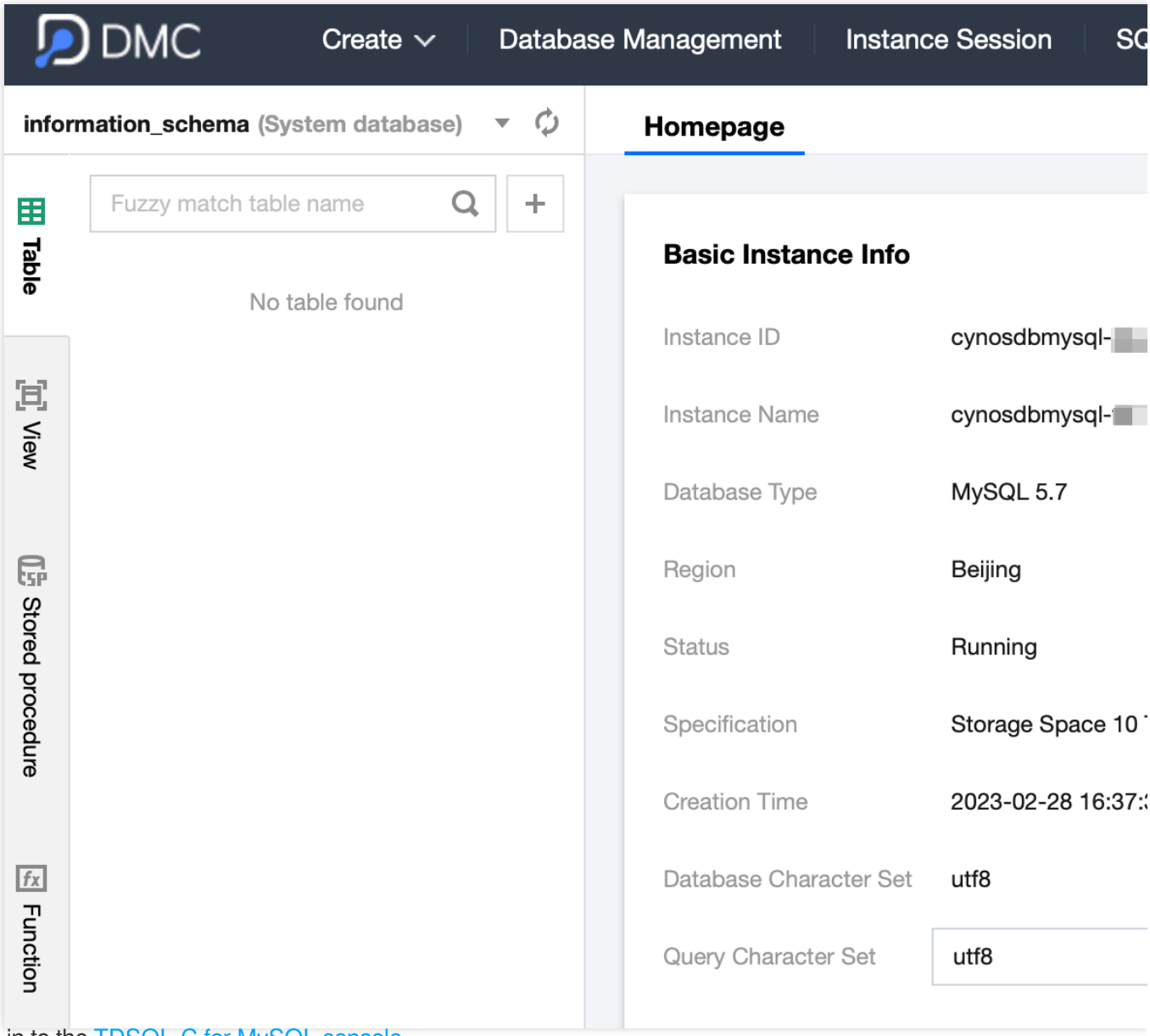
Account: Enter the account name corresponding to the cluster.

Password: Enter the password corresponding to the account.

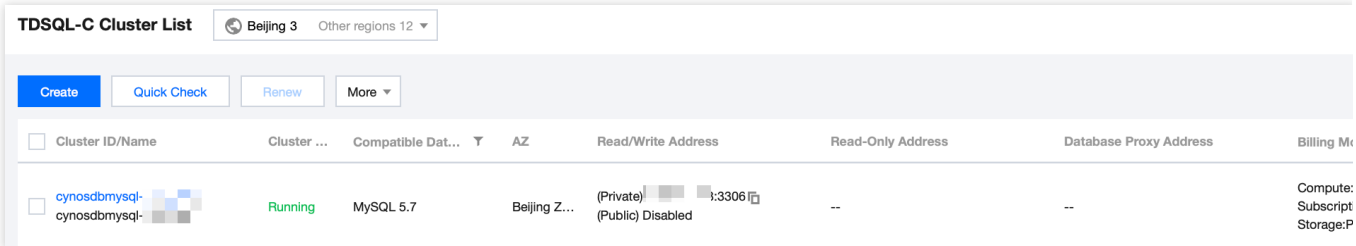
Note

The cluster ID can be obtained in the **Cluster List** or on the **Cluster Details** page in the TDSQL-C for MySQL console.

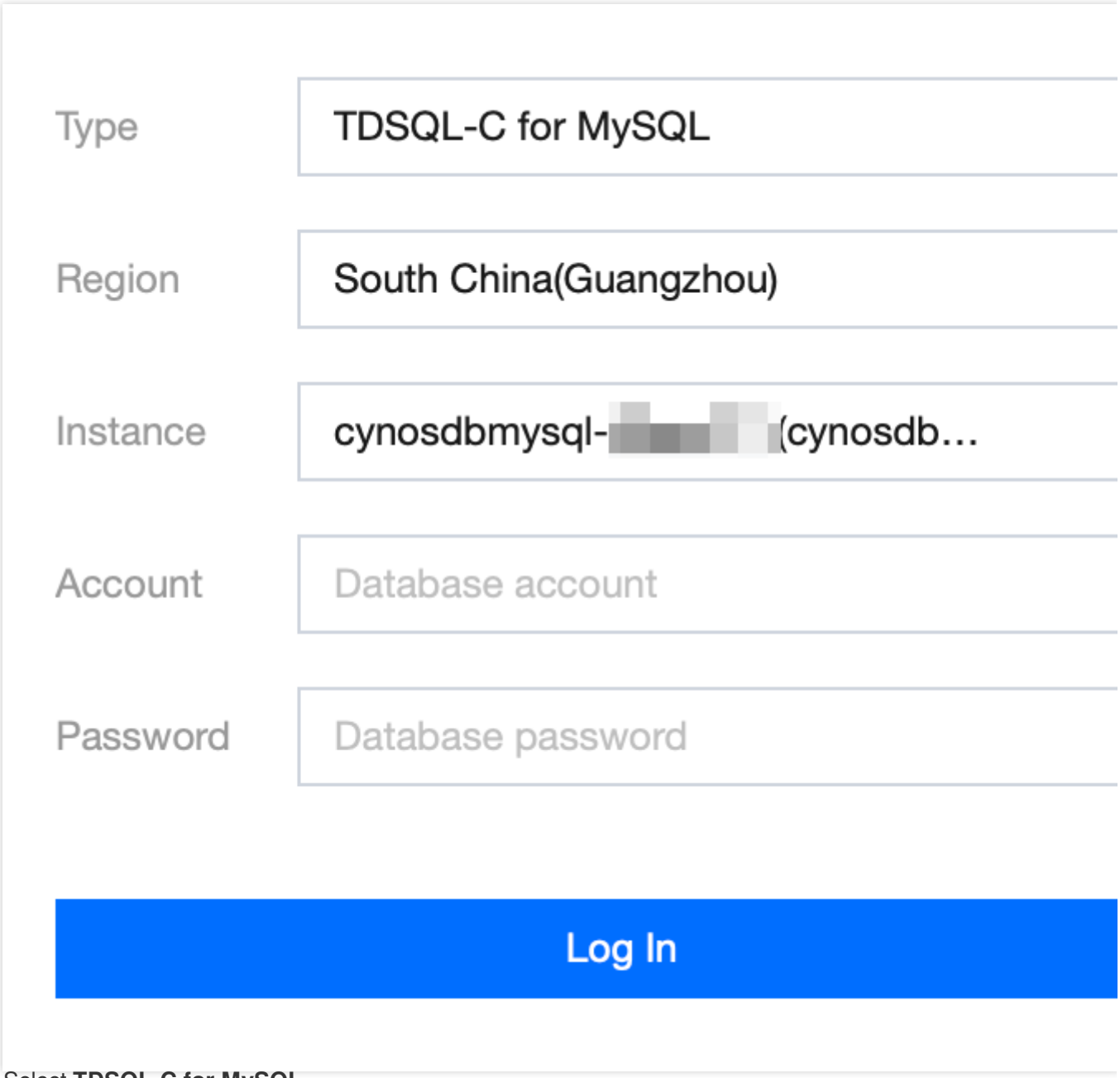
5. The UI after successful login is as follows:



1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click **Log In** in its **Operation** column.



3. In the login window, enter the created database account and password and click **Log In**.



Type: Select **TDSQL-C for MySQL**.

Region: Select the region of the cluster.

Instance: Select the cluster to be connected to, or search for the cluster by ID in the drop-down list.

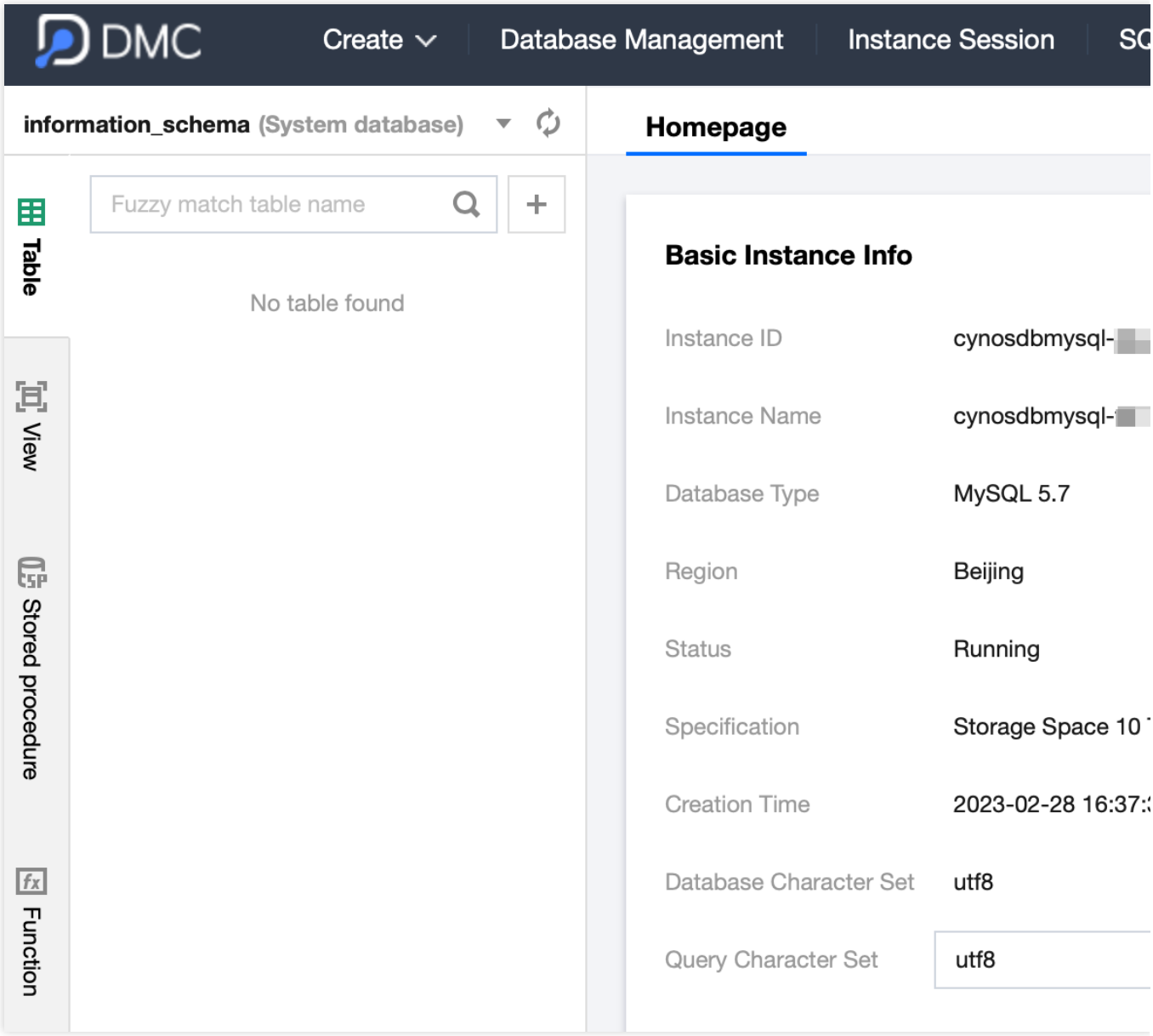
Account: Enter the account name corresponding to the cluster.

Password: Enter the password corresponding to the account.

Note

The cluster ID can be obtained in the **Cluster List** or on the **Cluster Details** page in the TDSQL-C for MySQL console.

4. The UI after successful login is as follows:



Other ways to connect to a cluster

Connecting to Cluster at Private or Public Network Address from Linux CVM Instance

Connecting to Cluster at Private or Public Network Address from Windows CVM Instance

Connecting to Cluster at Private or Public Network Address from Linux CVM Instance

Last updated：2023-11-01 17:06:45

TDSQL-C for MySQL supports instance-level dedicated IP address, which allows you to connect to the instance through the IP address of a read-write or read-only instance in a cluster. This document describes how to connect to an instance in a cluster from a Linux CVM instance via private or public network.

Prerequisite

You have created a database cluster account. For more information, see Creating Account.

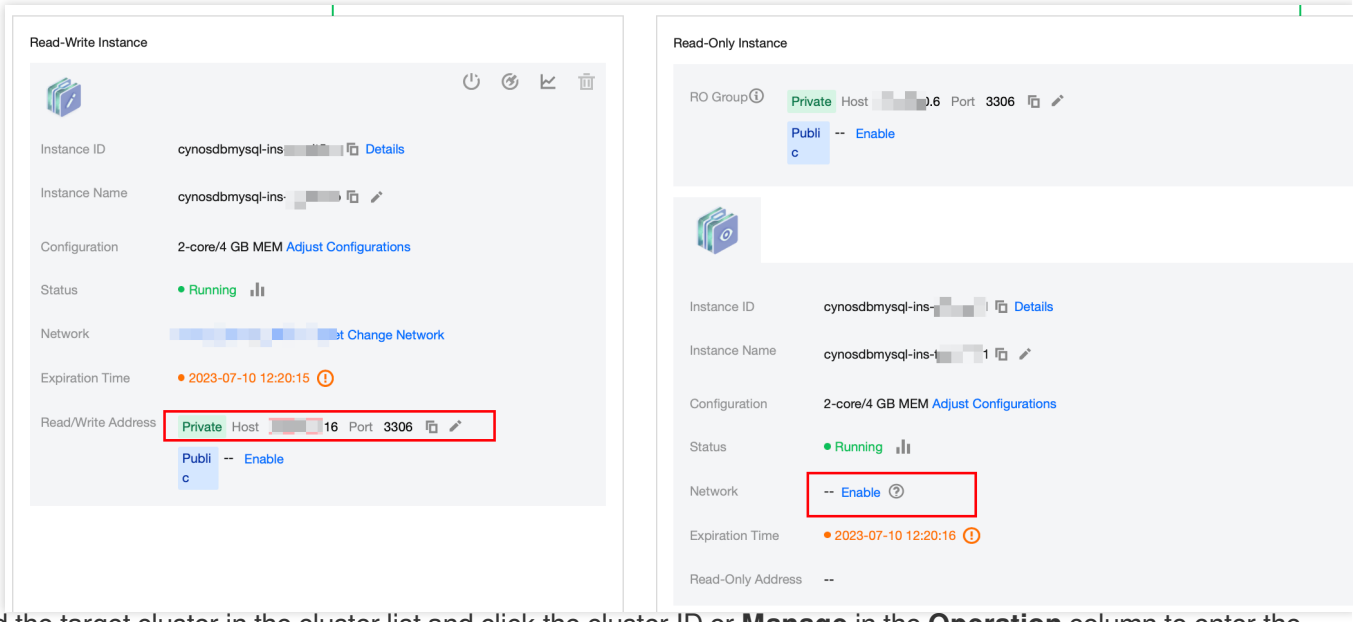
Step 1. Query the private and public IP address of the instance to be connected.

1. Log in to the TDSQL-C for MySQL console.
2. On the cluster list page, proceed based on the actually used view mode.

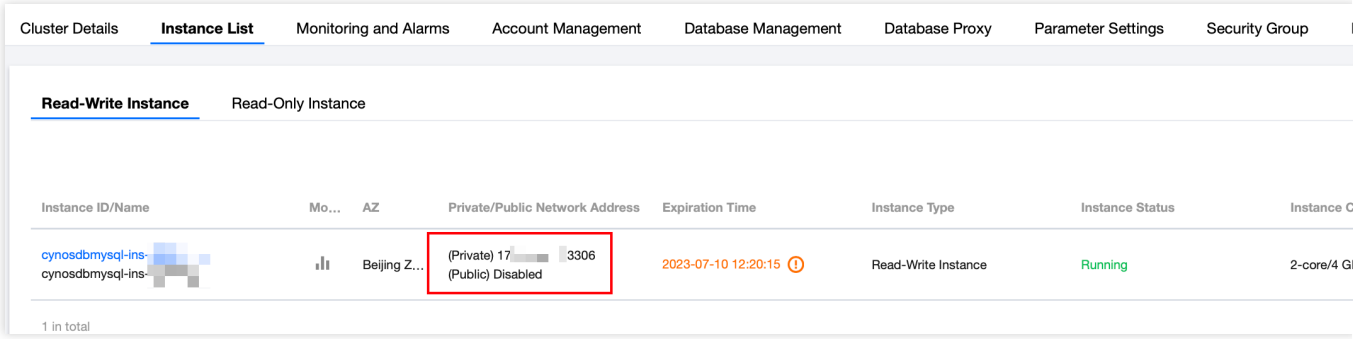
Tab view

List view

1. Click **Target Cluster** in the cluster list on the left to enter the cluster management page.
2. On the **Cluster Details** tab, find the target instance, and view its private and public network IP address under **Network**.

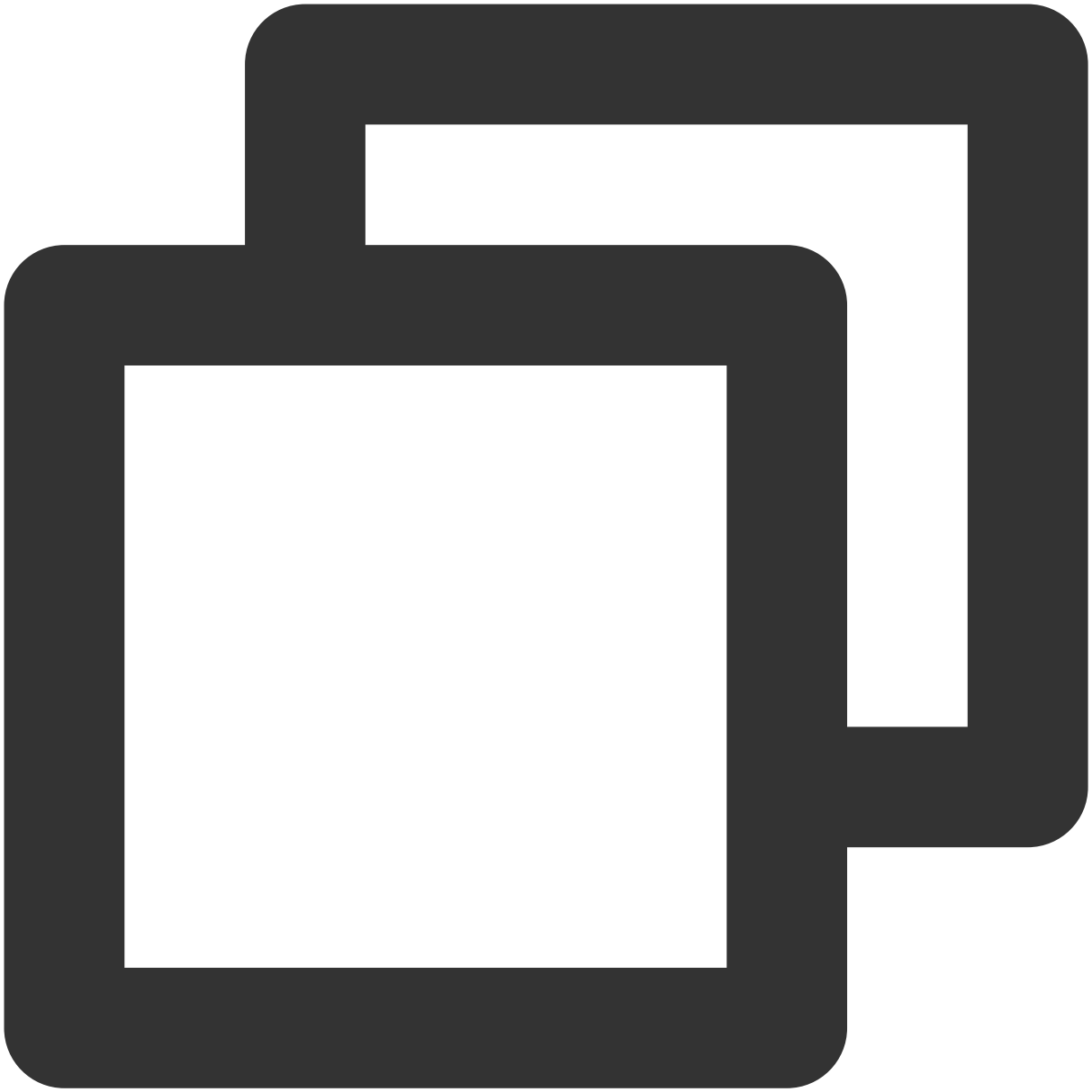


1. Find the target cluster in the cluster list and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
2. On the cluster management page, select the read-write or read-only instance on the instance list page, and you can view the instance private and public network IP address under **Private/Public Network Address**.



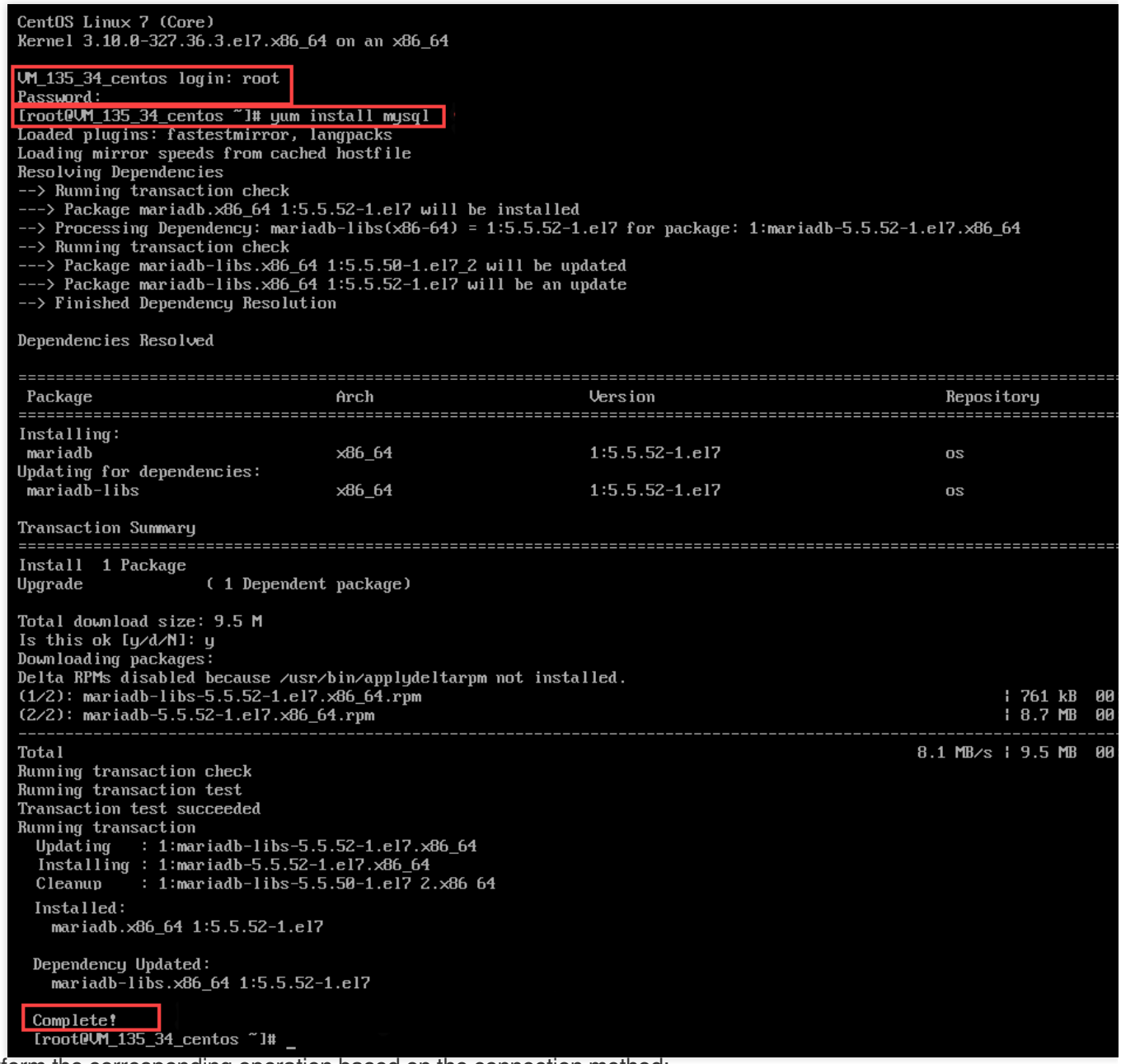
Step 2. Connect to the target instance in a cluster

1. Log in to a Linux CVM instance. For more information, see Quickly Configuring Linux CVM Instance.
2. Taking a CVM instance on CentOS 7.2 (64-bit) as an example, run the following command to install the MySQL client.



```
yum install mysql
```

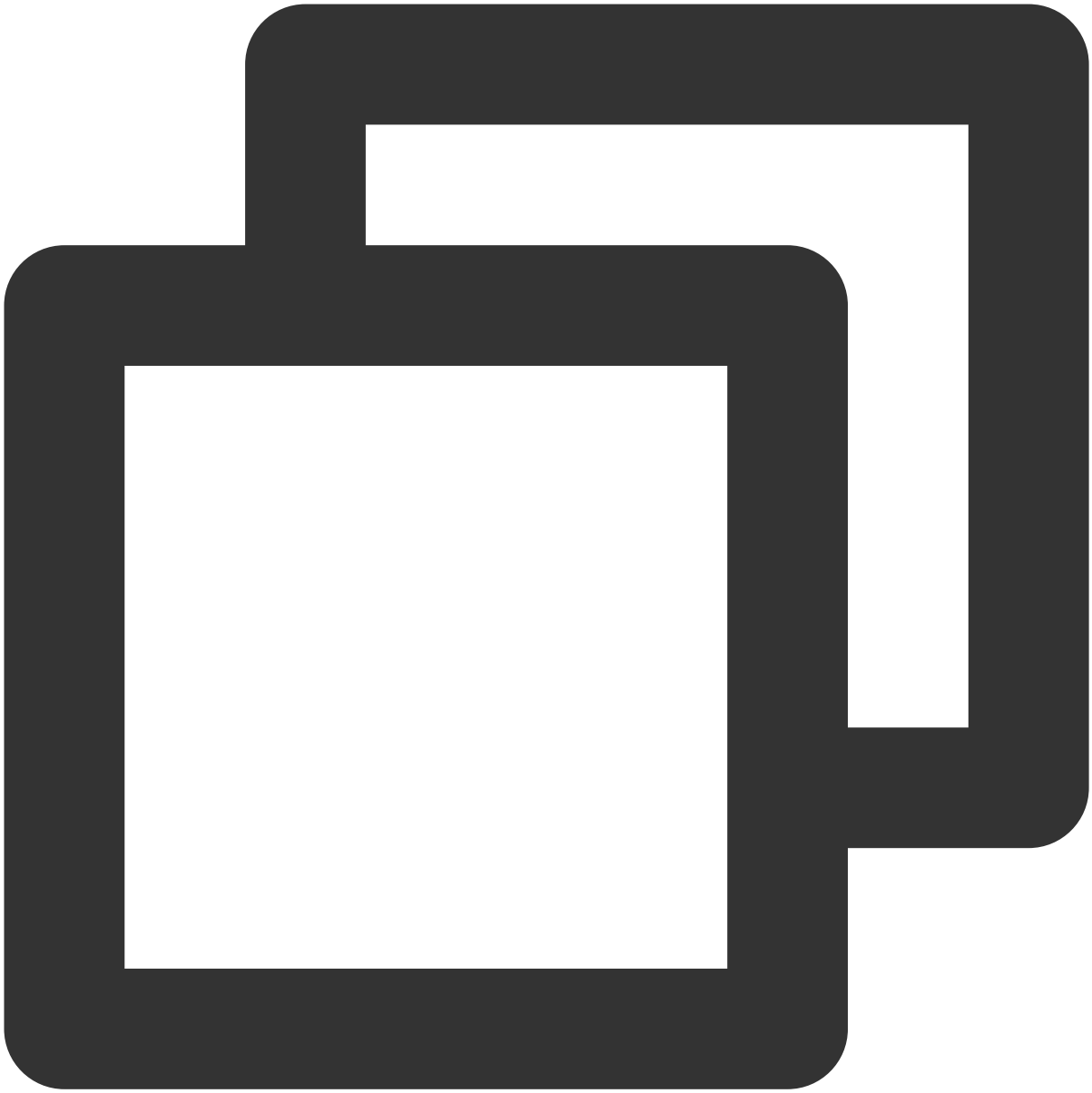
If `Complete!` is displayed, the MySQL client is installed successfully.



3. Perform the corresponding operation based on the connection method:

Private network connection:

3.1 Run the following command to log in to the TDSQL-C for MySQL cluster.



```
mysql -h hostname -P port -u username -p
```

hostname: Replace it with the private network address of the target TDSQL-C for MySQL cluster as instructed in Step 1.

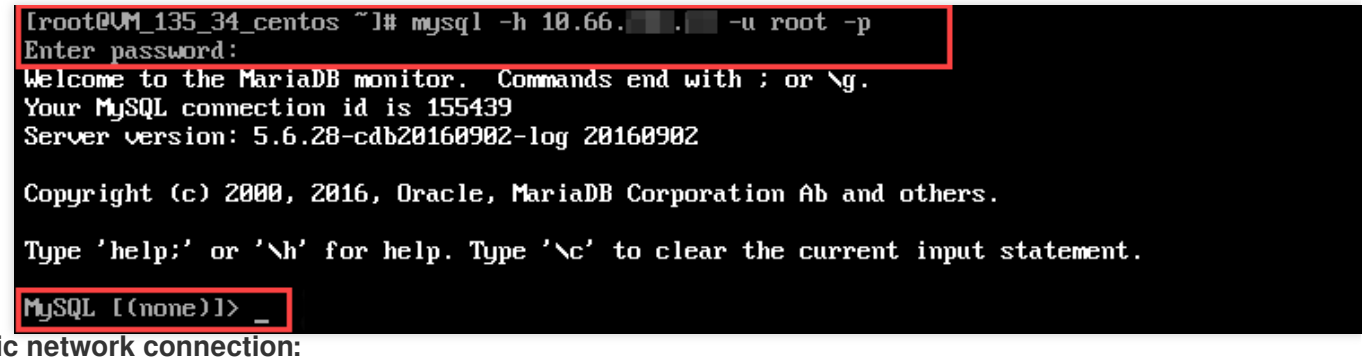
port: Replace it with the private network port number.

username: Replace it with the name of the account created in Prerequisites, which is the default account name `root` here for example.

For example, if the private network address is `10.0.168.14:5308` and the account name is `root`, enter the following connection command: `mysql -h 10.0.168.14 -P 5308 -u root -p`.

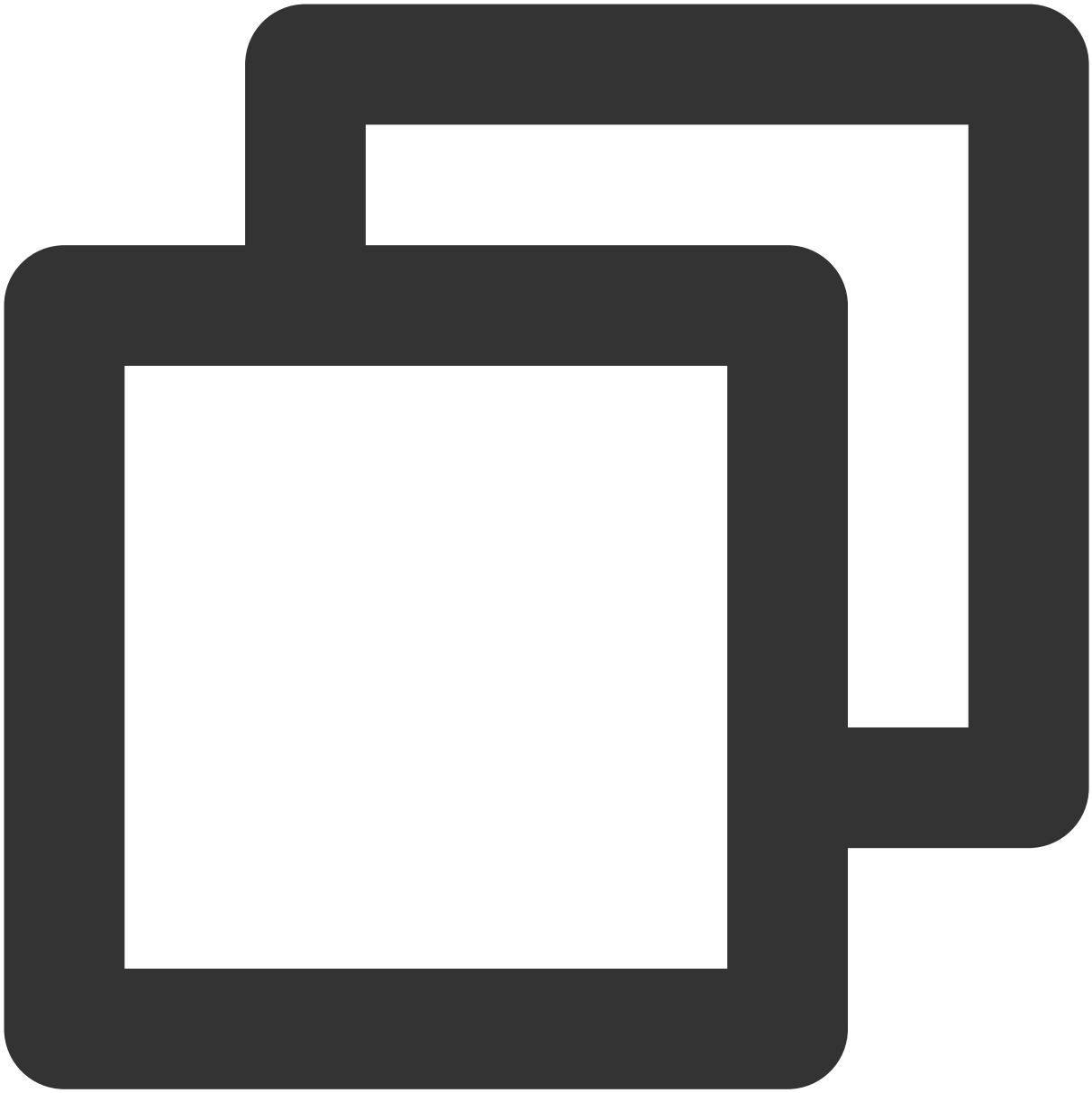
3.2 Enter the password corresponding to the account in the above command after `Enter password:` is prompted. If you forgot the password, you can reset it as instructed in Resetting Password.

If `MySQL [(none)]>` is displayed, you have logged in to TDSQL-C for MySQL successfully.



Public network connection:

3.3 Run the following command to log in to the TDSQL-C for MySQL cluster.



```
mysql -h hostname -P port -u username -p
```

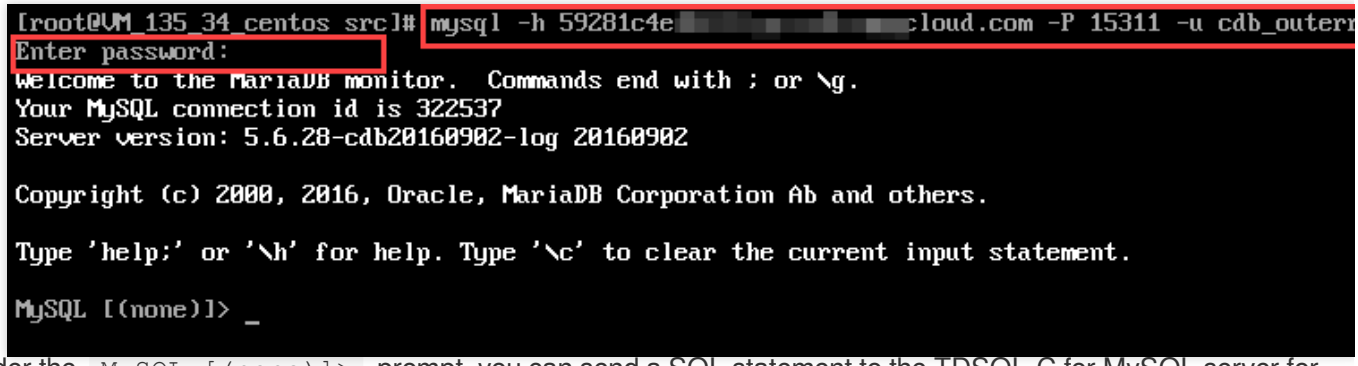
hostname: Replace it with the public network address of the target TDSQL-C for MySQL cluster as instructed in Step 1. If public network address is not enabled, enable it as instructed in Enabling/Disabling Public Network Address.

port: Replace it with the public network port number.

username: Replace it with the account name for public network connection. We recommend that you create a separate account in the console for easier connection control.

3.4 Enter the password corresponding to the account name for public network connection after `Enter password:` is prompted. If you forgot the password, reset it as instructed in Resetting Password.

In this example, `hostname` is 59281c4exxx.myqcloud.com and public network port is 15311.



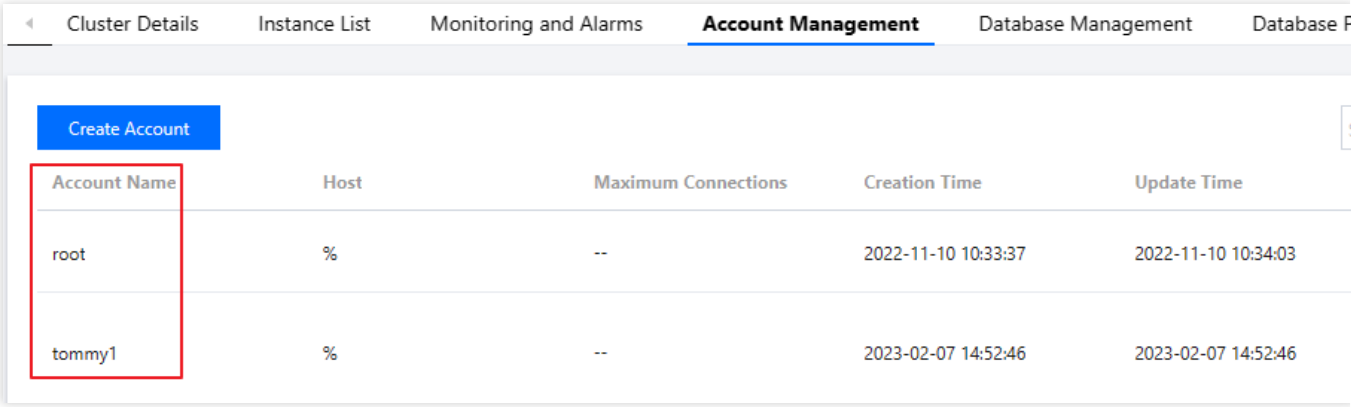
4. Under the `MySQL [(none)]>` prompt, you can send a SQL statement to the TDSQL-C for MySQL server for execution. For specific command lines, see mysql Client Commands.

Below take `show databases;` as an example:

FAQs

How do I find the account name or reset the password to log in to the cluster?

You can view the account name you created for cluster connection under **Account Management** on the cluster management page in the TDSQL-C for MySQL console. If you forgot the password, perform the **Reset Password** operation to change the password.

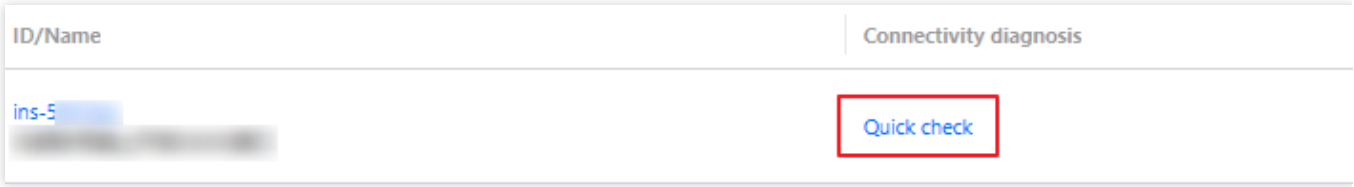


Do I need to configure a security group for the TDSQL-C for MySQL cluster when connecting to it over the private or public network?

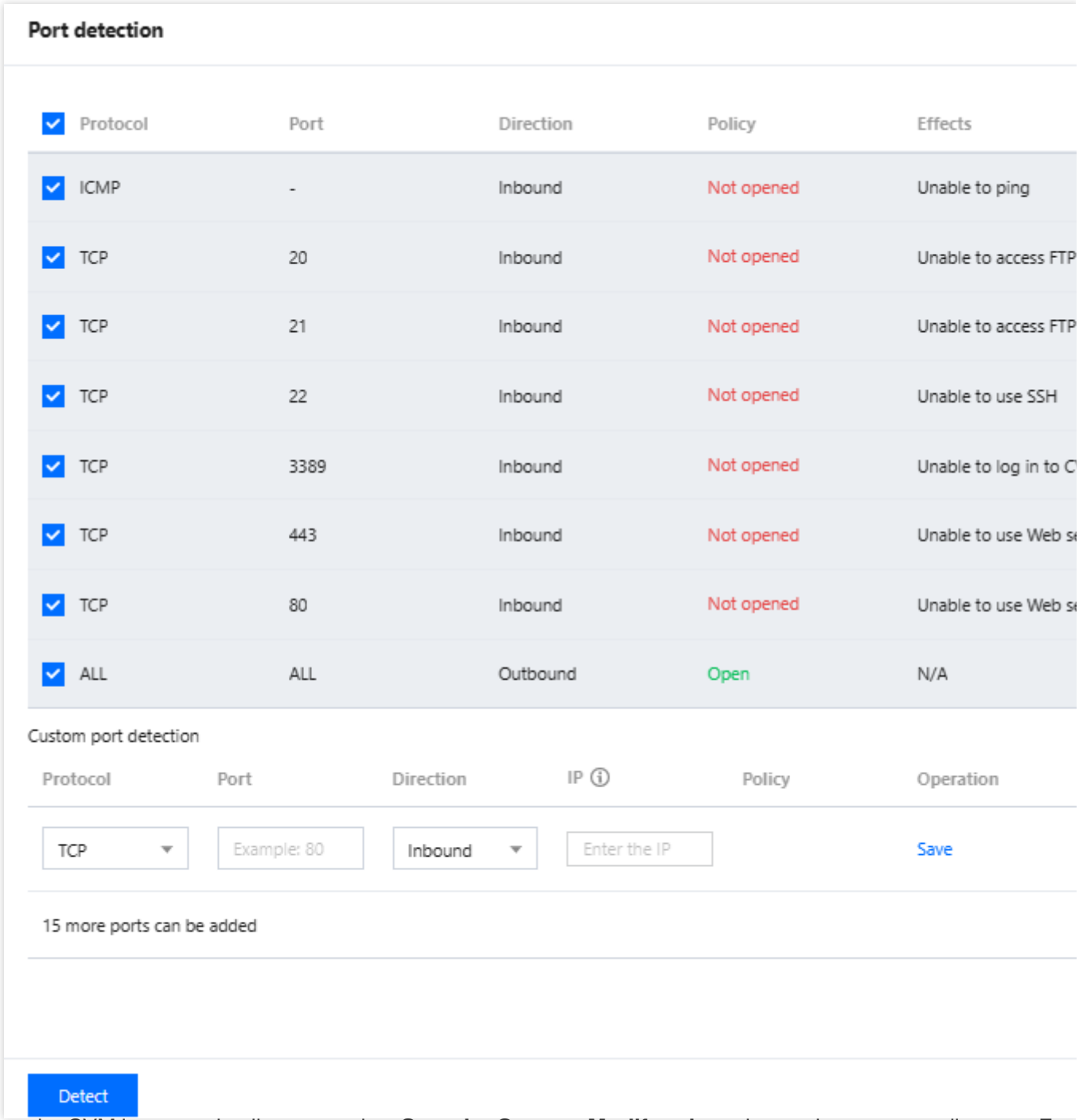
Yes. For detailed directions, see Creating and Managing TencentDB Security Groups. Note that if the public network access is enabled for connection over the public network, the private network port needs to be opened in the configured security group rule.

What should I do if I can't log in to the Linux CVM instance because the security group rules of the instance don't match properly?

If the security group rules of the Linux CVM instance don't match properly (for example, the corresponding port is not opened), and you can't log in to the instance, you can use the Security Group (Port) Verification Tool to check the connectivity of the security group.



You can identify possible causes for login failures by quick check. The port numbers to be opened may vary depending on the actual operating system of your CVM. For Linux CVM, the default port is usually 22 or 36000; for Windows CVM, it is usually 3389.



Then, on the CVM instance details page, select **Security Group** > **Modify rule** and open the corresponding port. For detailed directions, see Adding Security Group Rules.

What settings may cause a cluster connection failure?

Check whether the CVM instance and the TDSQL-C for MySQL cluster are in the same VPC in the same region under the same Tencent Cloud account. If any of these three prerequisites is not meet, the cluster cannot be connected.

Connecting to Cluster at Private or Public Network Address from Windows CVM Instance

Last updated：2023-09-12 14:47:55

TDSQL-C for MySQL supports instance-level dedicated IP address, which allows you to connect to the instance through the IP address of a read-write or read-only instance in a cluster. This document describes how to connect to an instance in a cluster from a Windows CVM instance via private or public network.

Prerequisites

You have created a database cluster account. For more information, see Creating Account.

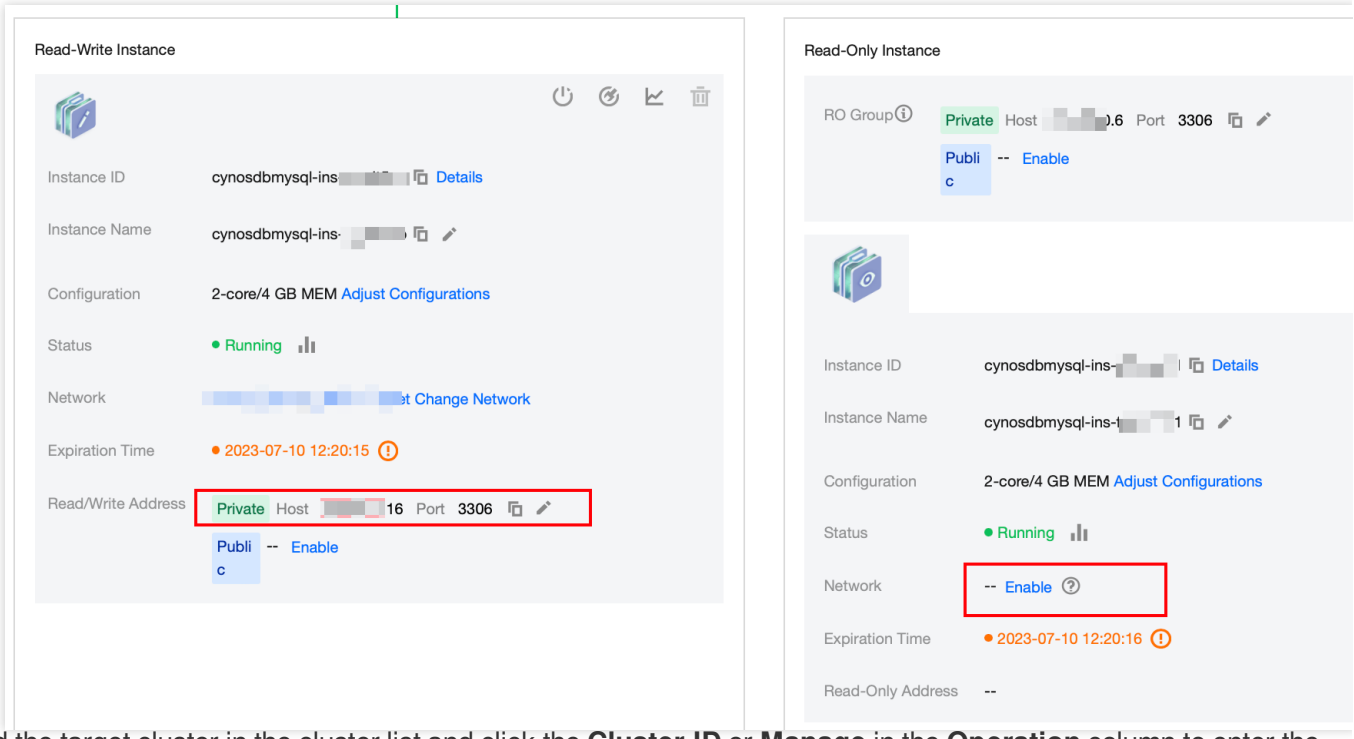
Step 1. Query the private and public IP address of the instance to be connected.

1. Log in to the TDSQL-C for MySQL console.
2. On the cluster list page, proceed based on the actually used view mode.

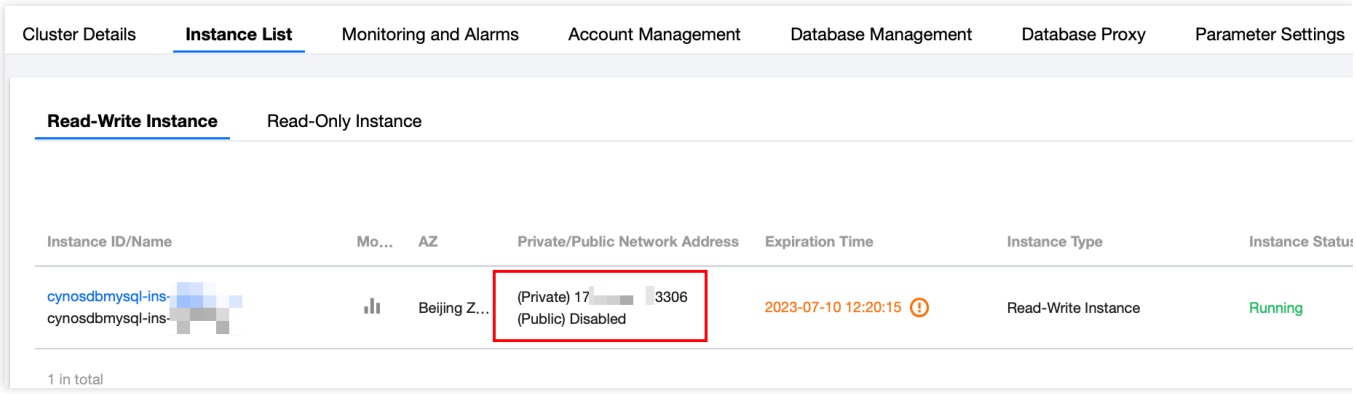
Tab view

List view

1. Click the target cluster in the cluster list on the left to enter the cluster management page.
2. On the **Cluster Details** tab, find the target instance, and view its private and public network IP address under **Network**.



1. Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.
2. On the cluster management page, select the read-write or read-only instance on the instance list page, and you can view the instance private and public network IP address under **Private/Public Network Address**.

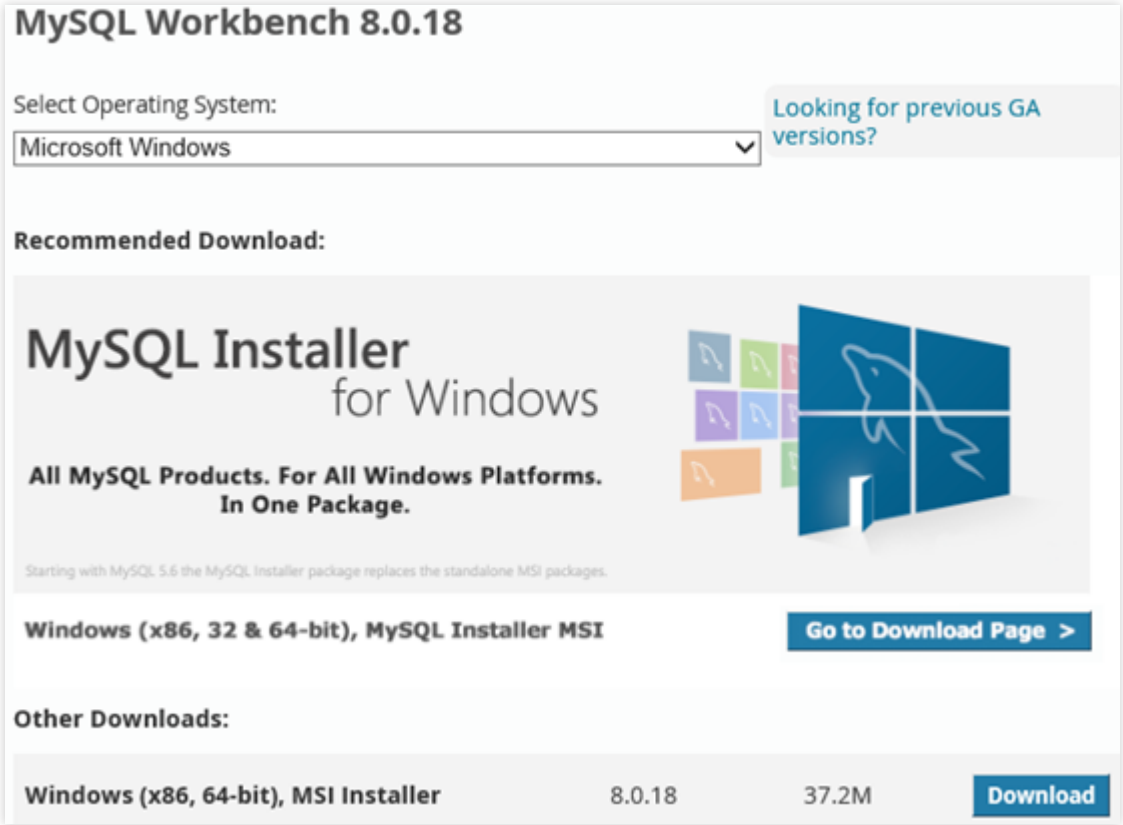


Step 2. Connect to the target instance in a cluster

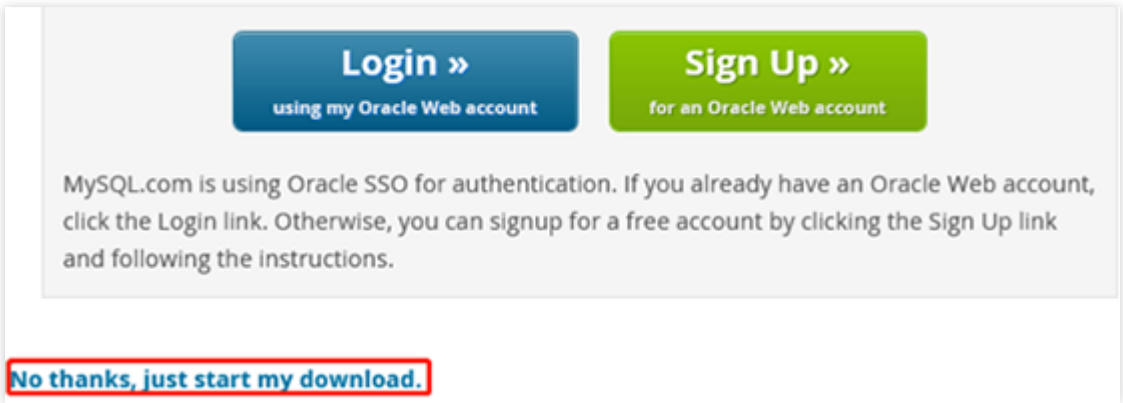
1. Log in to a Windows CVM instance. For more information, see Quickly Configuring Windows CVM Instance.
2. Download a standard SQL client.

Note

We recommend that you download MySQL Workbench. Click here and download an installer based on your operating system.



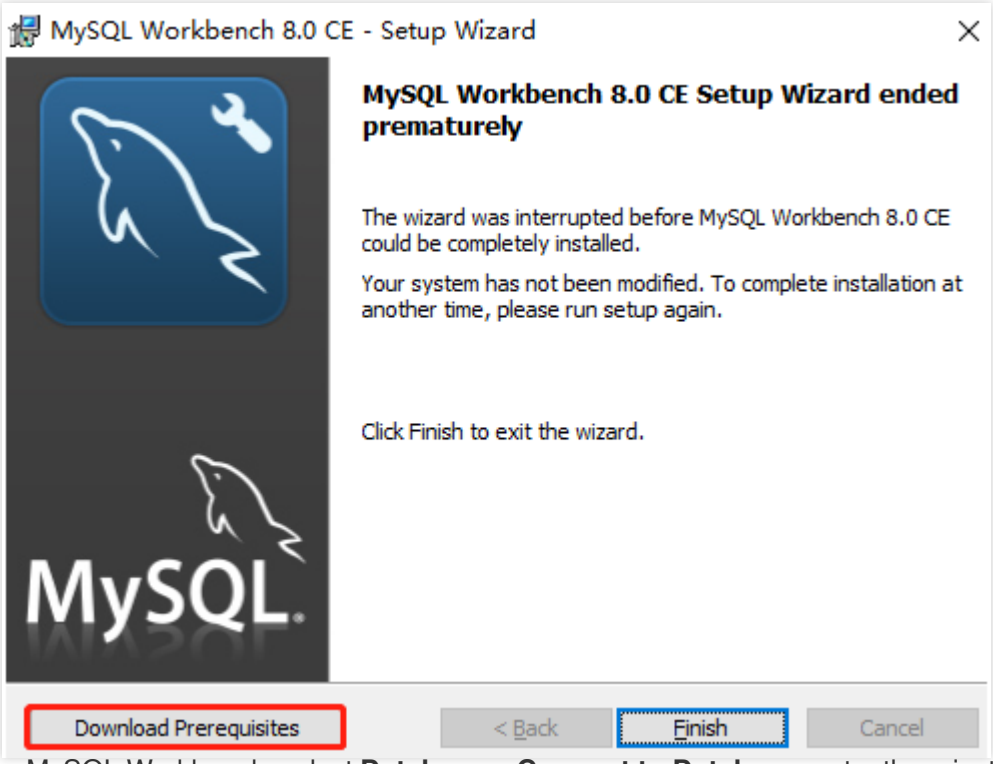
3. **Login**, **Sign Up**, and **No thanks, just start my download.** will appear on the page. Select **No thanks, just start my download.** to download quickly.



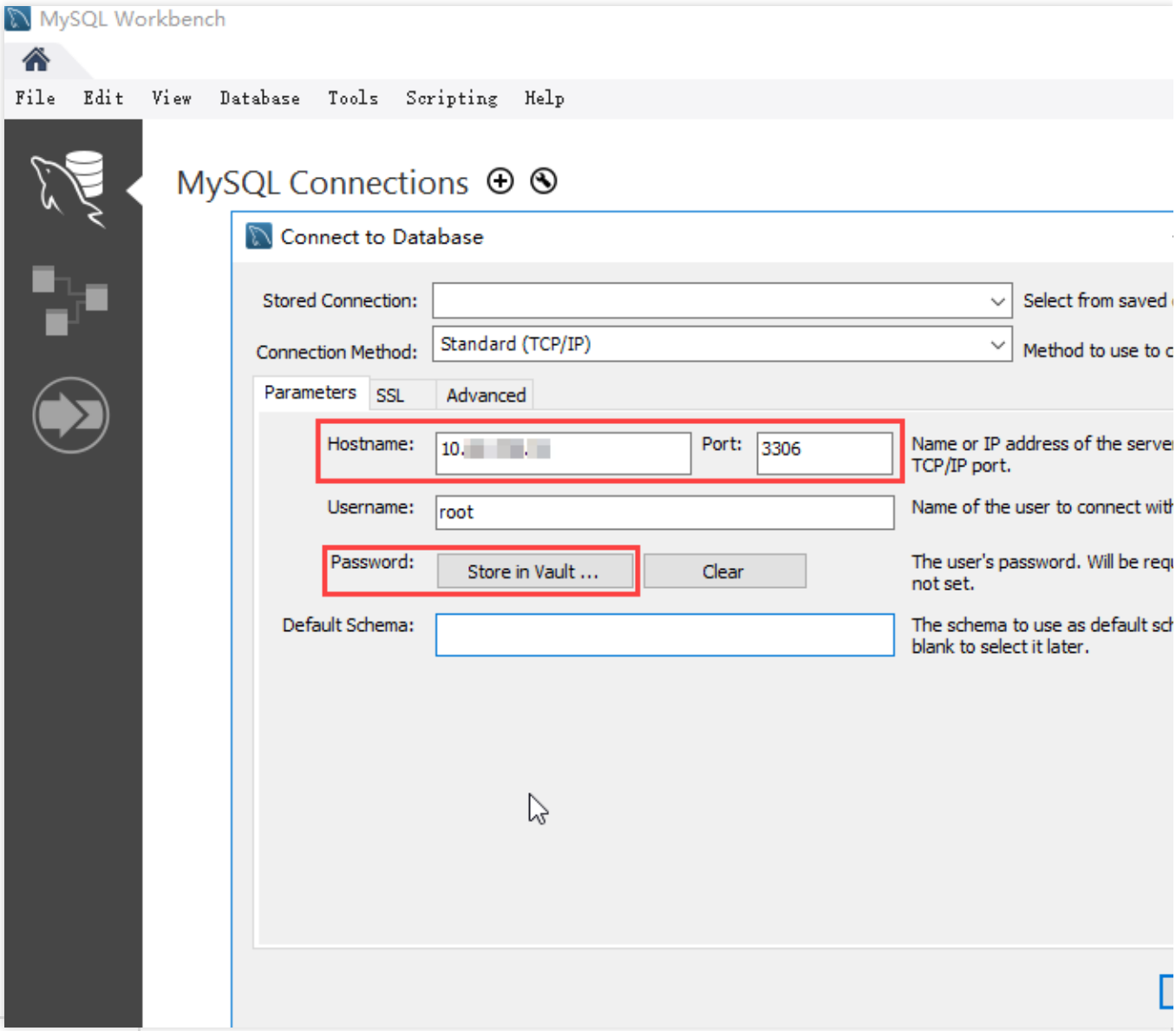
4. Install MySQL Workbench on this CVM instance.

Note

Microsoft .NET Framework 4.5 and Visual C++ Redistributable for Visual Studio 2015 are required for the installation. You can click **Download Prerequisites** in the MySQL Workbench installation wizard to enter the corresponding page to download and install them. Then, install MySQL Workbench.

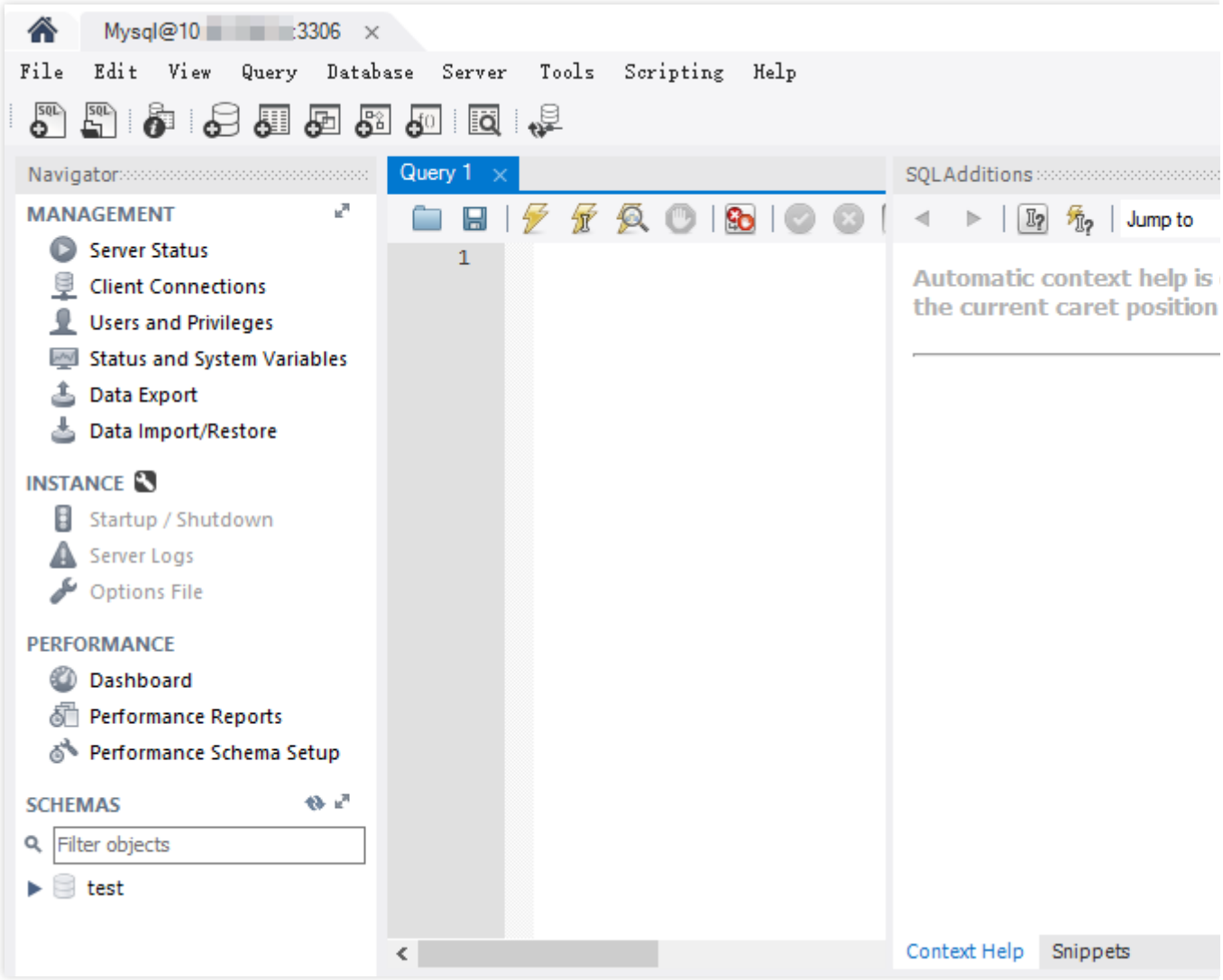


5. Open MySQL Workbench, select **Database** > **Connect to Database**, enter the private (or public) network address, username, and password of your TDSQL-C for MySQL cluster and click **OK** to log in.



Parameter	Description
Hostname	Enter the public network address of the target TDSQL-C for MySQL instance in a cluster as instructed in Step 1. For public network address, check whether it has been enabled as instructed in Enabling/Disabling Public Network Address.
Port	Private (or public) network port.
Username	Enter the account name configured when creating the database account, which is the default account `root` here for example.
Password	Enter the password corresponding to the username. If you forgot the password, reset it in the console.

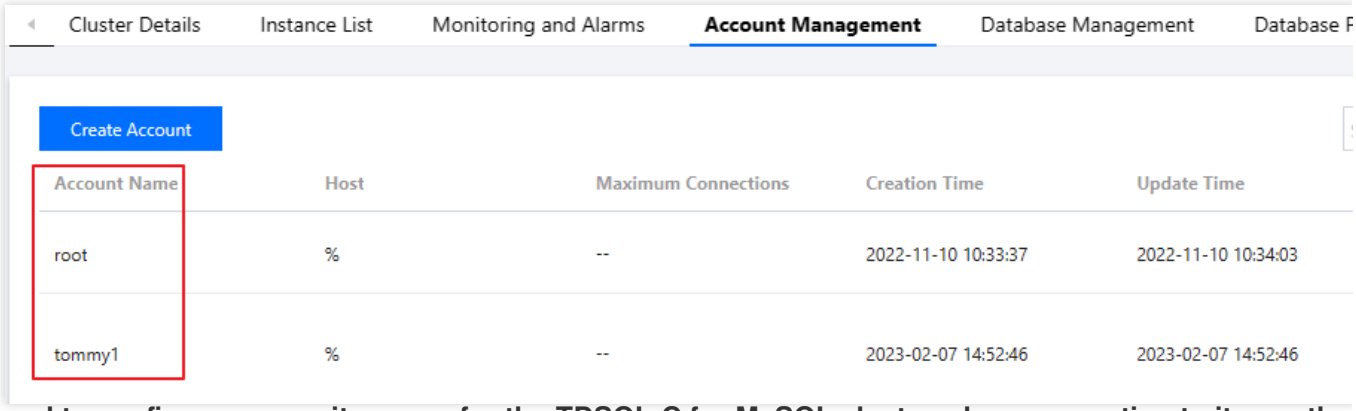
6. After successful login, a page will appear, where you can view the modes and objects of the database, create tables, and perform operations such as data insertion and query.



FAQs

How do I find the account name or reset the password to log in to the cluster?

You can view the account name you created for cluster connection under **Account Management** on the cluster management page in the TDSQL-C for MySQL console. If you forgot the password, perform the **Reset Password** operation to change the password.

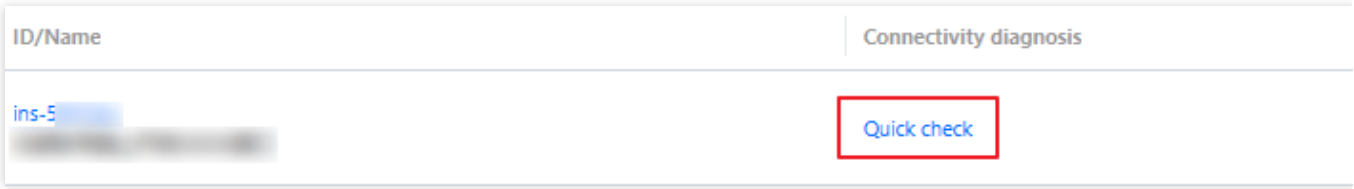


Do I need to configure a security group for the TDSQL-C for MySQL cluster when connecting to it over the private or public network?

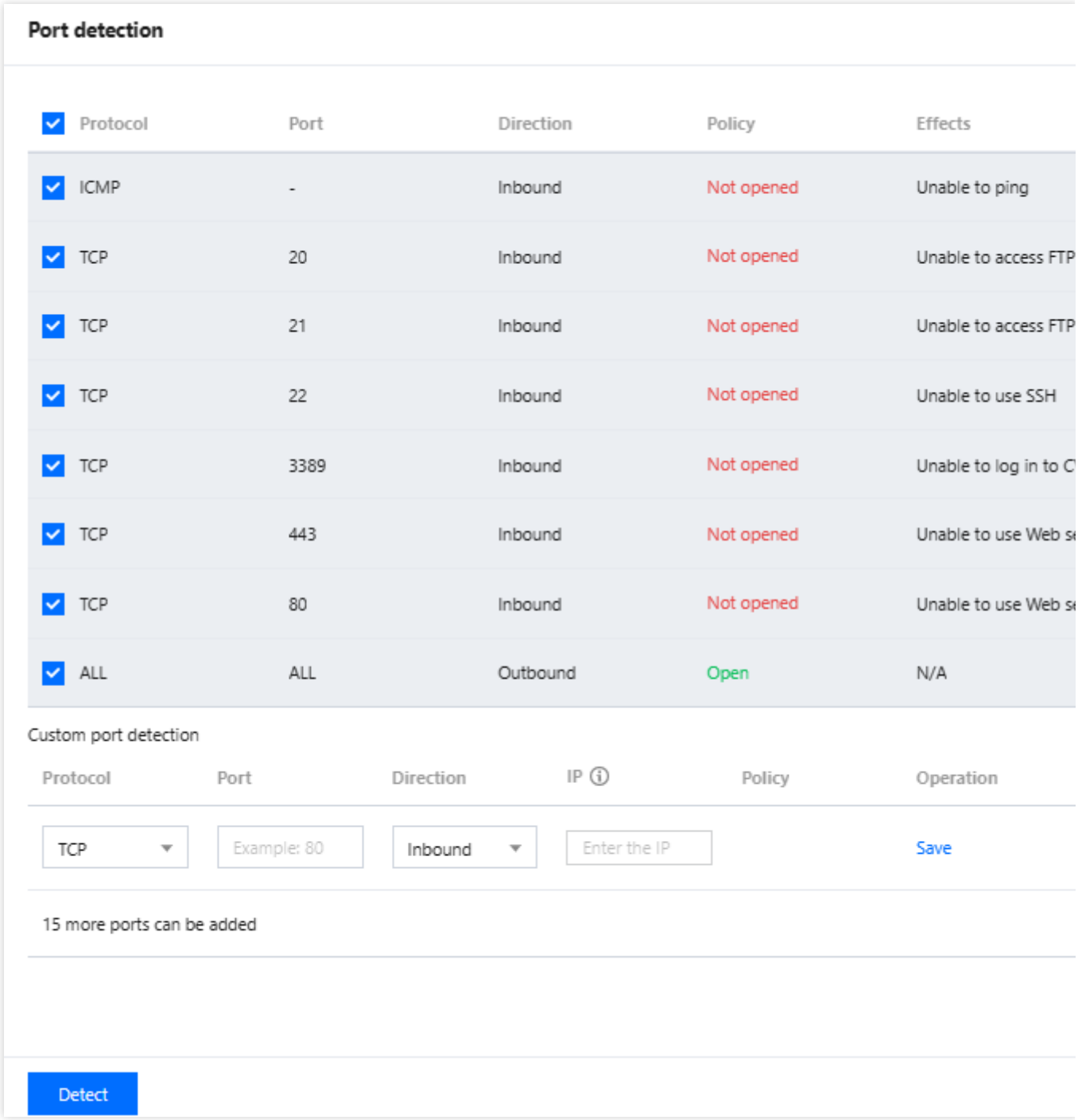
Yes. For detailed directions, see Creating and Managing TencentDB Security Groups. Note that if the public network access is enabled for connection over the public network, the private network port needs to be opened in the configured security group rule.

What should I do if I can't log in to the Linux CVM instance because the security group rules of the instance don't match properly?

If the security group rules of the Linux CVM instance don't match properly (for example, the corresponding port is not opened), and you can't log in to the instance, you can use the Security Group (Port) Verification Tool to check the connectivity of the security group.



You can locate the possible causes of the login failure through quick check.



Then, on the CVM instance details page, select **Security Group** > **Edit Rule** and open the corresponding port. For detailed directions, see Adding Security Group Rules.

What settings may cause a cluster connection failure?

Check whether the CVM instance and the TDSQL-C for MySQL cluster are in the same VPC in the same region under the same Tencent Cloud account. If any of these three prerequisites is not meet, the cluster cannot be connected.

Instance Management Overview

Last updated：2023-09-12 14:40:53

This document describes what an instance is and how to view the instance list and manipulate an instance.

Instance overview

An instance is a database resource in Tencent Cloud. A cluster is the basic management unit of TDSQL-C for MySQL, which can contain multiple instances of two types: read-write instance and read-only instance.
The types and number of instances supported by a cluster are as described below:

Instance Type	Maximum Instances	Supported IP
Read-write instance	1	The private read-write address is enabled by default, and you can enable the public read-write address manually.
Read-only instance	15	The private read-only address is enabled by default, and you can enable the public read-only address manually.

Access address

TDSQL-C for MySQL supports read-write and read-only addresses. The former corresponds to the read/write service of a TDSQL-C for MySQL read-write instance, while the latter corresponds to one or multiple read-only instances on the backend for load balancing.

Instance management overview

You can configure and manage instances in a cluster in the console, with the following configuration items:

Instance Configuration Item	Description	Operation
Instance Name	a. You can rename read-write/read-only instances in the cluster. b. The instance name can contain up to 60 letters, digits, hyphens, underscores, and dots.	Renaming Instance

Character set	a. You can modify the character set of an instance. b. The following character sets are supported: UTF-8, GBK, Latin-1, and UTF8MB4.	Modifying Character Set
Adjust Configurations	a. You can adjust the instance configuration as needed. b. You can adjust the computing specification and storage space of an instance.	Adjusting Computing Configuration Adjusting Storage Space
Maintenance Time	a. You can set the maintenance time and window of an instance to schedule maintenance operations. b. The maintenance window can be continuous or discontinuous days between Monday and Sunday. c. You can set the maintenance start time and duration.	Modifying Instance Maintenance Window
Restart Instance	a. You can restart an instance for routine maintenance or to clear the database buffer. b. To avoid affecting normal business operations, make sure that the instance has no other ongoing tasks.	Restarting Instance
Delete Instance	You can delete an instance when it is no longer used.	Deleting Instance
Restore Instance	a. You can restore an isolated instance as needed. b. The recycle bin displays the period during which the instance can be restored.	Restoring Instance

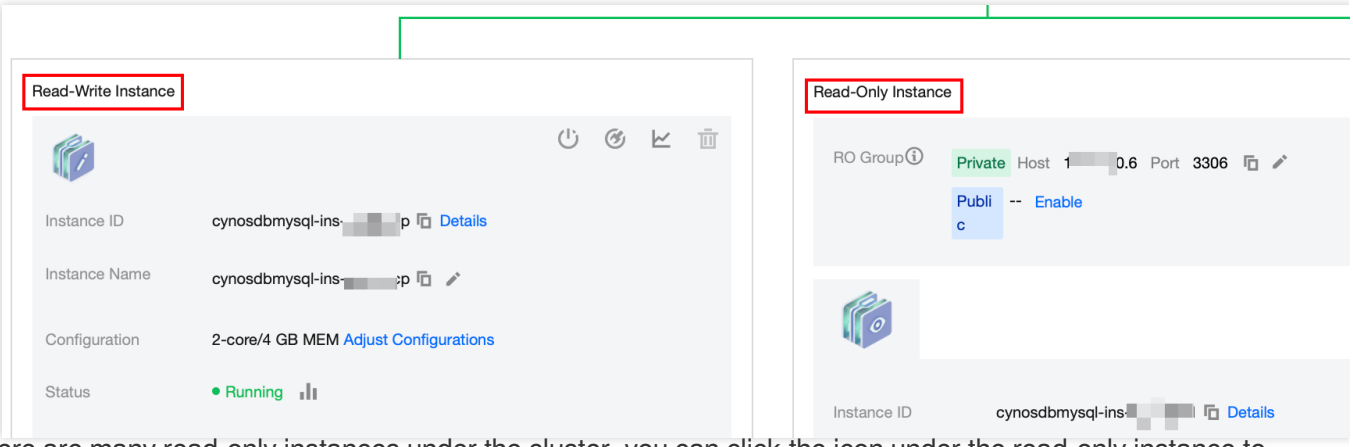
Viewing the instance list

On the cluster management page, proceed according to the actually used view mode.

Tab view

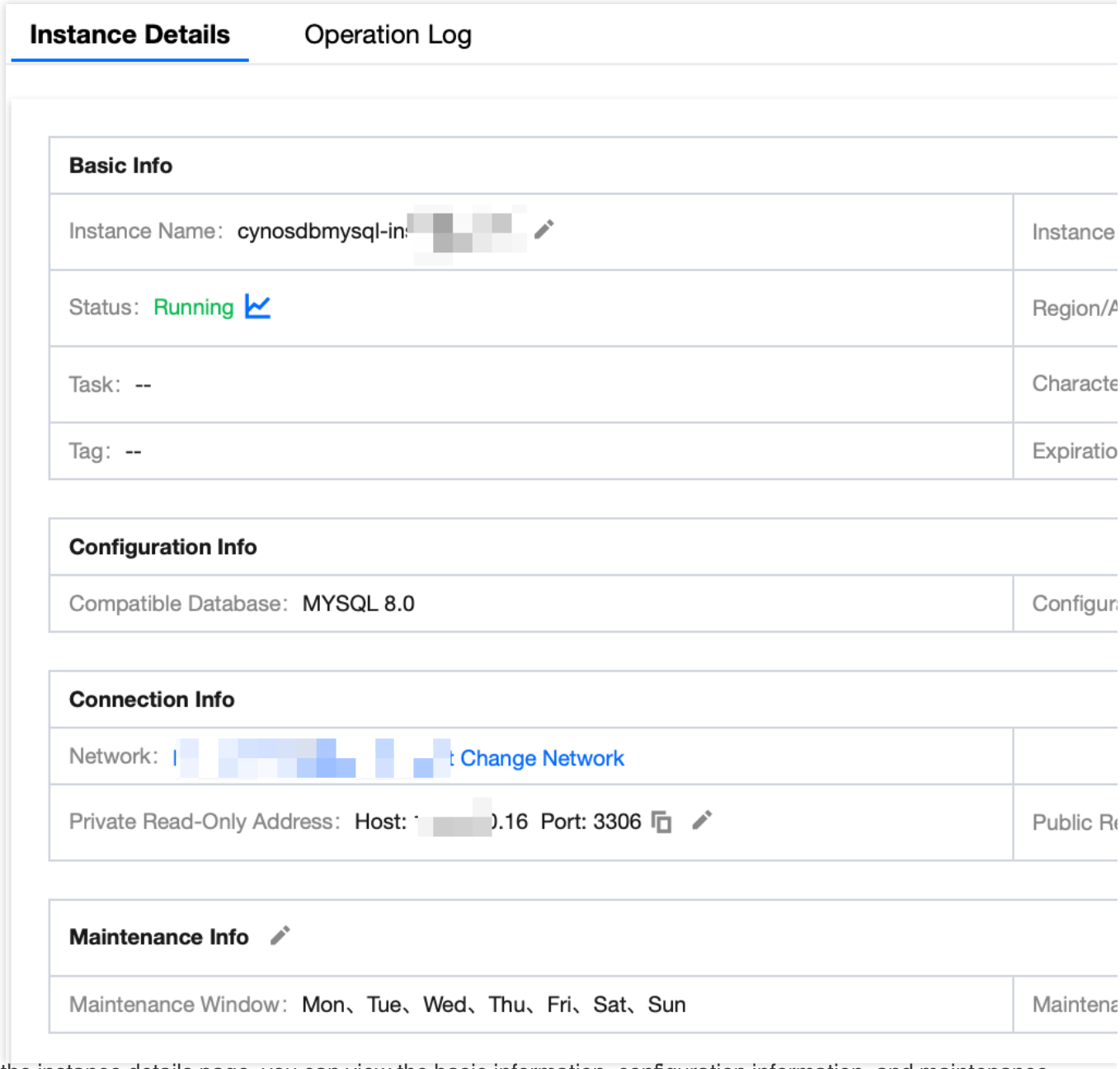
List view

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. On the **Cluster Details** page, view all the read-write instance and read-only instances in the cluster.



3. If there are many read-only instances under the cluster, you can click the icon under the read-only instance to switch to view. If you have 5 or more read-only instances, you can also click the drop-down button on the right of the icon to filter out the target instance ID.

4. You can view the information of the read-write instance and read-only instance: instance ID/name/status, and public/ private read-write/read-only address. You can also perform the following operations on the instance: restart the instance, modify the instance name, modify the private network address, enable or disable the public network address, restart the instance, adjust the configuration, or terminate the instance. Then, you can click **Details** to enter the corresponding instance details page.



5. On the instance details page, you can view the basic information, configuration information, and maintenance information of the instance, as well as the operation log of the instance.

1. Log in to the TDSQL-C for MySQL console.

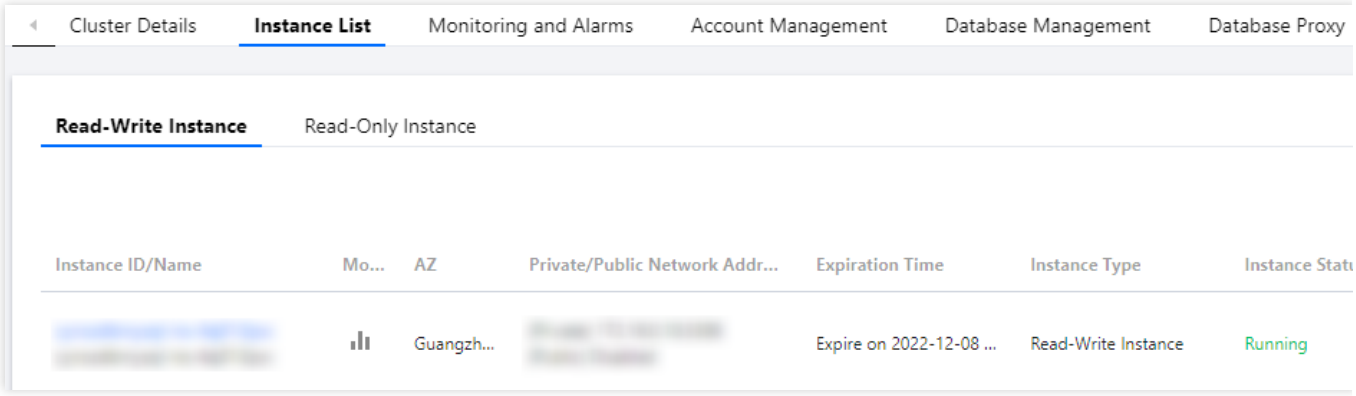
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.

3. On the cluster management page, select the **Instance List** tab to view the list of read-write and read-only instances in the cluster.

The instance list displays the **Instance ID/Name**, **Monitoring**, **AZ**, **Private/Public Network Address**, **Expiration Time**, **Instance Type**, **Instance Status**, **Instance Configuration**, and **Operation** by default.

If there are many instances in the cluster, you can also enter an instance ID or name keyword in the search box on the right to quickly find the target instance.

Click the **instance ID** to enter the instance details page, where you can query the instance information and operation logs.



Task List

Last updated：2023-03-01 14:33:46

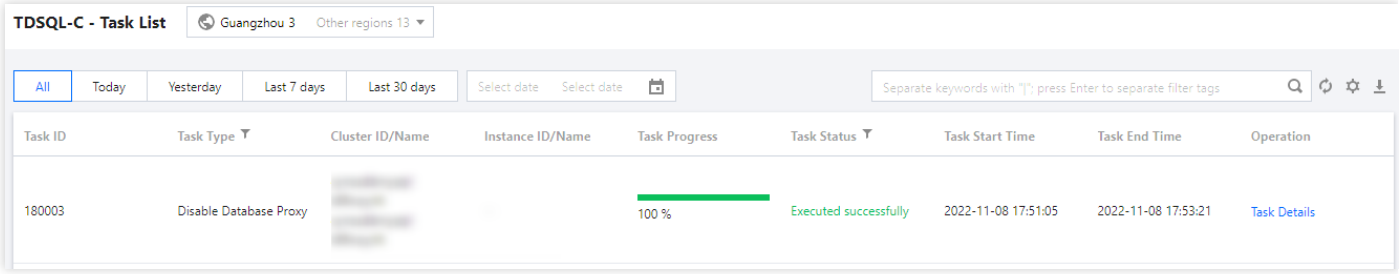
This document describes how to view the task list in the TDSQL-C for MySQL console. In this way, you can stay up to date with the progress of operations on clusters and instances.

Task type

The task list contains tasks of the following types: cluster creation, cluster rollback, instance upgrade/downgrade, parameter modification, restart, manual backup, version upgrade, replica AZ adding/modification/deletion, AZ switch, binlog settings modification, database proxy creation/upgrade/disabling/connection pool configuration/node rebalancing/kernel minor version upgrade, and read/write separation configuration.

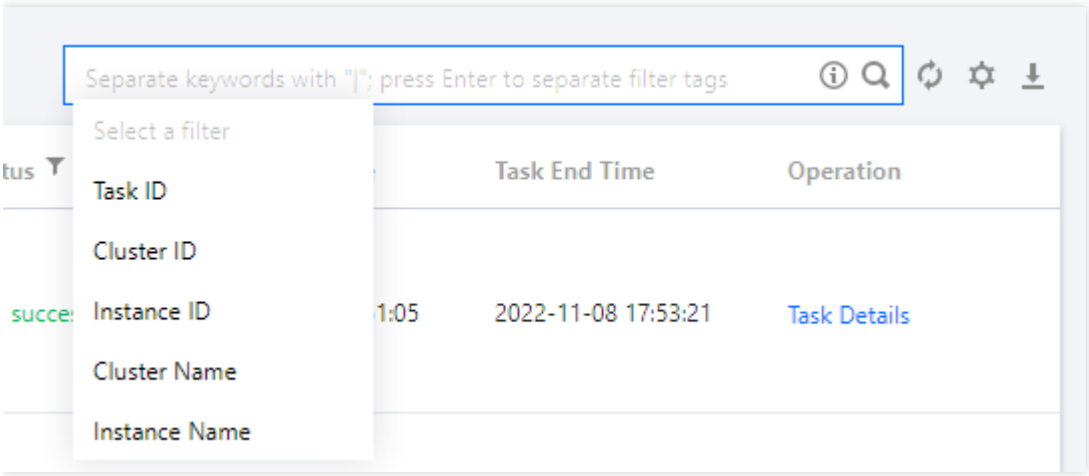
Viewing the task list

1. Log in to the TDSQL-C for MySQL console.
2. On the left sidebar, click **Task List**.
3. Select the region at the top.
4. Directly find or search for the target task by keyword to view its details.



Searching by keyword

In the task list, you can search for the target task by task ID, cluster ID/name, and instance ID/name. Separate multiple keywords by vertical bar "|" and separate filter tags by carriage return.



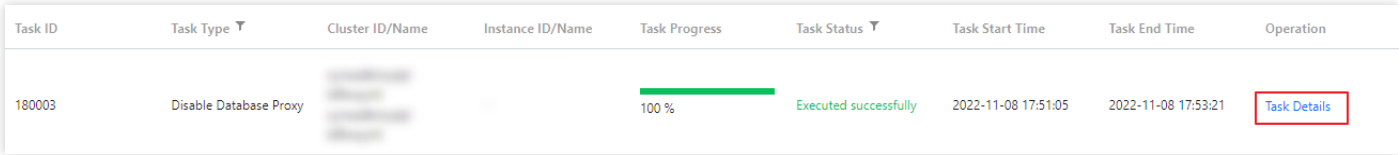
Downloading the task data

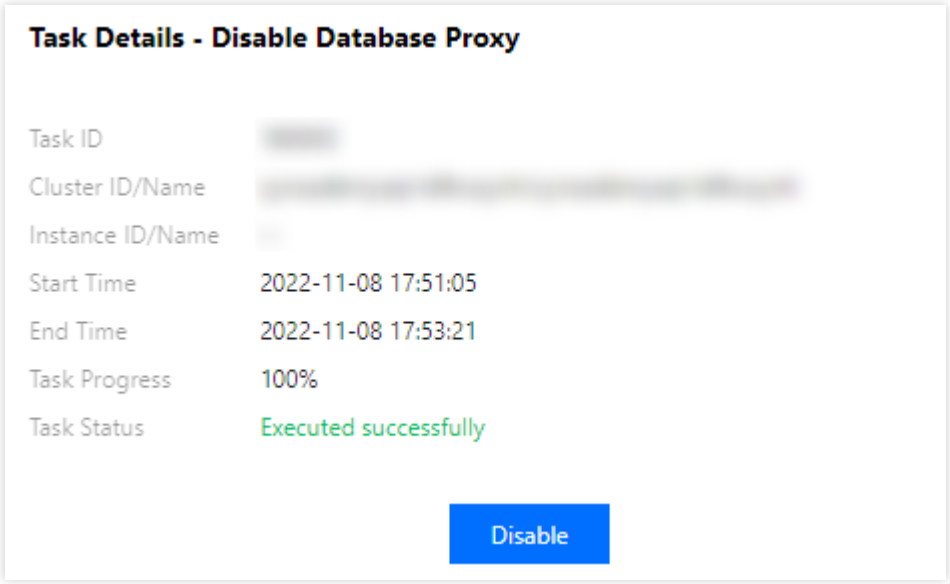
1. Log in to the TDSQL-C for MySQL console.
2. On the left sidebar, click **Task List**.
3. Select the region at the top.

4. Click the icon next to the search box to download the data on the current page or under the current search criteria.

Viewing task details

1. Log in to the TDSQL-C for MySQL console.
2. On the left sidebar, click **Task List**.
3. Select the region at the top.
4. Find the target task and click **Task Details** in the **Operation** column.





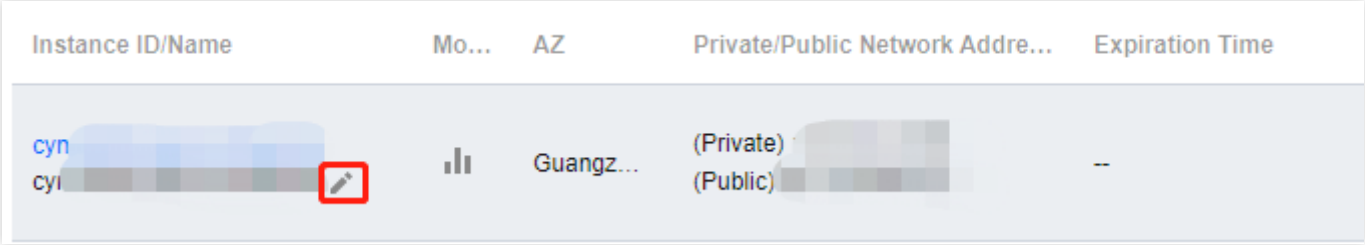
Renaming Instance

Last updated：2022-12-01 11:21:44

TDSQL-C for MySQL read-write and read-only instances can be identified and managed by their names, which are the same as instance IDs by default and can be changed in the console. The instance IDs cannot be changed though.

Renaming an instance in the instance list

1. Log in to the console and click the ID of the target cluster in the cluster list to enter the cluster management page.
2. On the instance management page, select the **Instance List** tab and click the modification icon after the name of the target instance.



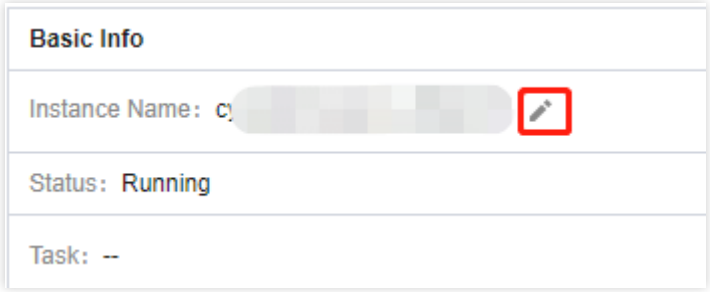
3. In the pop-up window, enter a new name and click **OK**.

> Note：
> The instance name can contain up to 60 letters, digits, hyphens, underscores, and dots.

Renaming an instance on the instance details page

1. Log in to the console and click the ID of the target cluster in the cluster list to enter the cluster management page.
2. On the instance management page, select the **Instance List** tab and click the ID of the target instance to enter the instance details page.

3. In the **Basic Info** section of the instance details page, click the modification icon after the instance name.



4. In the pop-up window, enter a new name and click **OK**.

> Note：
> The instance name can contain up to 60 letters, digits, hyphens, underscores, and dots.

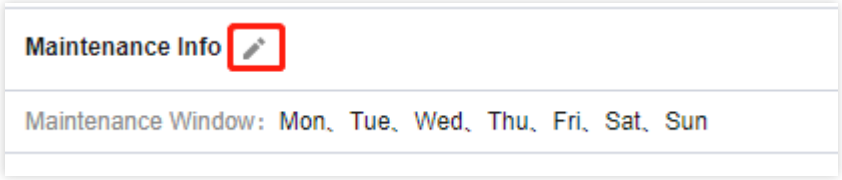
Modifying Instance Maintenance Window

Last updated：2022-04-02 00:01:27

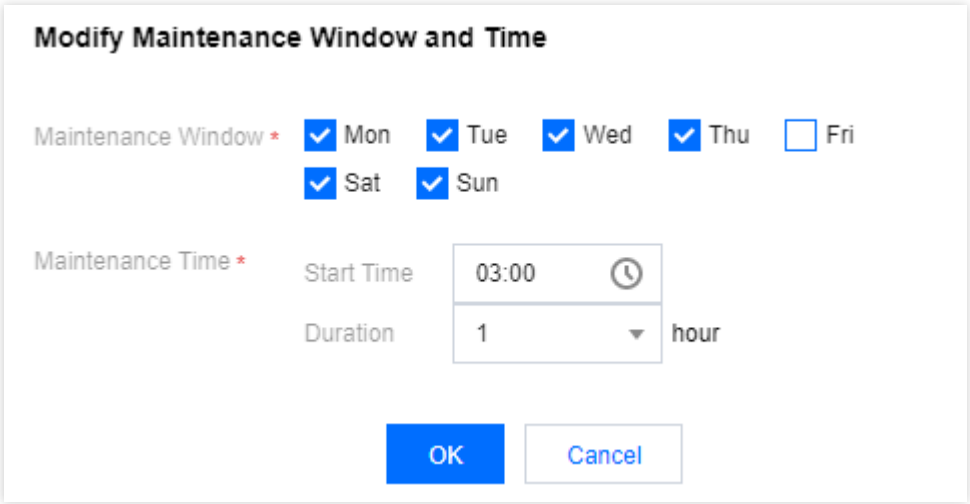
This document describes how to modify the instance maintenance time. If you want to perform configuration adjustment or other operations that will cause a momentary disconnection or affect the instance, you can specify the instance maintenance time.

Directions

1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list to enter the cluster management page.
2. On the instance management page, select the **Instance List** tab and click the target instance ID to enter the instance details page.
3. In the **Maintenance Info** section on the instance details page, click the modification icon.



4. In the pop-up window, modify the maintenance window and time and click **OK**.
   - **Maintenance Window** indicates a day of the week as the maintenance window. Each week must has a maintenance window.
   - **Maintenance Time** indicates the maintenance time on a day. You can specify any time and set the maintenance duration to 1, 1.5, 2, or 2.5 hours.



Modifying Character Set

Last updated：2023-11-01 17:37:18

A character set is a mapping relationship or encoding rule, including a coded character set and character encoding. The code points corresponding to a character set are mapped into binary sequences, so that they can be stored and processed by a computer.

This document describes how to change the character set of an instance in the console.

Note

Modifying the character set of an instance may cause the instance to restart. So we recommend that you modify it during the instance maintenance time.

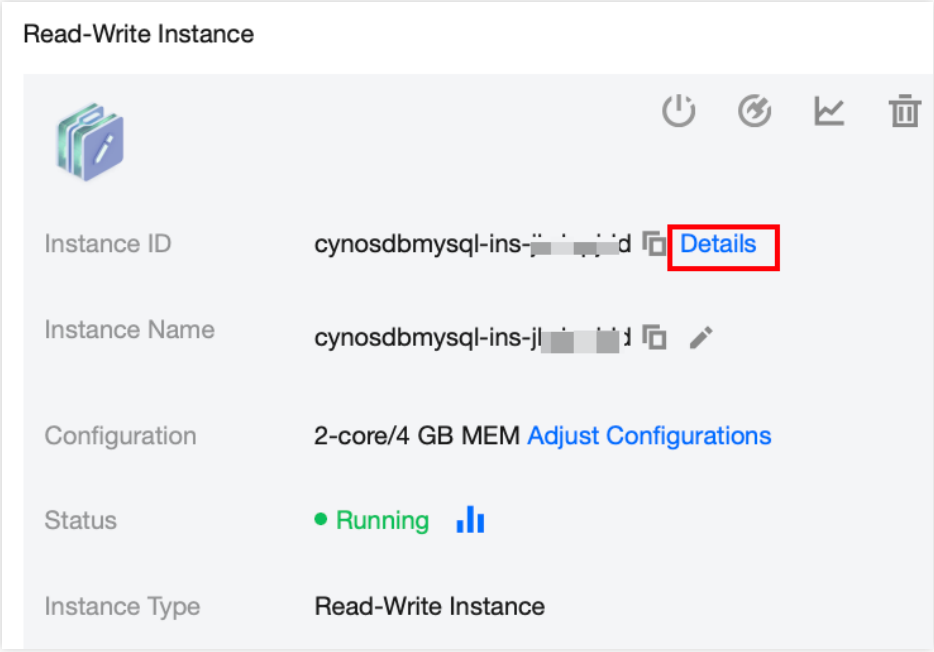
Directions

On the cluster list page, proceed based on the actually used view mode.

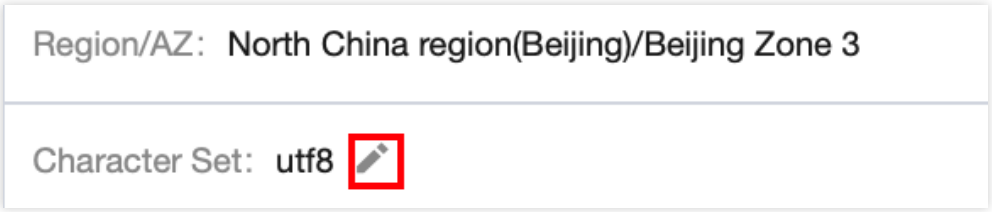
Tab view

List view

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. On the **Cluster Details** page, find the target instance, and click **Details** next to the instance ID to enter the instance details page.



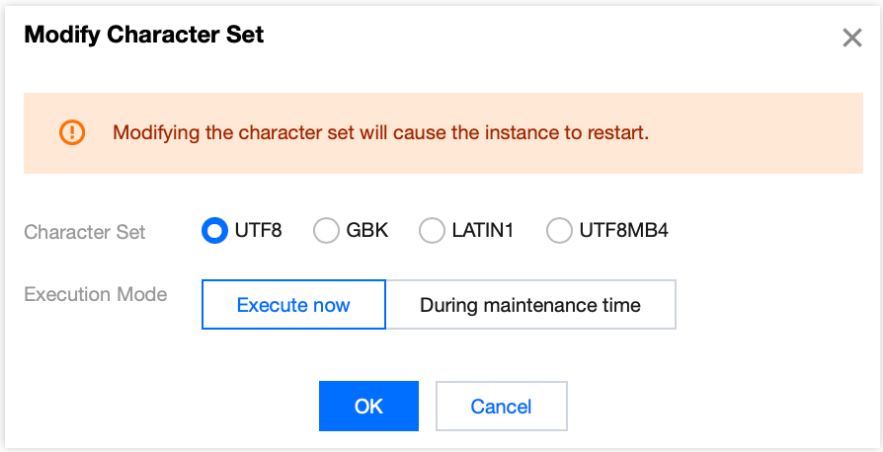
3. On the instance details page, click the modification icon next to **Character Set**.



4. In the pop-up window, select a character set and click **OK**.

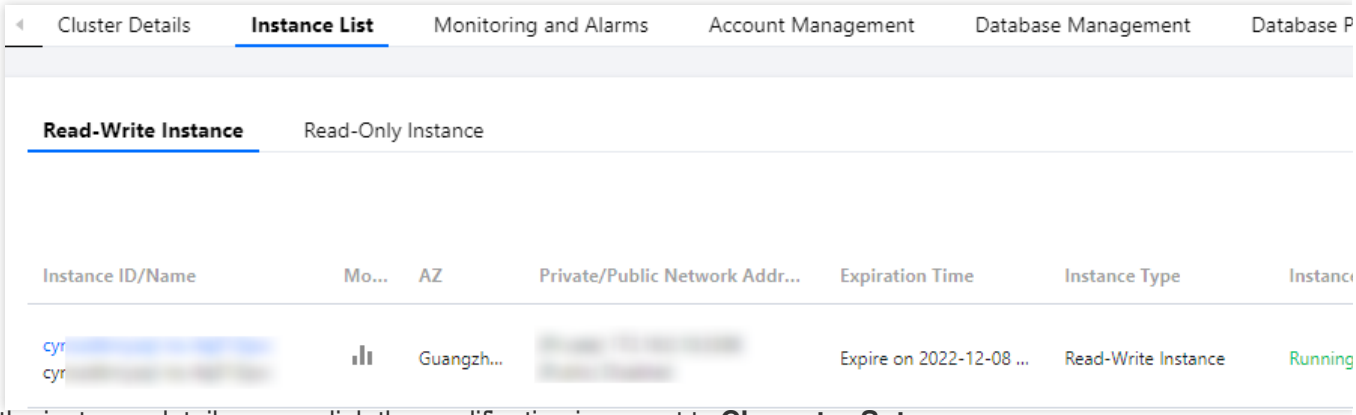
Note

The supported character sets include UTF8, GBK, LATIN1, and UTF8MB4. For more information, see MySQL's official documentation.



1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list to enter the cluster management page.

2. On the instance management page, select the **Instance List** tab and click the ID of the target read-write or read-only instance to enter the instance details page.

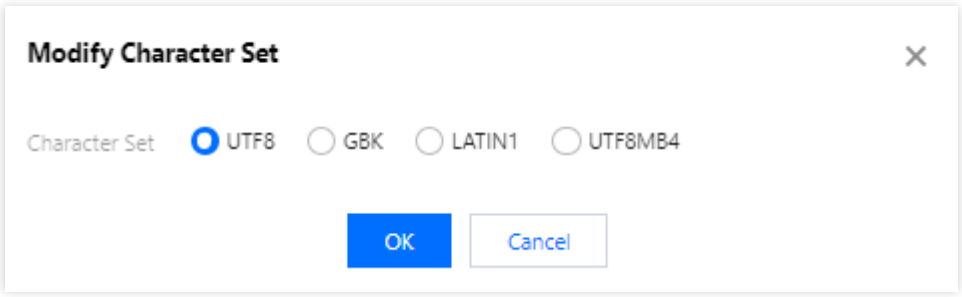


3. On the instance details page, click the modification icon next to **Character Set**.

4. In the pop-up window, select a character set and click **OK**.

Note

The supported character sets include UTF8, GBK, LATIN1, and UTF8MB4. For more information, see MySQL's official documentation.



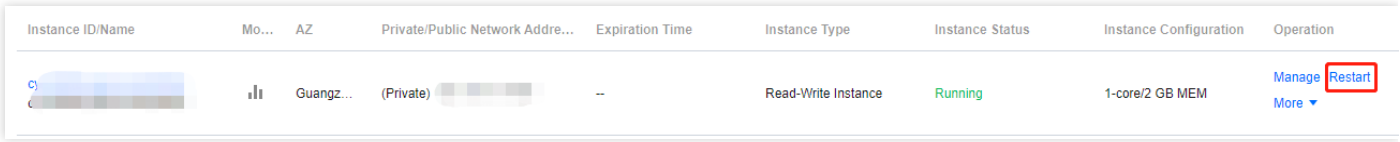
Restarting Instance

Last updated：2022-12-01 11:21:44

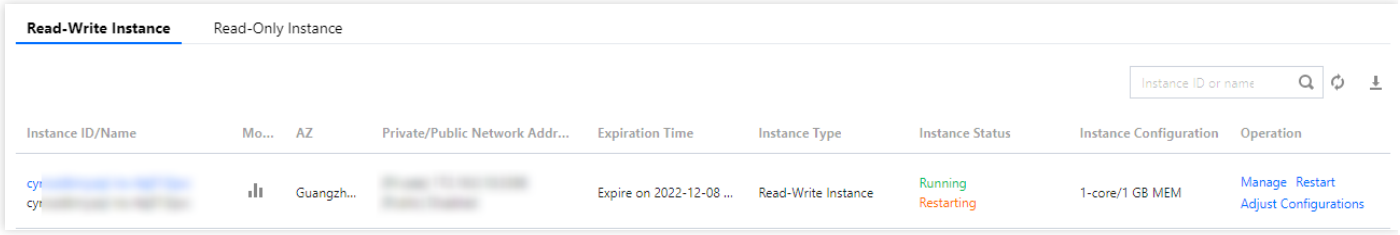
You can manually restart an instance if you want to maintain it routinely, clear the database buffer, or if a performance problem occurs.

Directions

1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list or **Manage** in the **Operation** column to enter the cluster management page.
2. On the cluster management page, select the **Instance List** tab, find the target instance, and click **Restart**.



3. In the pop-up window, confirm that the information of the instance to be restarted is correct and click **OK**.
4. Once the instance status changes from **Restarting** to **Running**, the restart is completed.



Deleting Instance

Last updated：2024-06-17 16:12:03

This document describes how to delete an instance from a cluster when it is no longer used.

Note:

After a read-write instance is deleted, its data and backup files will also be deleted and cannot be recovered in the cloud. Store your backup files safely elsewhere in advance.

After an instance is deleted, its IP resources will also be released. Therefore, confirm that your business no longer needs to access the instance before deleting it.

If you delete a cluster, all instances in the cluster will be deleted.

Data is deleted only when the read-write instance is deleted. Deleting read-only instances only deletes computing resources.

Use Cases

When the cluster has pre-configured resources as its instance type, deleting read-only instances under the cluster is supported. The read-write instance can only be deleted after all read-only instances under the cluster have been removed. After the read-write instance is deleted, the cluster is also deleted.

When the cluster has Serverless as its instance type, and the architecture is set to cluster mode, it is not possible to delete read-write instances or read-only instances individually. To reduce the number of read-only instances, one must refer to the Read-Only Instance Management guidelines. Conversely, when the architecture is set to a single-node mode, deleting the read-write instance equates to the deletion of the entire cluster.

Lifecycle

After a monthly-subscribed instance is deleted, it will be moved to the recycle bin and retained there for seven days. During the retention period, the instance cannot be accessed, but it can be restored after renewal.

After a pay-as-you-go/serverless instance is deleted, it is moved to the recycle bin and retained there for three days. During the retention period, the instance cannot be accessed. If you want to restore the instance, you can renew the instance in the recycle bin.

Note：

For Serverless clusters, the scenario of restoration from the recycle bin is exclusively applicable to read-write instances. Read-only instances that have been reduced through the Read-Only Instance Management operation will not be isolated in the recycle bin but will be directly eliminated.

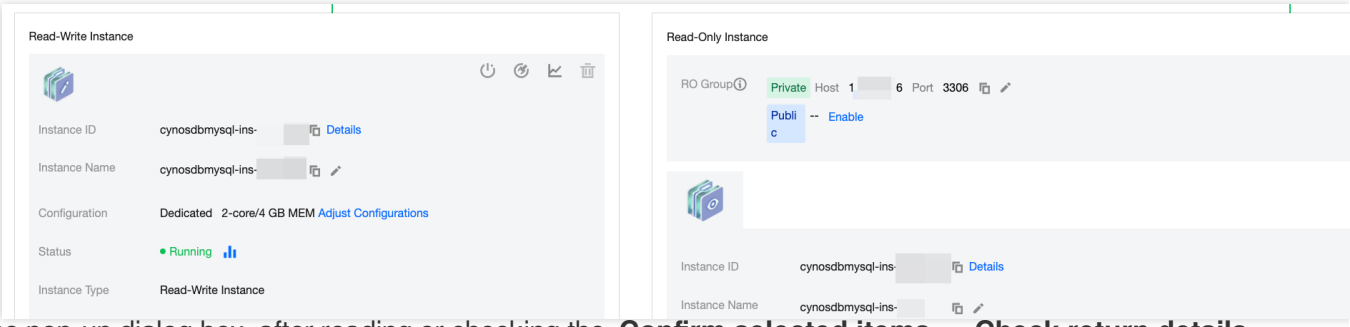
Directions

On the cluster list page, proceed based on the actually used view mode:

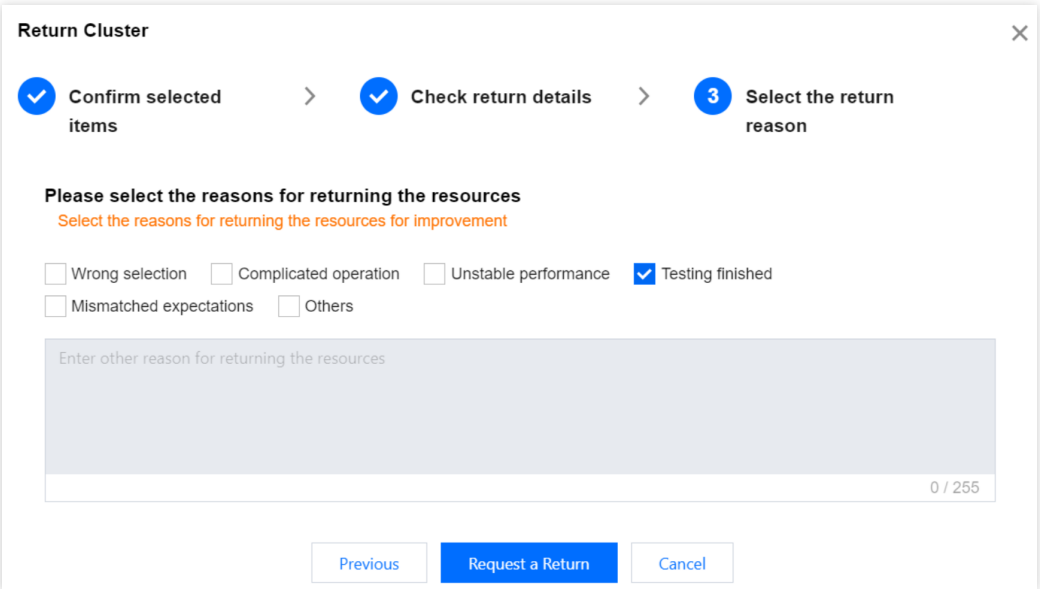
Tab View

List View

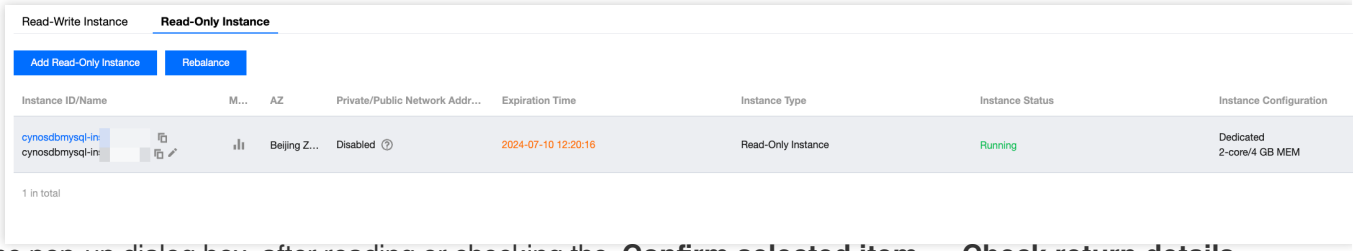
1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. Under **Cluster Details**, find the target instance and click the deletion icon.



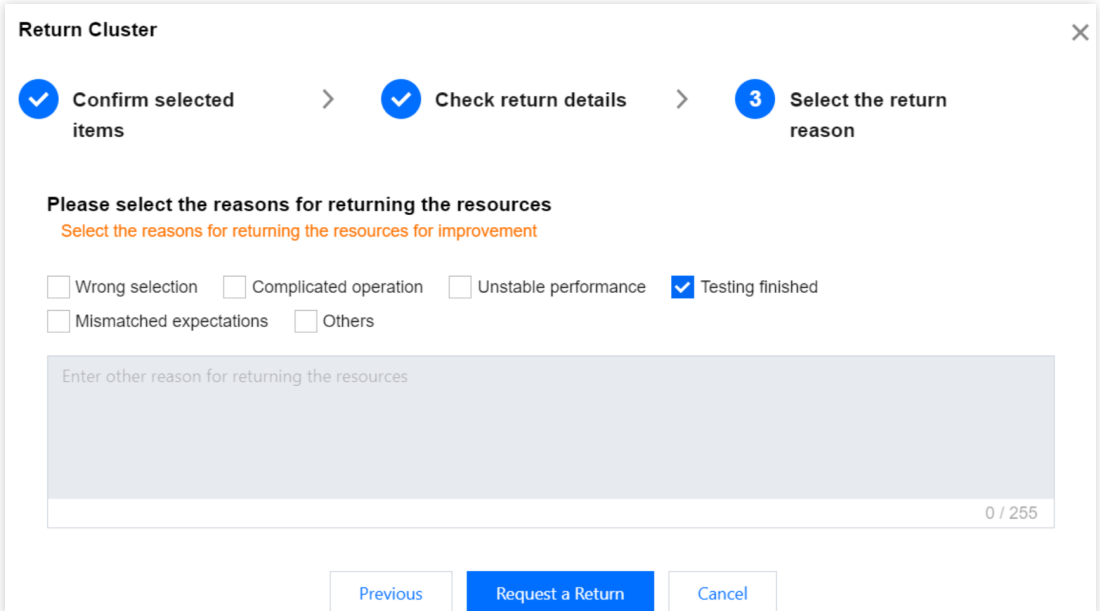
3. In the pop-up dialog box, after reading or checking the **Confirm selected items** > **Check return details** > **Select the return reason** , click **Request a Return** .



1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list or **Manage** in the **Operation** column to enter the cluster management page.
2. On the cluster management page, select the **Instance List** tab, find the target read-write or read-only instance, and select **More** > **Return/Refund** in the **Operation** column.



3. In the pop-up dialog box, after reading or checking the **Confirm selected item** > **Check return details** > **Select the return reason** , click **Request a Return** .



Restoring Instance

Last updated：2024-06-17 10:45:57

This document describes how to restore an isolated instance in the console.

Overview

If an instance is deleted by mistake, due to overdue payment, or upon expiration, you can restore it from the recycle bin before it is eliminated.

Note:

After an instance is restored, it uses the same configurations as before.

An instance cannot be terminated, restored and terminated again in a short time.

For the Serverless clusters, the scenario of restoration from the recycle bin is exclusively applicable to read-write instances. Read-only instances that have been reduced through the Read-Only Instance Management operation will not be isolated in the recycle bin but will be directly eliminated.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select **Recycle Bin** on the left sidebar, select the region, find the target instance, and click **Restore** in the **Operation** column.

Note:

Read-only instances in a cluster can be restored only after the read-write instance in the cluster is restored.

You can only restore an instance before it is terminated.

If you click **Release Now** in the **Operation** column, the instance will be eliminated immediately and cannot be restored. Therefore, proceed with caution.

3. In the pop-up window, select a renewal period, confirm the instance information, and click **OK**.
4. After the restoration is completed, the instance becomes **Running**, and you can see it in the instance list.

Configuration Adjustment
Adjusting Compute Configuration

Last updated：2024-06-17 15:34:02

If your instance is overprovisioned or underprovisioned, your business needs cannot be best met, and you can adjust its specifications to fully utilize resources and reduce unnecessary costs.
This document introduces how to adjust the compute configuration for clusters with pre-configured resource instance types. For information on adjusting configurations for serverless clusters, refer to Adjusting Configurations.
Note:
If the cluster instance type is a pre-configured resource and a database proxy is enabled, you can choose whether to adjust the configuration of the database proxy on the compute configuration adjustment page.

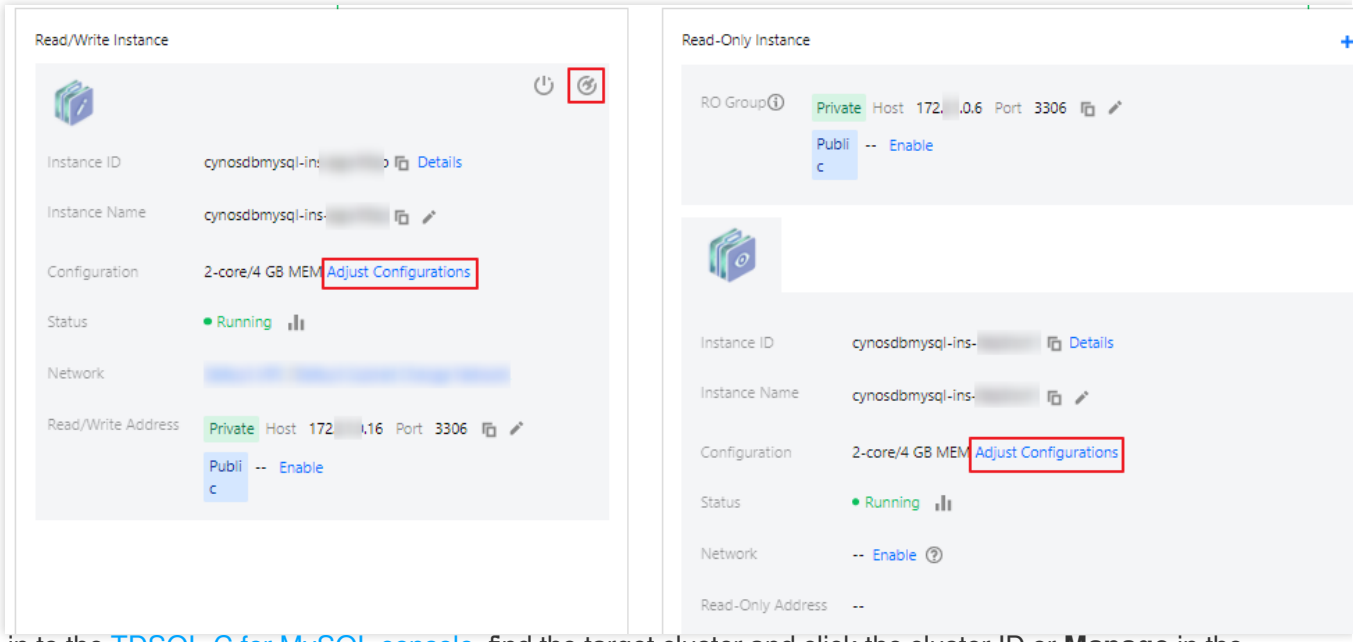
Configuration Adjustment Capabilities

TDSQL-C for MySQL uses an architecture where computing and storage resources are separated and all compute nodes share the same data. In cross-server configuration adjustment, no data migration is required; therefore, configuration upgrade/downgrade can be completed within seconds.
When an instance undergoes configuration adjustment, a new primary instance of higher specifications will be automatically started. Once set up, the new primary instance is automatically promoted to the primary database, and the disconnection of the old primary database completes the configuration adjustment.

Directions

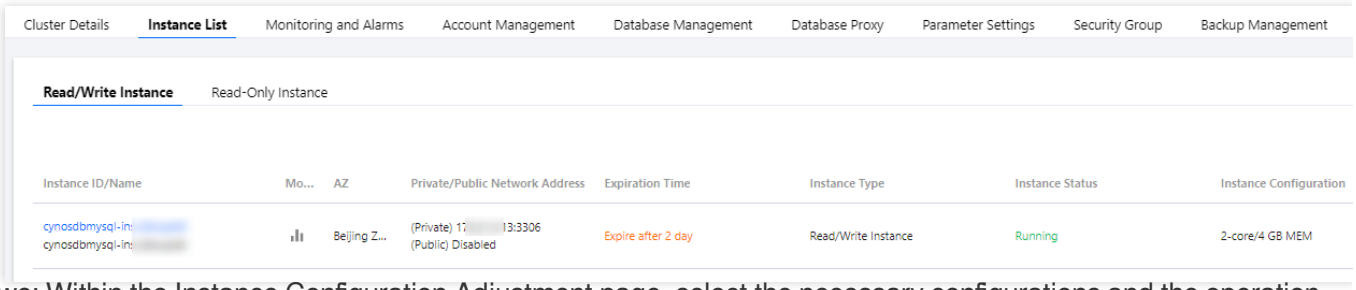
Step one: On the cluster list page, operate based on the actually used view mode, and navigate to the configuration adjustment page:
Tab View
List View
1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. On the **Cluster Details** tab, find the target instance, and click the icon

or click **Adjust Configurations** to enter the configuration adjustment page.



1. Log in to the TDSQL-C for MySQL console, find the target cluster and click the cluster ID or **Manage** in the **Operation** column in the cluster list to enter the cluster management page.

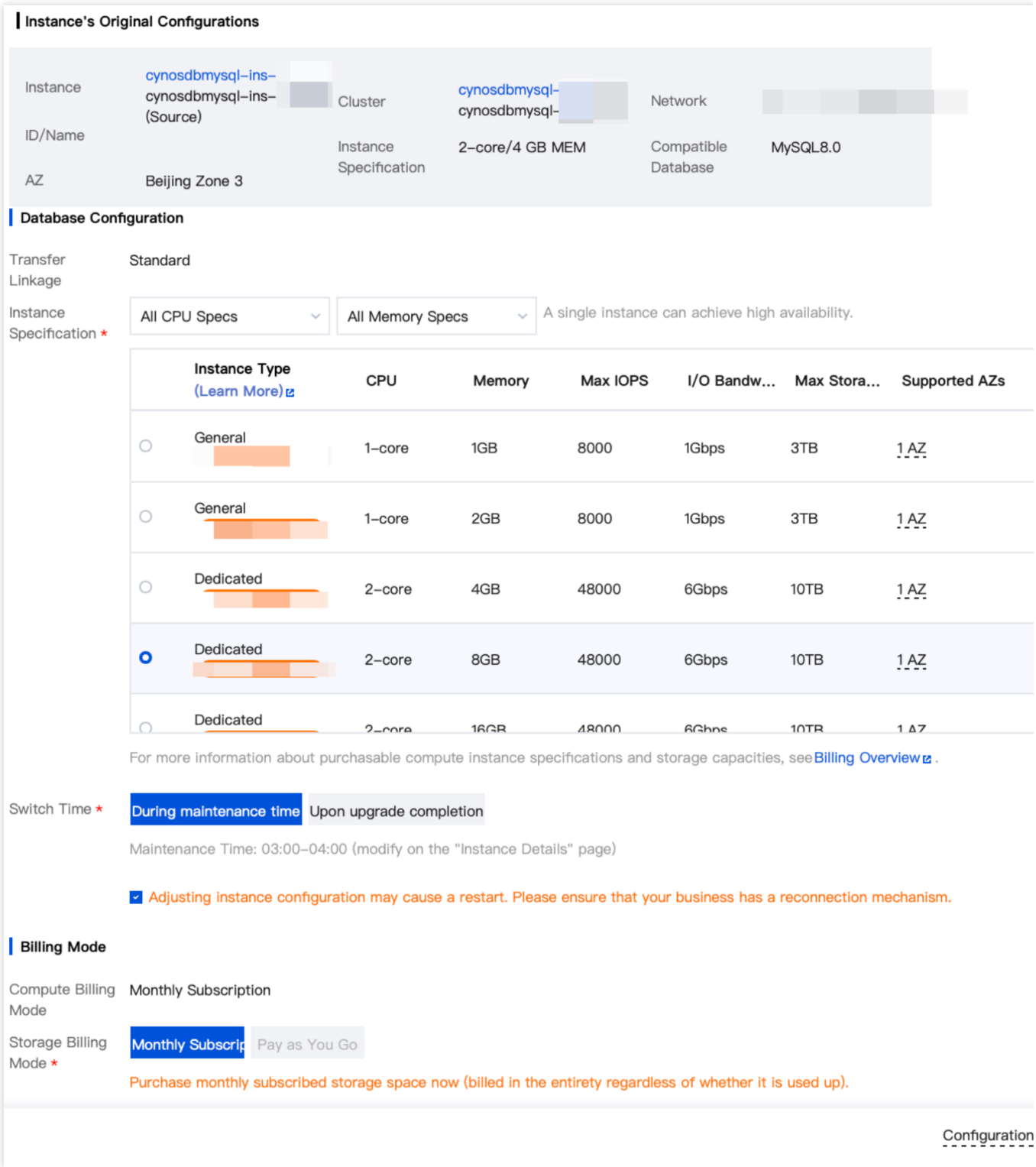
2. On the **Instance List** tab, locate the desired instance and select **More** > **Adjust Configurations** in the **Operation** column.



Step two: Within the Instance Configuration Adjustment page, select the necessary configurations and the operation time, and click **Buy Now**.

Note：

Adjustment to instance configurations may result in momentary disconnections. Please ensure your operations have a reconnection mechanism in place.



Instance Specification: Select the CPU and memory you need to adjust.

Configure Database Proxy: Choose whether to adjust the configuration of the database proxy nodes. For information on how the adjustment of each configuration on the database proxy nodes affects, refer to Adjusting Database Proxy Configuration. This option is only displayed when the instance form of the cluster is pre-configured resources and the database proxy is enabled.

Switch Time

Upon selecting **Upon Upgrade Completion**, the switch-over action will be immediately triggered following the completion of configuration changes.
Select **During Maintenance Time**, and the instance will undergo a flash cut switch within the **Maintenance Window**. For modifying the maintenance window, refer to Modify Instance Maintenance Time.

Adjusting Storage Space

Last updated：2024-06-17 15:46:41

If the cluster storage space cannot meet your business needs, you can adjust it as needed.
This document introduces how to adjust the storage space for clusters with pre-configured resources as the instance type. Clusters with a serverless instance are charged for storage space on a pay-as-you-go basis, without needing to pre-purchase storage space. However, there is a storage space limit corresponding to the compute unit configuration. For details, refer to Compute Unit. If you need to adjust the compute unit configuration to obtain a higher storage space limit, refer to Modifying Cluster Compute Unit Configuration.
Note:
For clusters with pre-configured resources as the instance type, if both the compute and storage resource billing modes are pay-as-you-go, there is no need to expand the storage space. The maximum available storage capacity is that of the cluster's read-write instance corresponding to the compute specifications. To use storage space beyond this limit, you can upgrade the compute specifications of the read-write instance. Furthermore, you can convert pay-as-you-go to monthly subscription to adjust the storage space after changing the cluster billing mode to monthly subscription. However, note that billing modes temporarily do not support switching back from monthly subscription to pay-as-you-go. For more information, refer to Product Specifications.
For clusters with pre-configured resources as the instance type, if the compute billing mode is monthly subscription and the storage billing mode is pay-as-you-go, since the storage billing mode supports switching from pay-as-you-go to monthly subscription, you can adjust the storage space at the same time as you make the switch.
If the storage billing mode is monthly subscription, you can adjust the cluster storage space in the console. To use a storage space larger than the maximum capacity of the current compute specifications, upgrade the compute specifications of the read-write instance. For more information, refer to Product Specifications.
If the storage space is monthly subscribed, the new storage space will be charged from the time it is adjusted until the cluster expires.

Directions

Note:
The prepaid storage specifications after adjustment must exceed the used storage capacity.
Changing the storage billing mode will not affect existing operations.
Scenario 1: The storage billing mode is pay-as-you-go, and you need to use a storage space larger than the maximum storage space of the current compute specifications
If the storage space is pay-as-you-go, you don’t need to expand it. The maximum storage space you can use is the same as that of the computing specification for the read-write instance. To use more storage space, you can upgrade

the compute specifications for the read-write instance. For directions, see Adjusting Compute Configuration. For more information on compute node specifications and corresponding maximum storage spaces, see Product Specifications.

Scenario 2: The storage billing mode is pay-as-you-go, and you need to switch it to monthly subscription and specify the storage space

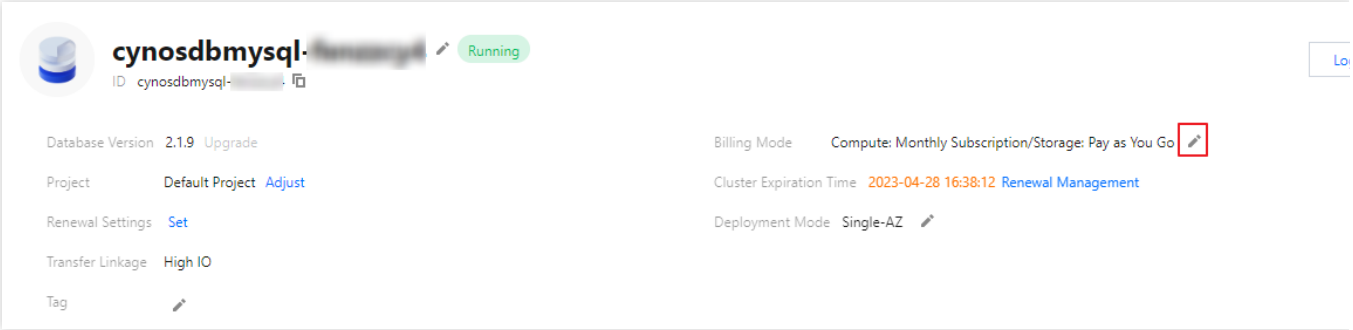
On the cluster list page, proceed based on the actually used view mode:

Tab view

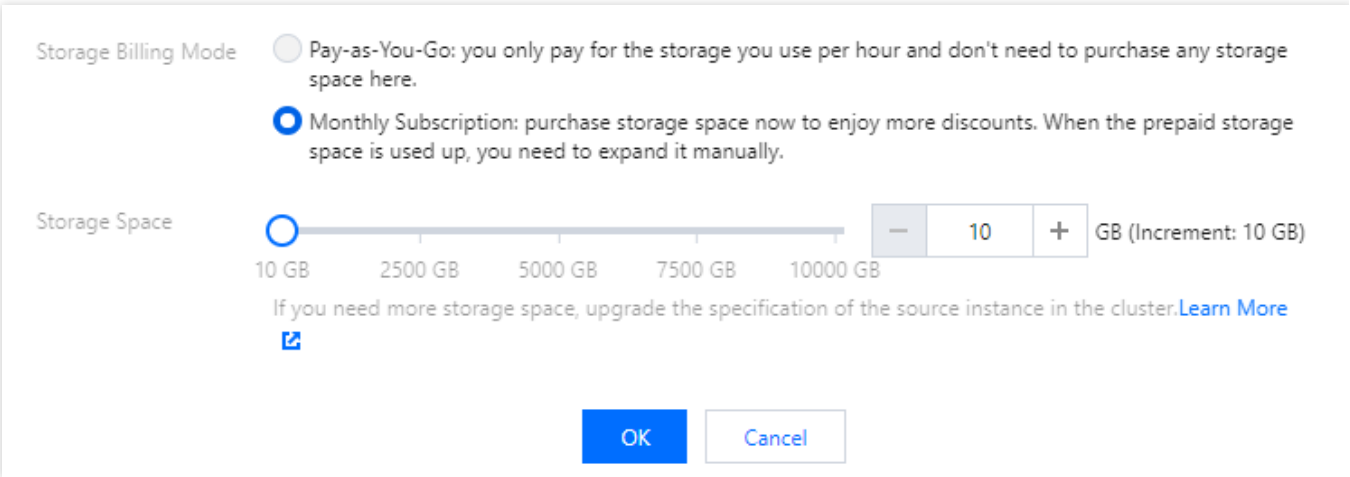
List view

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. Select the billing mode on the cluster management page and click

.



3. In the setting window that pops up, select **Monthly Subscription** for the billing mode, specify the storage space size, and click **OK** to make the payment.



1. Log in to the TDSQL-C for MySQL console, and click ID of the target cluster in the cluster list or **Manage** in the **Operation** column to enter the cluster management page.
2. Select **Cluster Details** > **Configuration Info** > **Database Storage (Used/Total)** and click

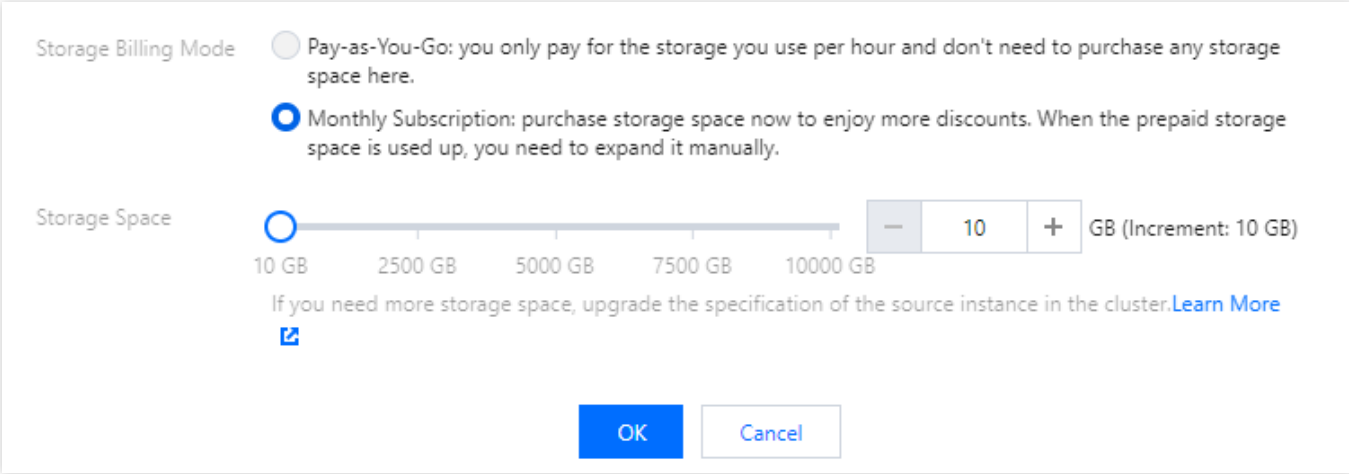
.

Note:

You can also click **Billing Info** > **Storage Billing** on the cluster details page, and click

for adjustment.

3. In the pop-up window, select **Monthly Subscription** for the billing mode, specify the storage space size, and click **OK** to make the payment.



Scenario 3: The storage billing mode is monthly subscription, and you need to adjust the storage space

On the cluster list page, proceed based on the actually used view mode:

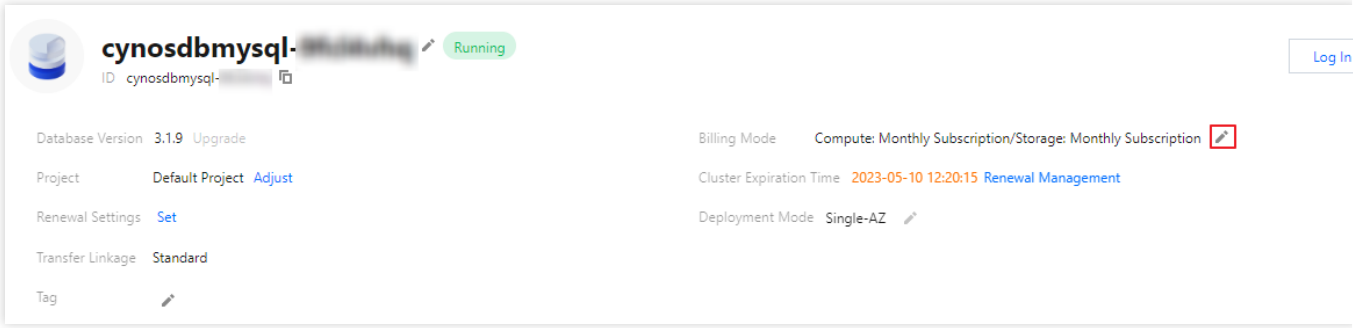
Tab view

List view

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.

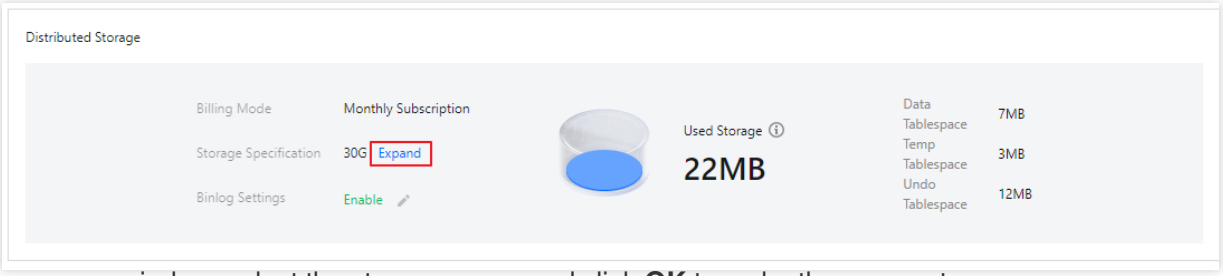
2. Select the billing mode on the cluster management page and click

.

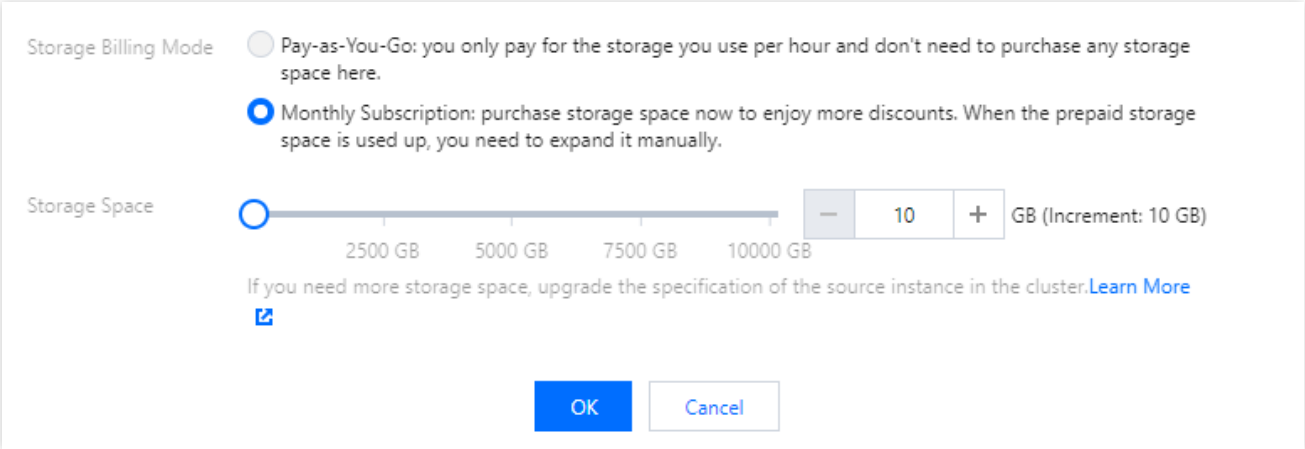


Note:

You can also adjust the distributed storage module under the cluster details page by clicking **Expand** next to the storage specification.



3. In the pop-up window, select the storage space and click **OK** to make the payment.



1. Log in to the TDSQL-C for MySQL console, and click ID of the target cluster in the cluster list or **Manage** in the **Operation** column to enter the cluster management page.
2. Select **Cluster Details** > **Configuration Info** > **Database Storage (Used/Total)** and click

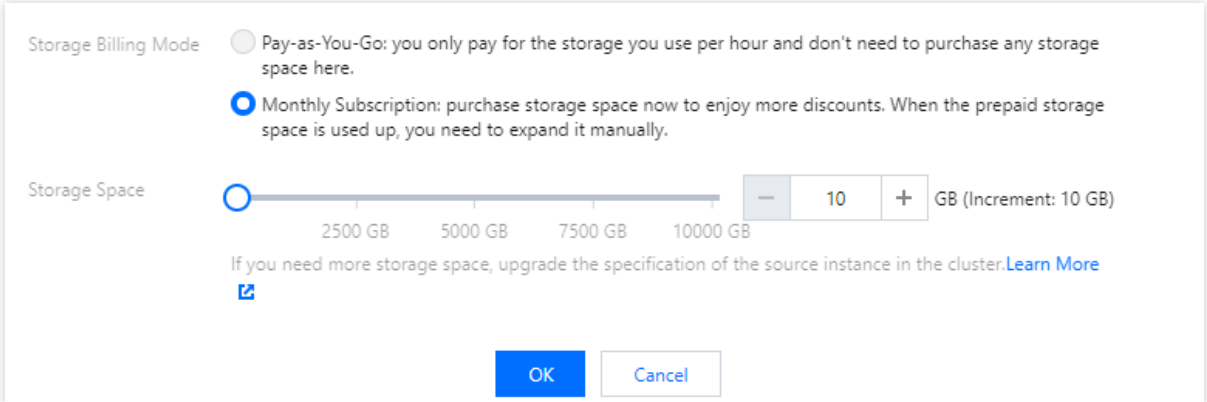
.

Note:

You can also click **Billing Info** > **Storage Billing** on the cluster details page, and click

for adjustment.

3. In the pop-up window, select the storage space and click **OK** to make the payment.



Upgrading Kernel Minor Version

Last updated：2024-06-18 09:32:10

TDSQL-C for MySQL supports automatic or manual kernel minor version upgrade. Upgrading adds new features, improves performance, and fixes issues.
For more information on the TDSQL-C for MySQL kernel minor version, see Kernel Version Release Notes.

Overview

Automatic upgrade:
Scenario 1: When the cluster encounters a critical bug or security vulnerabilities, the system will perform database kernel minor version upgrade during the maintenance window and send upgrade notifications through the Message Center and SMS.
Scenario 2: Upon the release of a new kernel minor version, you can set a version upgrade period (such as not upgrading within 30 days after release). Outside this period, when operations that trigger cluster migration occur (such as cluster specification scaling, disk capacity scaling, and database version upgrade), the system will upgrade your cluster to the latest kernel minor version.
Manual upgrade:
Upon the release of a new kernel minor version, you can set a version upgrade period (such as not upgrading within 30 days after release). Within this period, when operations that trigger cluster migration occur (such as cluster specification scaling, disk capacity scaling, and database version upgrade), the system will not upgrade your cluster to the latest kernel minor version. If an upgrade is required, you can manually upgrade the kernel minor version.

Upgrade Rules

To ensure database replication consistency, the kernel minor versions of all associated instances (read-write and read-only instances) in the cluster will also be upgraded.
TDSQL-C for MySQL upgrade involves data migration. The time it takes for migration depends on the data size. Your business will not be affected during the upgrade and can be accessed as per usual.

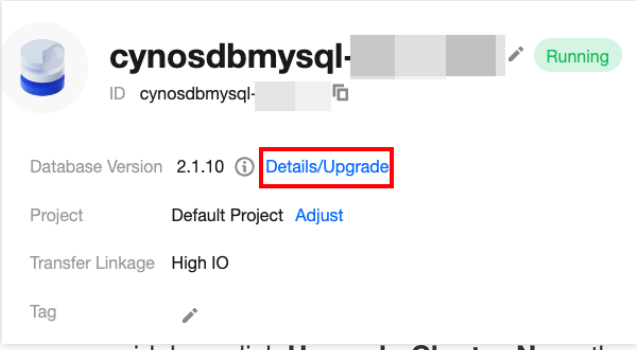
Notes

Cluster switch may be needed after version upgrade is completed (that is, the database may be disconnected for seconds). We recommend you use applications configured with auto reconnection feature and conduct the switch during the instance maintenance window.

If the number of tables in a single instance exceeds one million, upgrade may fail and database monitoring may be affected. Make sure that the number of tables in a single instance is below one million.
The kernel minor version cannot be downgraded once upgraded.

Upgrading Kernel Minor Version Immediately

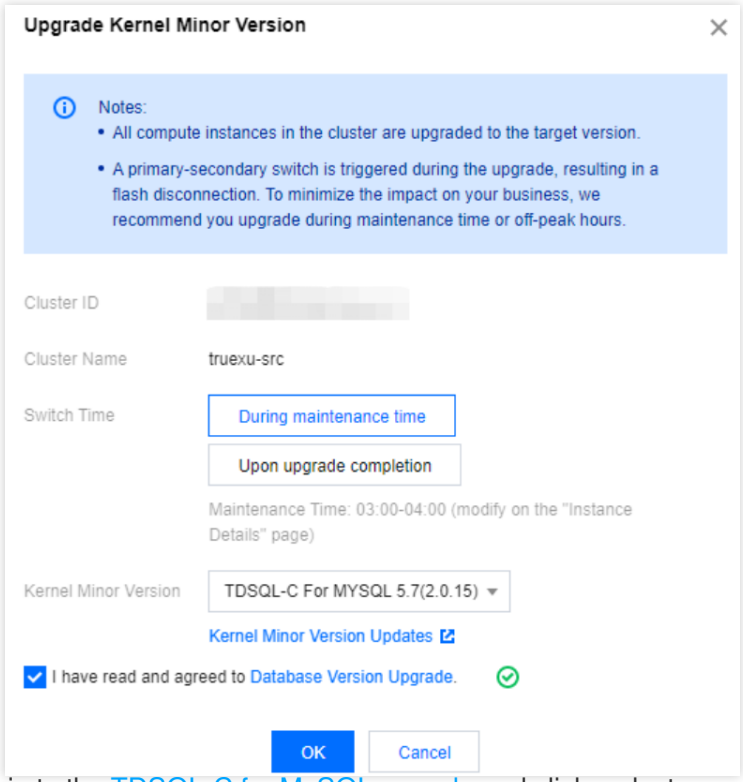
On the cluster list page, proceed based on the actually used view mode:
Tab View
List view
1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.
2. On the cluster management page, after the database version, click **Details/Upgrade**.



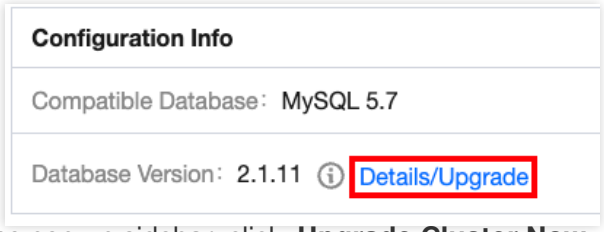
3. In the pop-up sidebar, click **Upgrade Cluster Now**, then in the pop-up window, select the switch time, specific kernel minor version, and click **OK**.

Note:

As database upgrade involves data migration, after the upgrade is completed, a momentary disconnection from the database lasting for just seconds may occur. We recommend you select **During maintenance time** as the switch time, so that the switch will be initiated within the next maintenance time after the instance upgrade is completed.



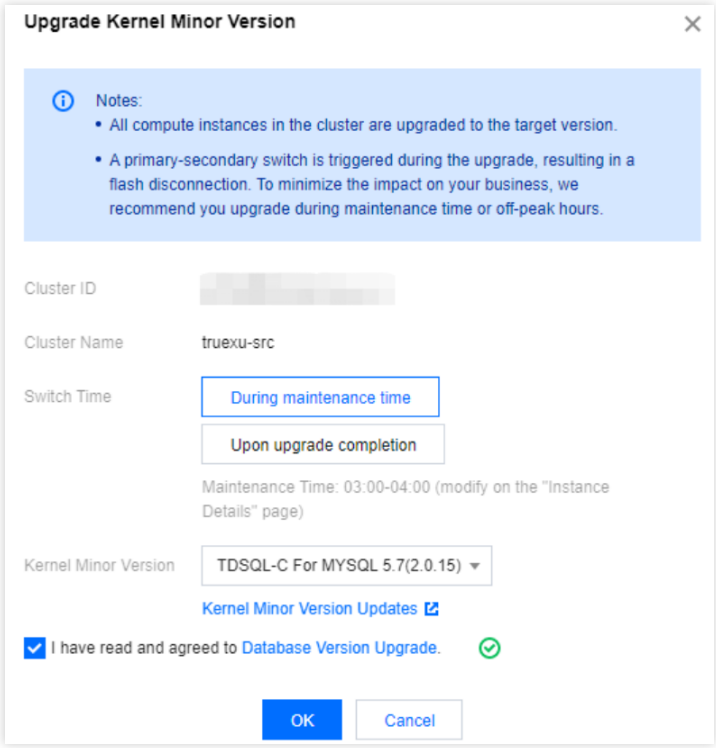
1. Log in to the TDSQL-C for MySQL console and click a cluster name in the cluster list or **Manage** in the **Operation** column to enter the cluster details page.
2. On the cluster details page, under **Configuration Info** > **Database Version**, click **Details/Upgrade**.



3. In the pop-up sidebar, click **Upgrade Cluster Now** . In the pop-up window, select the switch time and specific kernel minor version, and then click **OK** .

Note:

As database upgrade involves data migration, after the upgrade is completed, a momentary disconnection from the database lasting for just seconds may occur. We recommend you select **During maintenance time** as the switch time, so that the switch will be initiated within the next maintenance time after the instance upgrade is completed.



Setting Kernel Minor Version Upgrade Deadline

Note:

After the kernel minor version upgrade deadline is set, the system will send upgrade notifications to your Message Center account 5 days before the deadline and on the deadline day.

On the cluster list page, proceed based on the actually used view mode:

Tab View

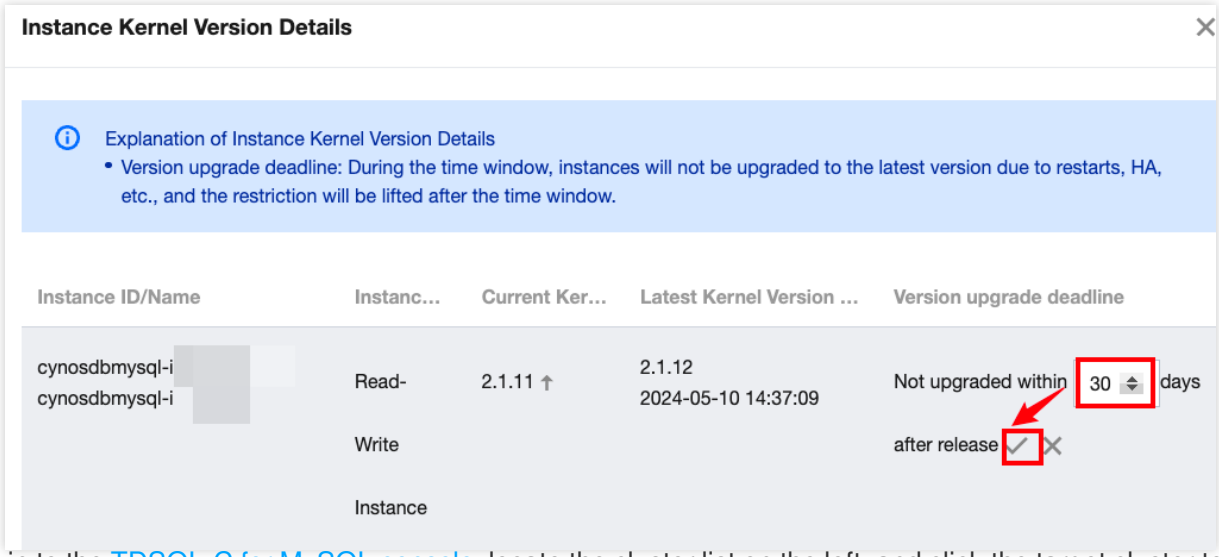
List view

1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.
2. On the cluster management page, click **Details**/**Upgrade** next to the database version.
3. In the pop-up sidebar, click the edit icon

after **Version upgrade deadline** .

4. Set the number of days after publication during which no upgrades will be performed (0 to 90 days supported), and then click

.

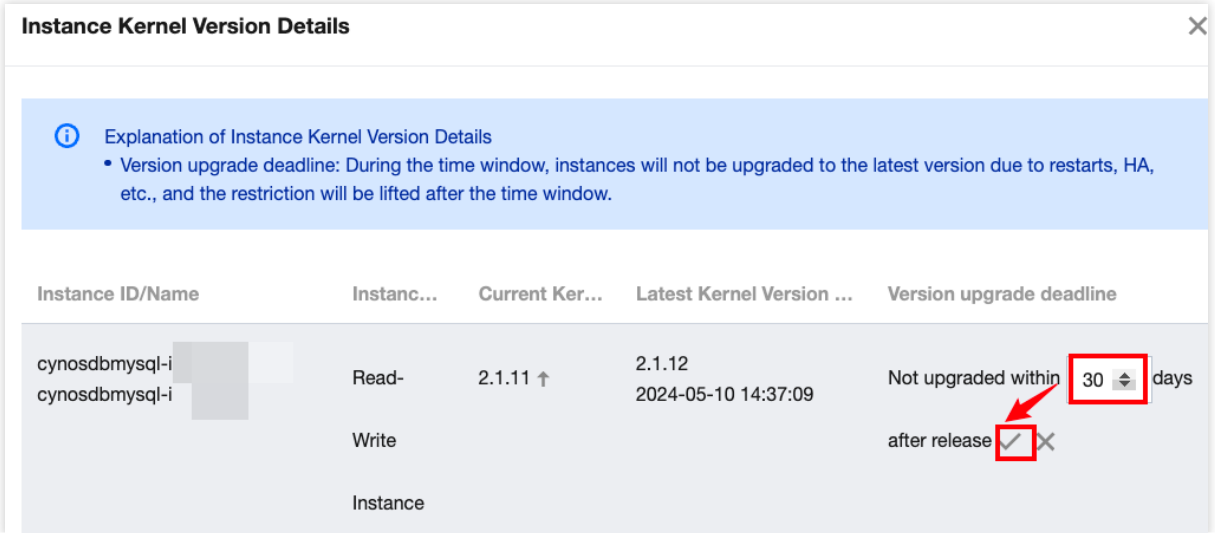


1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.
2. On the cluster details page, under **Configuration Info** > **Database Version** , click **Details/Upgrade** .
3. In the pop-up sidebar, click the edit icon

after **Version upgrade deadline** .

4. Set the number of days after publication during which no upgrades will be performed (0 to 90 days supported), and then click

.



Cluster Management Overview

Last updated：2023-01-04 10:47:30

This document describes what a TDSQL-C for MySQL cluster is and how to view the cluster list and manipulate a cluster.

Cluster overview

A cluster is a group of network resources containing read-write and read-only instances. As the basic management unit of TDSQL-C for MySQL, a cluster can have up to one read-write instance and 15 read-only instances.

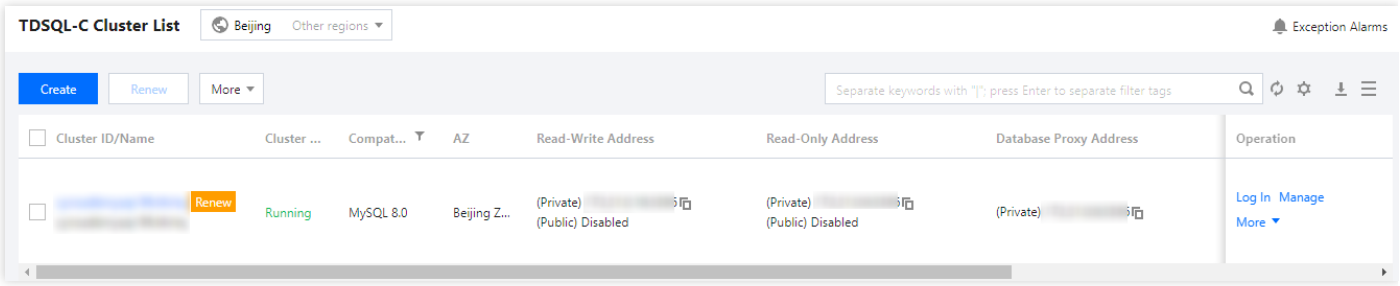
Cluster management overview

Cluster Configuration Item	Description	Operation Method
Renaming a cluster	• You can rename a cluster. • The cluster name can contain up to 60 letters, hyphens, underscores, and dots.	Renaming Cluster
Modifying the cluster project	You can set and modify the project of a cluster.	Modifying Cluster Project
Deleting a cluster	You can delete a cluster when it is no longer used.	Deleting Cluster
Restoring a cluster	• You can restore an isolated cluster as needed. • The recycle bin displays the period during which the cluster can be restored.	Restoring Cluster

Viewing the cluster list

1. Log in to the TDSQL-C for MySQL console.

2. Select the region and MySQL engine at the top to view the cluster list.



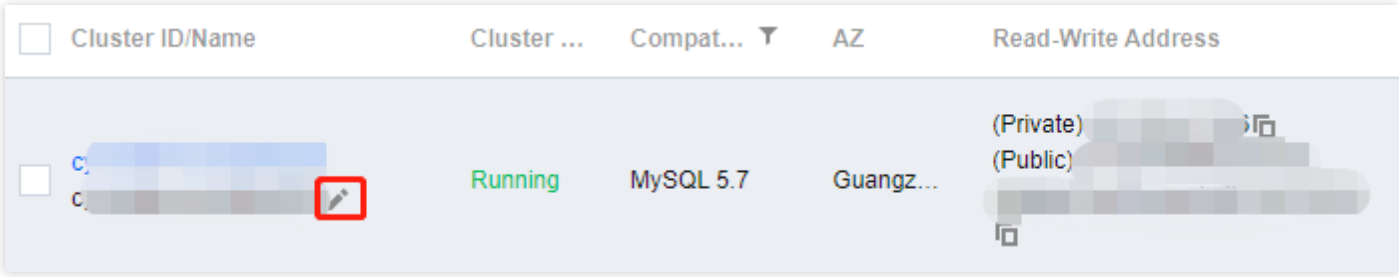
Renaming Cluster

Last updated：2022-04-01 22:57:02

You can name TDSQL-C for MySQL clusters to distinguish between and manage them. The default cluster name is the cluster ID, which you can change in the console.

Renaming Cluster in Cluster List

1. Log in to the TDSQL-C for MySQL console and click after a cluster name in the cluster list.



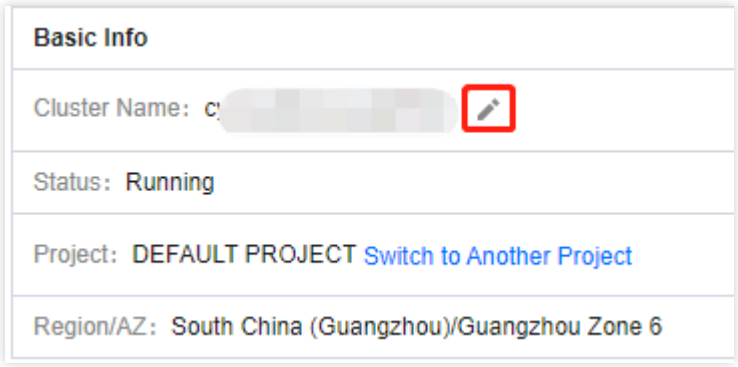
2. In the pop-up window, enter the new cluster name and click **OK**.

> Note：
> The cluster name can contain up to 60 letters, digits, hyphens, underscores, and dots.

Renaming Cluster in Cluster Details

1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list to enter the cluster details page.

2. On the cluster details page, click after the cluster name.



3. In the pop-up window, enter the new cluster name and click **OK**.

Note：
The cluster name can contain up to 60 letters, digits, hyphens, underscores, and dots.

Modifying Cluster Project

Last updated：2022-12-01 11:21:44

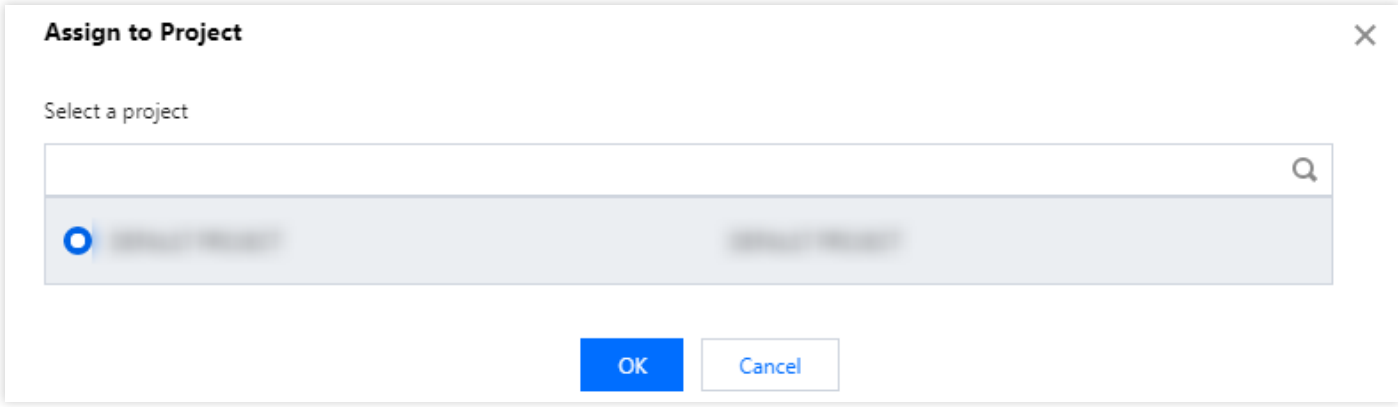
TDSQL-C for MySQL allows you to assign a cluster to different projects for management.

This document describes how to modify the project of a cluster in the console.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Click the **ID** of the target cluster in the cluster list or **Manage** in the **Operation** column to enter the cluster details page.
3. On the cluster details page, select the **Basic Info** tab, find **Project**, and click **Switch to Another Project**.

4. In the pop-up window, select a project and click **OK**.



Deleting Cluster

Last updated：2024-06-17 16:10:26

You can delete a TDSQL-C for MySQL cluster when it is no longer used.
This document describes how to delete a cluster in the console.

Note:

After a cluster is deleted, all instances (including read-write and read-only instances) in it will also be automatically deleted.
After a pay-as-you-go cluster is deleted, its billing will stop automatically.
If a monthly subscription cluster is deleted before it expires, the fees of all instances in it will be calculated again by the usage duration at the pay-as-you-go price, and your original payment will be refunded after fees incurred are deducted.

Use Cases

Delete the entire cluster.

Lifecycle

After a monthly subscribed instance is deleted, it will be moved to the recycle bin and retained there for seven days. During the retention period, the instance cannot be accessed. However, it can be restored after renewal.
After a pay-as-you-go/serverless instance is deleted, it is moved to the recycle bin and retained there for three days. During this period, the instance cannot be accessed. If you wish to restore the instance, you can renew it in the recycle bin.

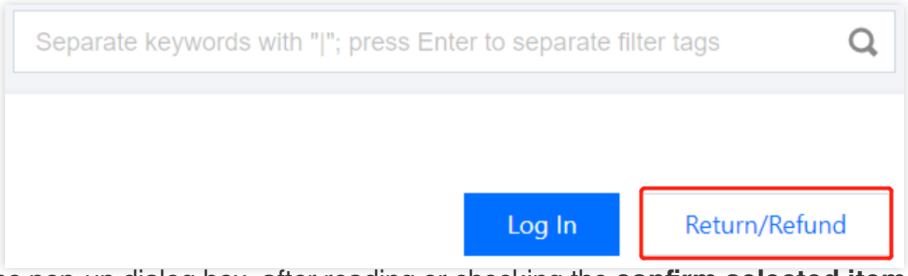
Note：

For Serverless clusters, the scenario of renewal and recovery from the recycle bin is exclusively applicable to read-write instances. Read-only instances that have been reduced through the Read-Only Instance Management operation will not be isolated in the recycle bin but will be directly eliminated.

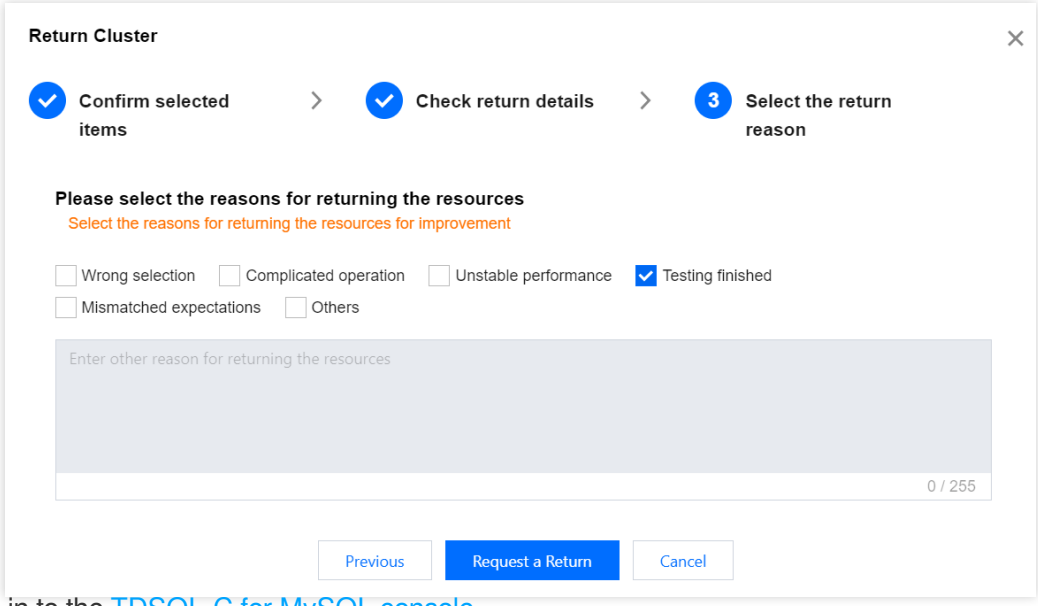
Directions

On the cluster list page, proceed based on the actually used view mode.
Tab view
List view

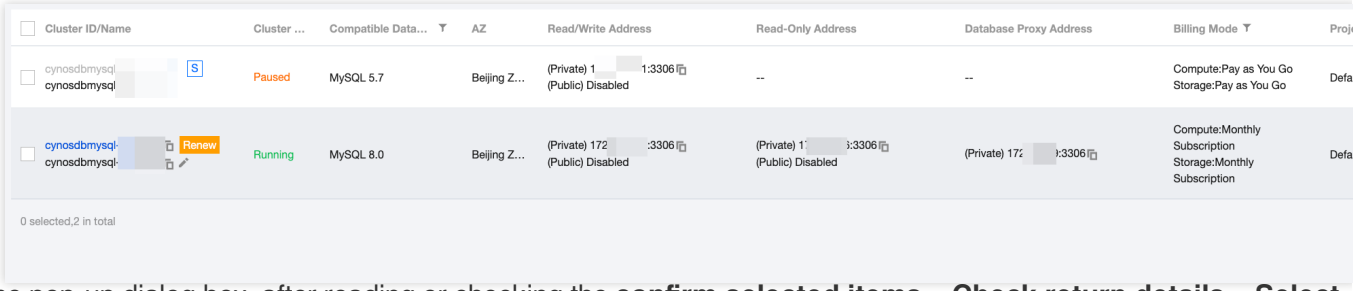
1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.
2. On the cluster management page, click **Return/Refund** on the right.



3. In the pop-up dialog box, after reading or checking the **confirm selected items** > **Check return details** > **Select the return reason**, click **Request a Return**.



1. Log in to the TDSQL-C for MySQL console.
2. Find the target cluster in the cluster list and click **More** > **Return/Refund** in the **Operation** column.



3. In the pop-up dialog box, after reading or checking the **confirm selected items** > **Check return details** > **Select the return reason**, click **Request a Return**.
4. After the cluster is deleted, its instances will be automatically moved into the recycle bin. The instances in the recycle bin cannot be accessed and managed, and no charges will be incurred. However, they can be restored from the recycle bin during this period.

Restoring Cluster

Last updated：2024-06-17 10:59:23

This document describes how to restore an isolated cluster in the console.

Overview

If a cluster is deleted by mistake, due to overdue payment, or upon instance expiration, you can restore all its read-write and read-only instances from the recycle bin before it is eliminated.

Note

After the cluster's read-write and read-only instances are restored, their configurations will remain unchanged.

A cluster cannot be deleted and recovered in consecutive attempts.

For Serverless clusters, the scenario of restoration from the recycle bin is exclusively applicable to read-write instances. Read-only instances that have been reduced through the Read-Only Instance Management operation will not be isolated in the recycle bin but will be directly eliminated.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select **Recycle Bin** on the left sidebar, select the region, find the target read-write and read-only instances in the recycle bin, and click **Restore** in the **Operation** column of the read-write instance.
3. Click **Restore** in the **Operation** column of the read-only instance.

Note

Read-only instances can be restored only after the read-write instance is restored.

You can only restore an instance before it is terminated.

If you click **Release Now** in the **Operation** column, the instance will be eliminated immediately and cannot be restored. Therefore, proceed with caution.

4. After the restoration is completed, the cluster becomes **Running**, and you can see it in the cluster list.

Scaling Instance
Read-Only Instance Overview

Last updated：2023-08-22 15:30:08

In scenarios with more reads but fewer writes, a single instance may not be able to handle the read load, which may even affect the business. To implement the auto-scaling of read capabilities and mitigate the pressure, you can create one or multiple read-only instances to sustain high numbers of database reads and increase the application throughput.

Prerequisites

The compatible database version is MySQL 5.7 or 8.0 (kernel version 3.1.2 or later).

Billing

Instance Specification Billing Mode: The billing mode of the read-only instance is the same as that of the read-write instance. For example, if the read-write instance is monthly subscribed, so is the new read-only instance.
Storage Billing Mode: As TDSQL-C for MySQL is billed by the actual storage used per hour, you don’t need to purchase storage space in advance. The storage of the read-only instance uses that of the entire cluster

Adjusting read-only instance configuration

Region and AZ: Read-only instance needs to be deployed in the same AZ of a region where the read-write instance resides.
Database Compatibility: It must be the same as that of the read-write instance.
Specification: It may be different from that of the read-write instance and can be modified at any time. The read-only instance must have a specification not less than that of the read-write instance; otherwise, it may experience high latency and load.
Instance Name: It must contain less than 60 letters, digits, or symbols (-_.).
Network: It is the same as that of the read-write instance.
Project: It is the same as that of the read-write instance.
Monitoring and Alarm: It provides rich monitoring views of system performance indicators, such as memory utilization, QPS, TPS, and max connections.

Feature Limits

The number of read-only instances can be created: 15

Database management: Databases can be neither created nor deleted.

Account management: Account creation, deletion, authorization, password modification of the account are not supported.

Creating Read-Only Instance

For more information, see Creating Read-Only Instance.

Creating Read-Only Instance

Last updated：2023-11-01 17:39:26

This document describes how to create a read-only instance in the TDSQL-C for MySQL console.

Overview

TDSQL-C for MySQL allows you to create one or more read-only instances in a cluster, which are suitable for read/write separation and one-write-multiple-read application scenarios and can greatly enhance the read load capacity of your database cluster.
A TDSQL-C for MySQL cluster supports two instance types: read-write instance and read-only instance.
A TDSQL-C for MySQL cluster provides private read-write and read-only addresses by default. You can access all read-only instances at the cluster's private read-only address. After a read-only instance is created, access requests to it at the private read-only address will be automatically forwarded to it.
A read-only instance is billed in the same way as the read-write instance. For more information, see Product Pricing.
Note
For more information on how to access TDSQL-C for MySQL, see Connecting to Cluster.

Note

Read-only and read-write instances share the same storage, so there is no need to maintain the account and database.
A read-only instance does not need to replicate or migrate data, nor does it need to be synced with a read-write instance through binlog. It can be created in seconds.
The delay between a read-only instance and the read-write instance is usually within milliseconds, which can be viewed through the read-only instance delay monitoring metric on the monitoring and alarms page.
Read-only and read-write instances can have different specifications. However, to make it easier for you to adjust the configuration based on the load, we recommend that you keep the specifications of read-only instances the same.
When you add a monthly subscribed read-only instance to a monthly subscribed cluster, the instance will expire at the same time as the cluster.

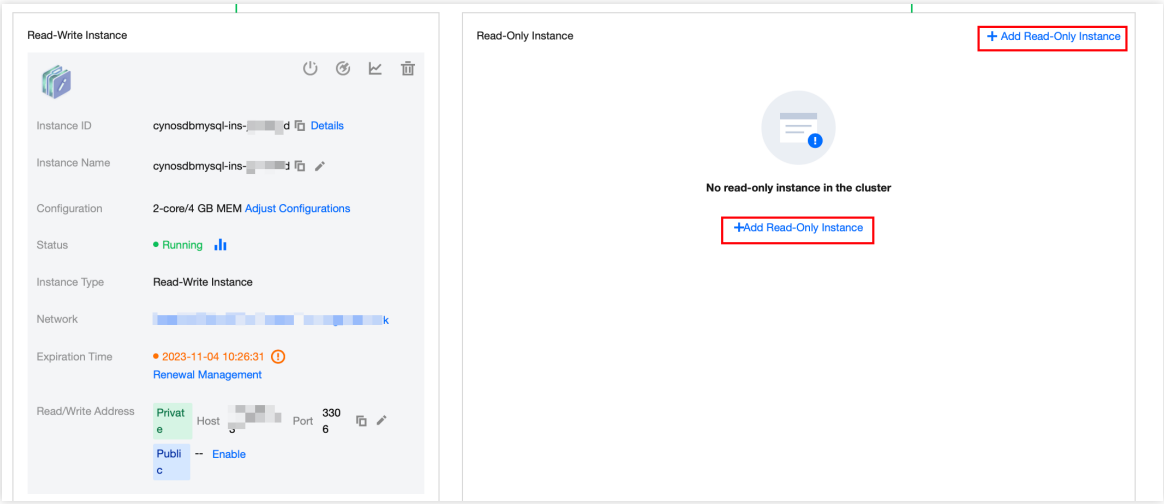
Directions

On the cluster list page, proceed based on the actually used view mode.
Tab view

List view

1. Log in to the TDSQL-C for MySQL console, and click the target cluster in the cluster list on the left to enter the cluster management page.

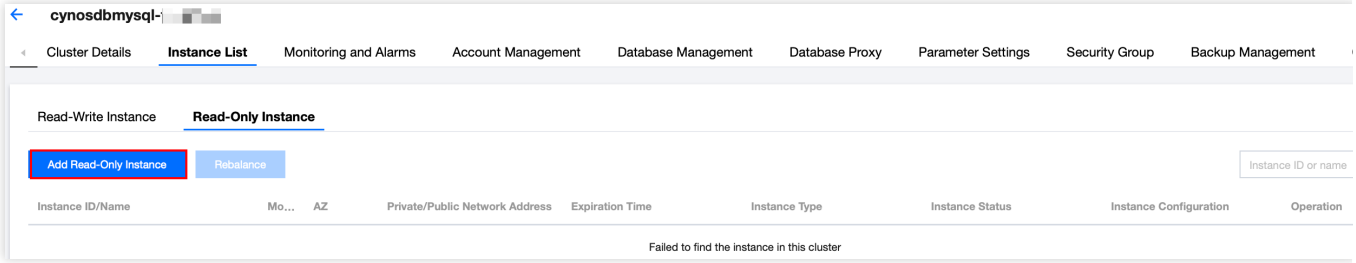
2. On the **Cluster Details** page, click **Add Read-Only Instance**.



3. On the purchase page, select the target read-only instance configuration, confirm that everything is correct, and click **Buy Now**. Then, you can view the new instance in the instance list.

1. Log in to the TDSQL-C for MySQL console and click a cluster ID/name in the cluster list or **Manage** in the **Operation** column to enter the cluster management page.

2. On the cluster management page, select the **Instance List** tab and click **Read-Only Instance** > **Add Read-Only Instance** to enter the read-only instance purchase page.



3. On the purchase page, select the target read-only instance configuration, confirm that everything is correct, and click **Buy Now**. Then, you can view the new instance in the instance list.

Database Proxy
Database Proxy Overview

Last updated：2023-01-03 17:21:17

This document describes the features and use cases of the database proxy service in TDSQL-C for MySQL.
Database proxy is a network proxy service between the TencentDB service and the application service. It is used to proxy all requests when the application service accesses the database.
The database proxy access address is independent of the original database access address. Requests arriving at the proxy address are all relayed through the proxy cluster to access the source and replica nodes of the database. Read/Write requests are separated, so that read requests are forwarded to read-only instances, which lowers the load of the source database.

Note:

The database proxy is currently in the beta test and will be available in more regions. If no database proxy option is offered in your cluster's region, the service is temporarily unavailable there.

Features

High stability

The database proxy is deployed in the cluster architecture, with multiple nodes ensuring smooth failover.

Strong isolation

The database proxy provides the proxy service for the current instance with independent resources. The resources of each proxy are isolated and not shared.

Ultra-high performance

Each proxy can process up to 100,000 requests per second.

Convenient and fast scaling

You can dynamically add 1–60 proxy nodes, with only 6 nodes supported during the beta test.

Comprehensive performance monitoring

Performance metrics are monitored at the second level, such as the number of read/write requests, CPU, and memory. The number of proxies can be adjusted according to the monitoring data and business planning.

Hot reload

When a read-write instance is switched, or its configuration is changed, or a read-only instance is added/removed, the database proxy can dynamically hot reload the configuration without causing network disconnections or restarts.

Automatic read/write separation

It can effectively reduce the read load of the read-write instance when the read/write separation feature of the database proxy is enabled. Read-only instances can be added to horizontally scale the database cluster and

automatically implement read/write separation, which eliminates the complexity of manually separating read and write requests of the business. This is especially suitable for scenarios with a high read load.
After the read/write separation feature of the database proxy is enabled, only one proxy connection address needs to be configured in the application. This address will automatically implement read/write separation and send read requests to read-only instances and write requests to the read-write instance. Even if you add or remove read-only instances, you don't need to adjust the application settings.

Use Cases

Businesses with low performance when there is a large number of non-persistent connections.
Businesses use multiple read-only instances, and read/write separation is implemented manually on applications, resulting in higher maintenance costs and risks.
High instance loading caused by too many connections.

Use Limits

Last updated：2023-11-01 17:21:10

This document describes the use limits for the TDSQL-C for MySQL database proxy.

You can set as many database proxy addresses as there are proxy nodes.

When a proxy connection address is used to implement read/write separation, the consistency of non-transactional reads is not guaranteed. If your business requires read consistency, you can encapsulate it into transactions or use the hint syntax.

When a proxy connection address is used, `show processlist` will merge the results of all nodes before returning them.

For the `PREPARE` statement, the database proxy will first send `PREPARE` to all nodes. When a subsequent `EXECUTE` request comes in, it will determine the execution route based onthe prepared statement type. For example, if a write statement is prepared, it will send the statement to the read-write instance during execution, and if a read statement outside transactions is prepared, it will send the statement to a read-only instance.

After a business connection arrives at the database proxy, the proxy will connect to the read-write instance and all configured read-only instances. The proxy itself does not have a limit on the maximum number of connections, which is mainly subject to the maximum number of connections of the backend database instance. The minimum value of this parameter of the read-write and read-only instances will affect the business performance.

After the database proxy is enabled, when a read-only instance is added or restarted, only new connection requests will be routed to it. You can view the performance metrics of each proxy node through the overview or performance monitoring. If you find that the number of connections on the nodes are unbalanced, you can distribute the connections through rebalancing.

When using the proxy connection address, if you don't enable transaction split, transaction requests will be routed to the source instance.

The database proxy supports cross-AZ configuration. The number of selectable AZs depends on how many AZs are available in the current region. If only one AZ can be selected, there is only one available AZ in the region.

The database proxy feature can only be enabled when the automatic start/stop feature is disabled for the serverless service.

Database Proxy Kernel Features
Kernel Minor Version Release Notes

Last updated：2023-02-07 11:53:30

This document describes the release notes of the TDSQL-C for MySQL database proxy.

Viewing database proxy version

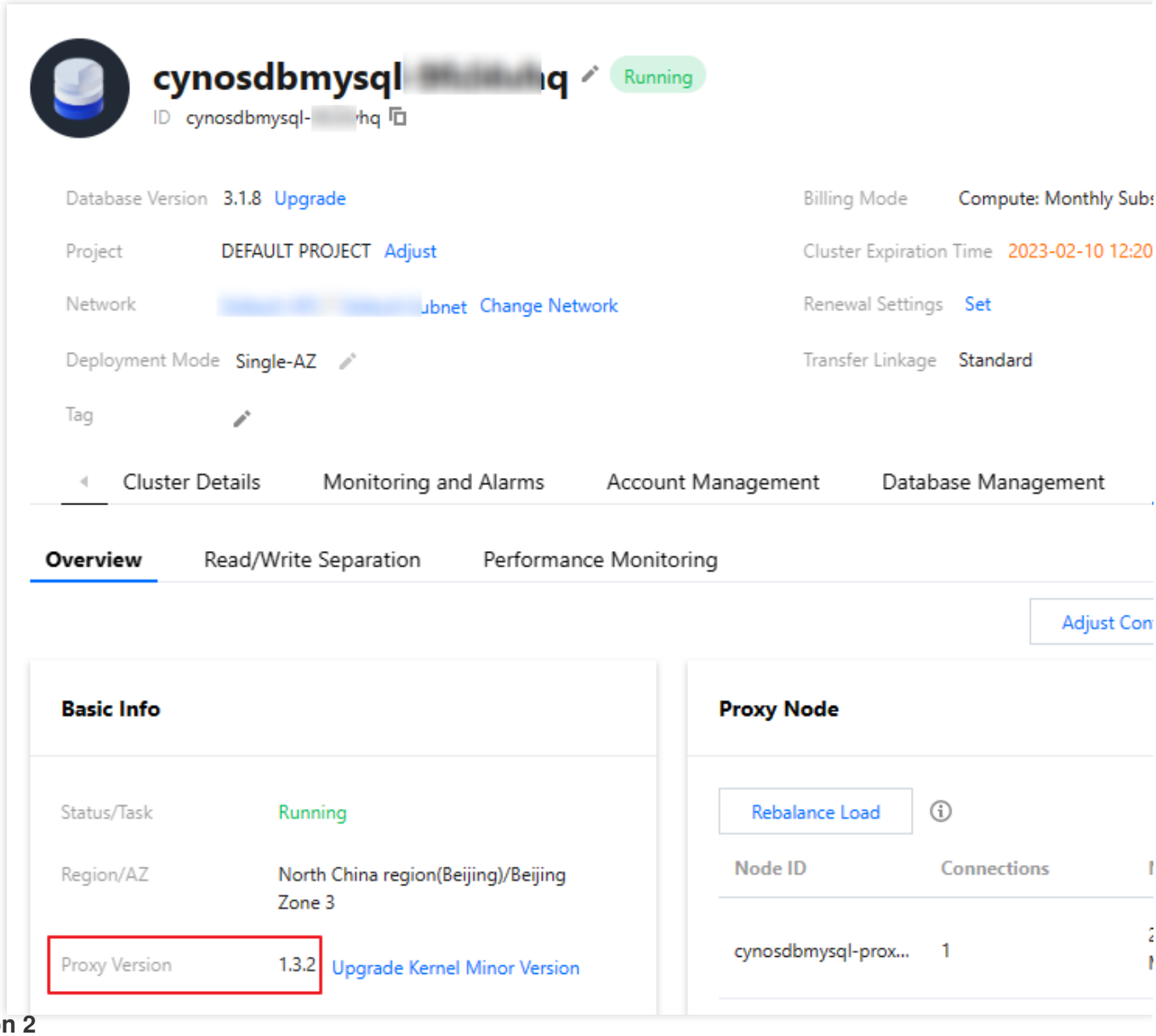
On the cluster list page, proceed according to the actually used view mode:

Tab view

List view

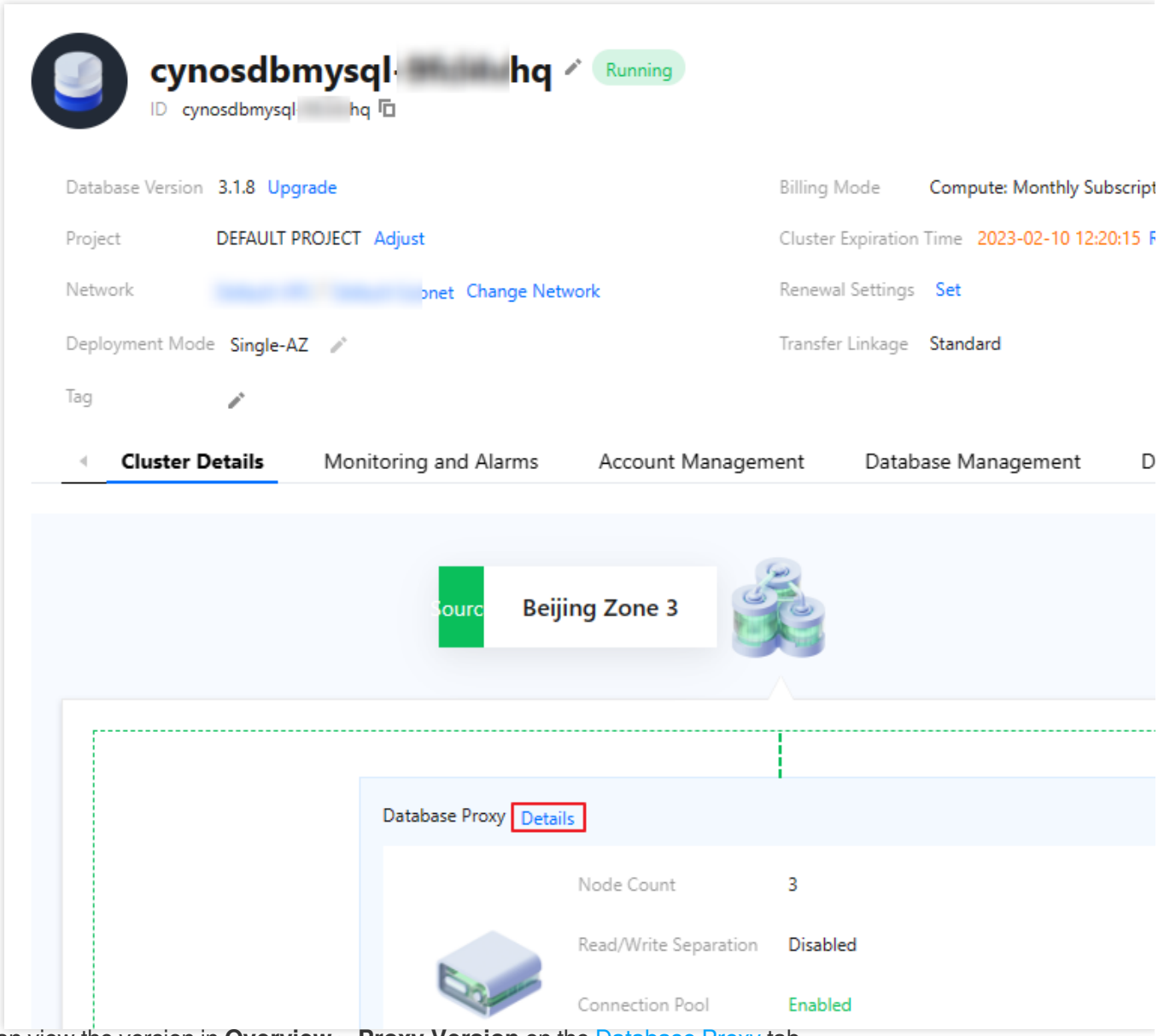
Option 1

On the cluster management page, click **Database Proxy** and view it in **Overview** > **Basic Info** > **Proxy Version**.

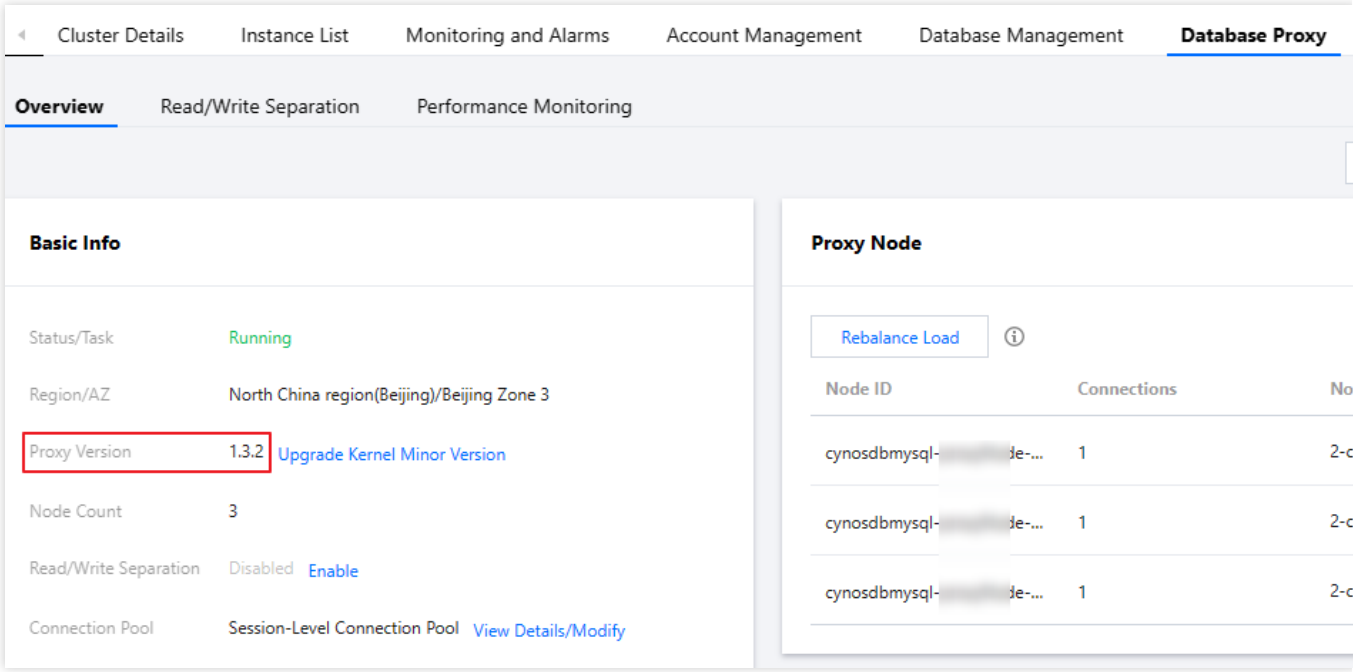


Option 2

On the **Cluster Details** tab on the cluster management page, click **Details** after **Database Proxy** in the architecture to enter the **Database Proxy** page, and view it in **Overview** > **Basic Info** > **Proxy Version**.



You can view the version in **Overview** > **Proxy Version** on the Database Proxy tab.



Version description

Note:

If the TDSQL-C for MySQL kernel version requirements are not met, you can upgrade the kernel version of your database first as instructed in Upgrading Kernel Minor Version.

Version	TDSQL-C for MySQL Kernel Version Requirement	Description
1.3.3	TDSQL-C for MySQL 5.7 ≥ 2.0.20/2.1.6TDSQL-C for MySQL 8.0 ≥ 3.1.6	Fixes Fixed the issue where an error was reported when the session connection pool reused connections to send `change_user` to the backend, and the issue where the PREPARE statement was not correctly handled by the database proxy after a new connection was established. Fixed the issue where the EXECUTE statement didn't have a parameter type.
1.2.1	-	Feature updates Supported MySQL 5.7/8.0. Supported cluster deployment to deploy multiple instances under one database proxy. Supported read/write separation and corresponding weight configuration. Supported the failover feature to send read requests to the read-write instance in case of read-only instance failures.

		Supported the load balancing feature to balance the connections to each proxy node. Supported using the HINT syntax to specify router nodes. Supported the session-level connection pool for scenarios where non-persistent connections were frequently established to the database. Allowed the database proxy to save connections and reuse them subsequently. Supported hot loading to modify configurations online without restarting the dedicated database proxy. Supported the reconnection feature for read-only instances. In persistent connection scenarios, when a read-only instance was restarted or added, the database proxy would automatically establish a connection and restore routing to it.

Upgrading Kernel Minor Version of Database Proxy

Last updated：2023-01-13 14:23:50

This document describes how to manually upgrade the kernel minor version of the database proxy in the console.

Prerequisite

You have enabled the database proxy. For more information,see Enabling Database Proxy.

Notes

There will be a momentary disconnection when you upgrade the kernel minor version of the database proxy. Therefore, upgrade the version during off-peak hours and make sure that your application has an automatic reconnection mechanism.

Directions

1. Log in to the TDSQL-C for MySQL console, select the region at the top of the page, and click the ID of the target cluster to enter the cluster management page.
2. On the cluster management page, select the **Database Proxy** tab.
3. In **Overview** > **Basic Info** > **Proxy Version** on the **Database Proxy** tab, click **Upgrade Kernel Minor Version**.
4. In the pop-up window, check the target version and select the upgrade switch time. After confirming that everything is correct, click **OK**.

Switch Time:

During maintenance time: The upgrade will be performed during the maintenance time, which can be modified on the instance details page.

Upon upgrade completion: The upgrade will be performed immediately after the upgrade operation is confirmed.

Note:

There will be a momentary disconnection during upgrade. Therefore, make sure that your business has a reconnection mechanism.

Except for abnormal nodes, all nodes are upgraded at the same time by default during the kernel minor version upgrade.

Managing Database Proxy
Enabling Database Proxy

Last updated：2023-02-07 12:04:33

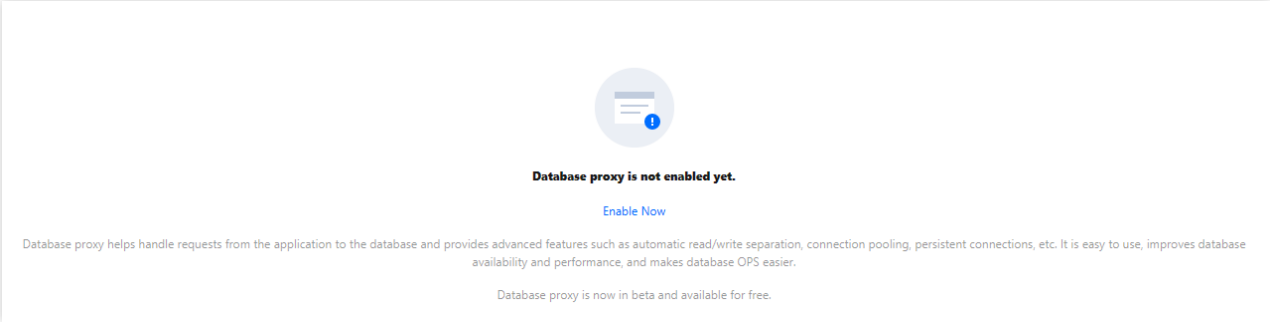
This document describes how to enable the database proxy in the TDSQL-C for MySQL console.
Database proxy is a network proxy service between the TencentDB service and the application service. It is used to proxy all requests when the application service accesses the database. It provides advanced features such as automatic read/write separation, connection pool, connection persistence, and consistency level configuration and boasts high availability, high performance, Ops support, and ease of use.

Notes

Database proxy currently is supported in the following regions:
Beijing (except Zones 1, 2, and 4), Shanghai (except Zone 1), Guangzhou (except Zones 1 and 2), Chengdu, Chongqing, Nanjing, and Hong Kong (China) (except Zone 1).
Tokyo (except Zone 1), Virginia (except Zone 1), Silicon Valley (except Zone 1), Seoul (except Zone 1), and Singapore (except Zones 1 and 2).
Database proxy currently is supported on the following versions: MySQL 5.7 (with kernel minor version 2.0.19 or later) and MySQL 8.0 (with kernel minor version 3.1.5 or later). If you upgrade the kernel minor version of the cluster, the associated read-write and read-only instances will be upgraded at the same time. For more information, see Upgrading Kernel Minor Version.
If the cluster is deployed across AZs, the database proxy feature cannot be enabled currently.

Directions

1. Log in to the TDSQL-C for MySQL console. In the cluster list, click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
2. On the cluster management page, select the **Database Proxy** tab and click **Enable Now**.



3. In the pop-up window, select the specification and node quantity and click **OK**.

Network: Only VPC is supported currently. The VPC of the read-write instance instance is selected by default.

Proxy Specification: 2-core 4000 MB memory, 4-core 8000 MB memory, or 8-core 16000 MB memory.

Node Quantity: Number of proxy nodes. We recommend you set the quantity to 1/8 (rounded up) of the total number of CPU cores on the read-write and read-only instances; for example, if the read-write instance instance has 4 CPU cores, and the read-only instance has 8 CPU cores, then the recommended node quantity will be (4 + 8) / 8 ≈ 2.

Connection Pool Status: For more information on connection pool, see Connection Pool Overview.

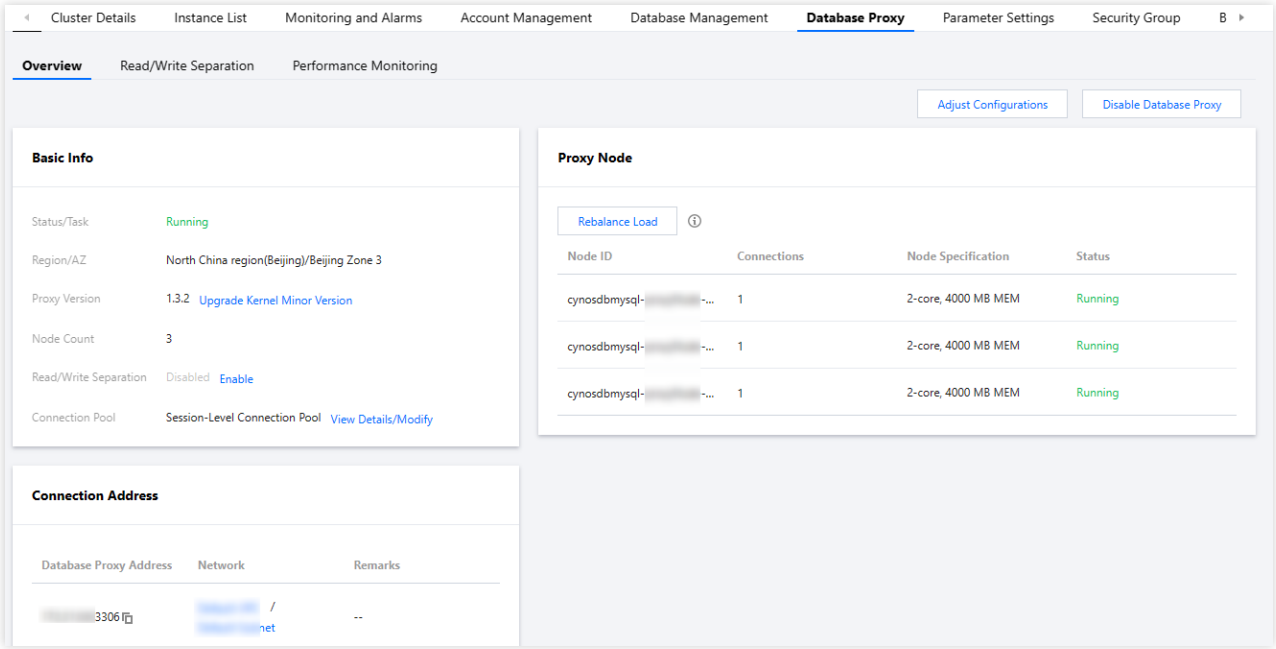
Security Group: It is an important means of network security isolation. You can choose an existing security group or create a new one as needed.

4. After successfully enabling the service, you can manage proxy nodes and view their basic information on the database proxy page. You can also modify the database proxy address and network type and add remarks in the **Connection Address** section.

Note:

You can view **Connections** in the proxy node list or view the performance monitoring data of each proxy node to check whether the numbers of connections on the nodes are unbalanced, and if so, you can distribute the connections by clicking **Rebalance**.

Rebalance will cause proxy nodes to restart, and the service will become unavailable momentarily during the restart. We recommend you restart the service during off-peak hours. Make sure that your business has a reconnection mechanism.


Cluster Details
Instance List
Monitoring and Alarms
Account Management
Database Management
Database Proxy
Parameter Settings
Security Group
Overview
Read/Write Separation
Performance Monitoring
Adjust Configurations
Disable Database Proxy
Basic Info
Status/Task
Running
Region/AZ
North China region(Beijing)/Beijing Zone 3
Proxy Version
1.3.2
Upgrade Kernel Minor Version
Node Count
3
Read/Write Separation
Disabled
Enable
Connection Pool
Session-Level Connection Pool
View Details/Modify
Proxy Node
Rebalance Load
Node ID
Connections
Node Specification
Status
cynosdbmysql-
1
2-core, 4000 MB MEM
Running
Connection Address
Database Proxy Address
Network
Remarks
3306
--

Setting Database Proxy Address

Last updated：2024-03-25 16:04:11

This document describes how to set the database proxy access address in the TDSQL-C for MySQL console.
The database proxy access address is independent of the original database access address. Requests proxied at the proxy address are all relayed through the proxy cluster to access the source and replica nodes of the database. Read/Write requests are separated, so that read requests are forwarded to read-only instances, which lowers the load of the source database.

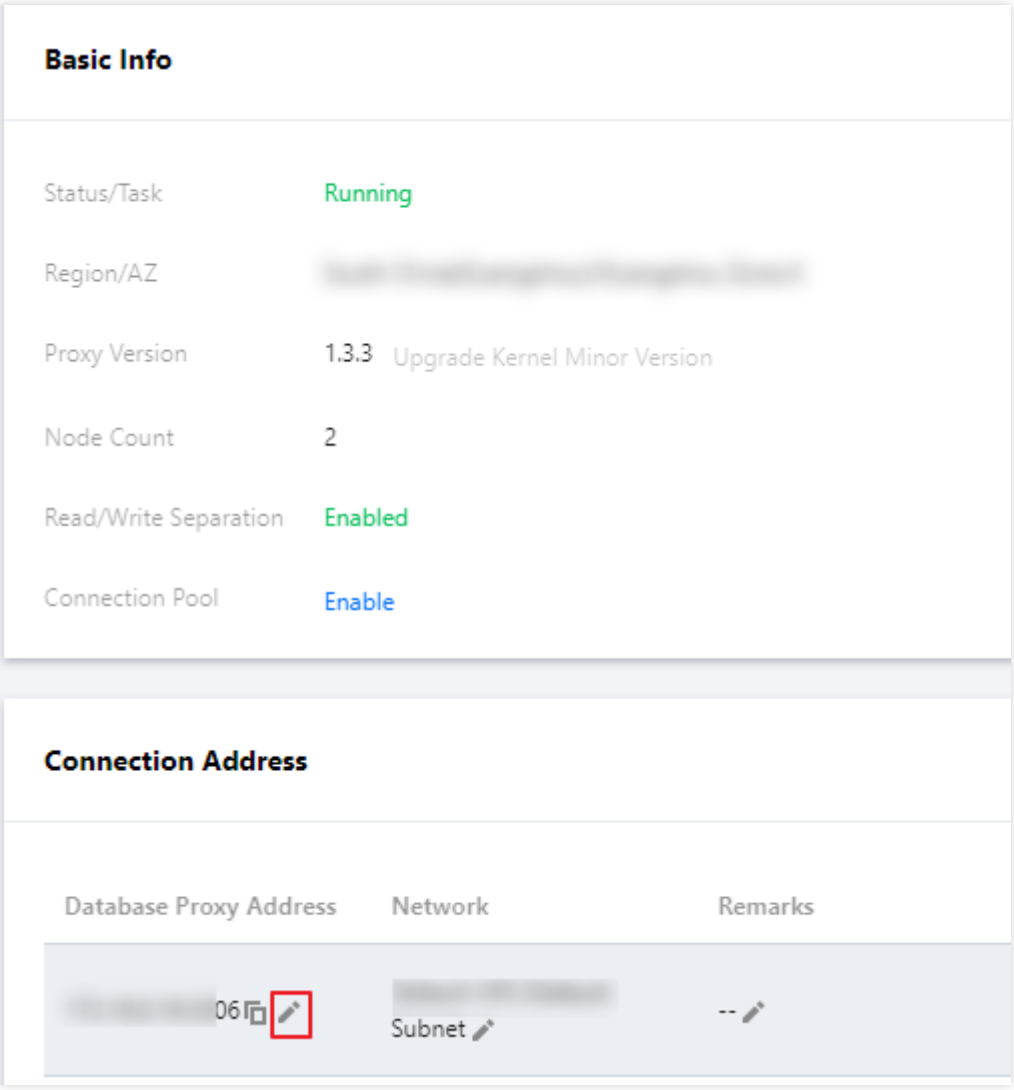
Prerequisite

You have enabled the database proxy. For more information, see Enabling Database Proxy.

Directions

1. Log in to the TDSQL-C for MySQL console. In the cluster list, click the ID or **Manage** in the **Operation** column of the cluster with the proxy enabled to enter the cluster management page.
2. On the cluster management page, select the **Database Proxy** tab and click the

icon next to **Database Proxy Address** on the **Connection Address** tab.

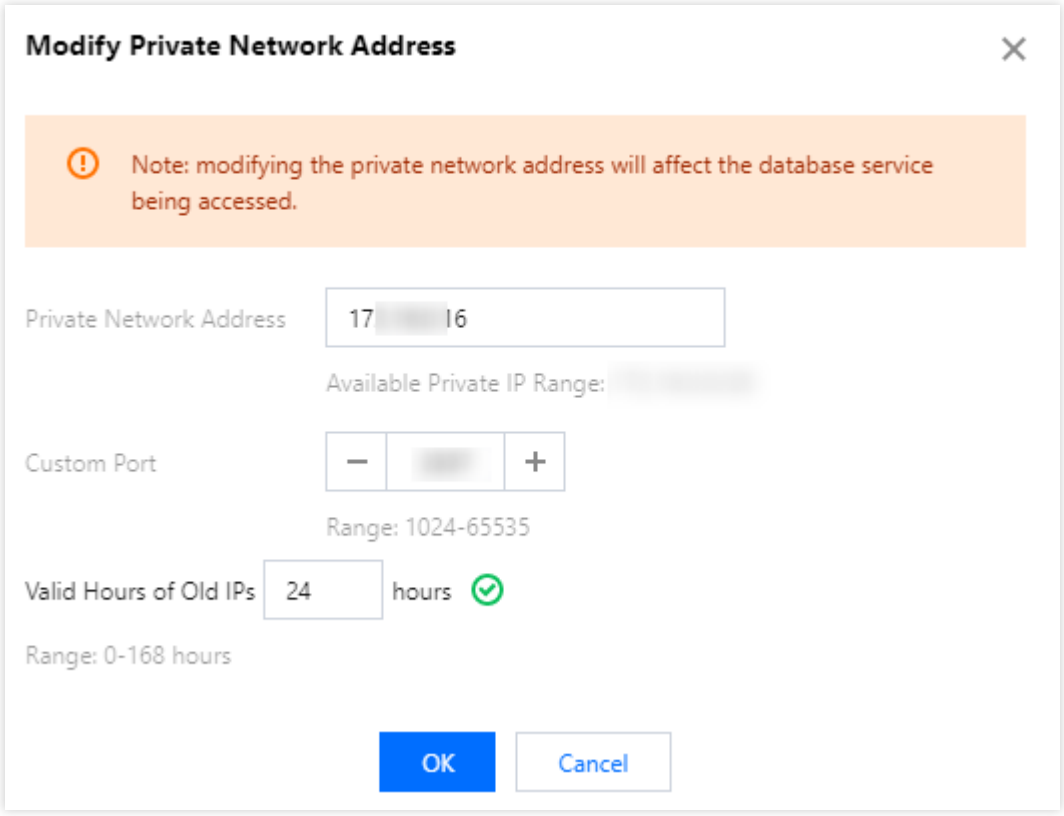


Item	Description
Database Proxy Address	It is independent of the original database access address. Proxied requests are all relayed through the proxy cluster to access the source and replica nodes of the database. It can be edited.
Network	You can switch the instance's network type between classic network and VPC according to your business needs. In the classic network, instances cannot be isolated through the network but rely on their own allowlist policies to block unauthorized access. A VPC is an isolated network environment and has a higher security level.
Remarks	It is a short description of the database proxy address for easier management.

3. In the pop-up dialog box, modify the proxy address and click **OK**.

Note:

Modifying the proxy address affects the database business being accessed. We recommend you modify the address during off-peak hours. Make sure that your business has a reconnection mechanism.



Modifying or Deleting Connection Addresses

Last updated：2023-12-13 10:24:53

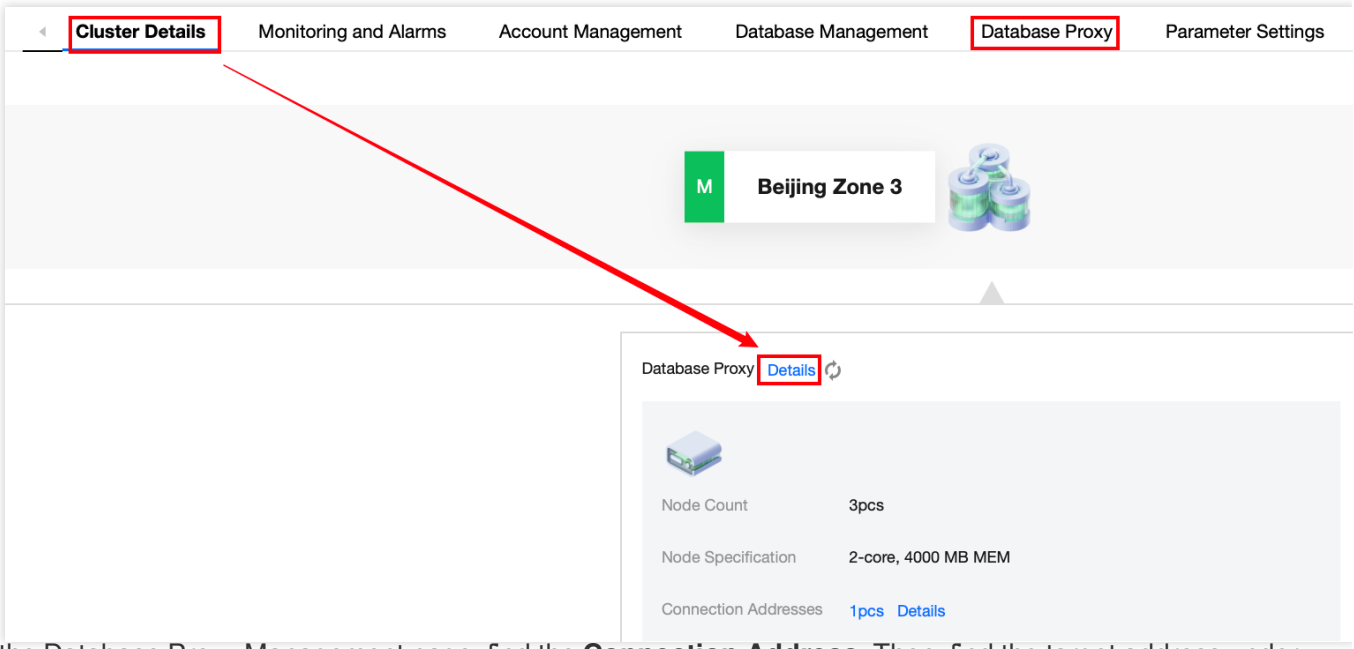
This document describes how to modify or delete database proxy connection addresses in the TDSQL-C for MySQL console.

Prerequisites

You have enabled the database proxy. For details, see Enabling Database Proxy.

Modifying Database Proxy Private Network Addresses

1. Log in to the TDSQL-C for MySQL console and select the **regions** to which the instance belongs.
2. In the cluster list, select the cluster with the proxy enabled and click on its ID to navigate to the cluster management page.
3. On the **Cluster Management** page, click **Cluster Details** > **Details** to open the Database Proxy Management page. Alternatively, you can directly enter the Database Proxy Management page by selecting **Database Proxy** on the **Cluster Management** page.



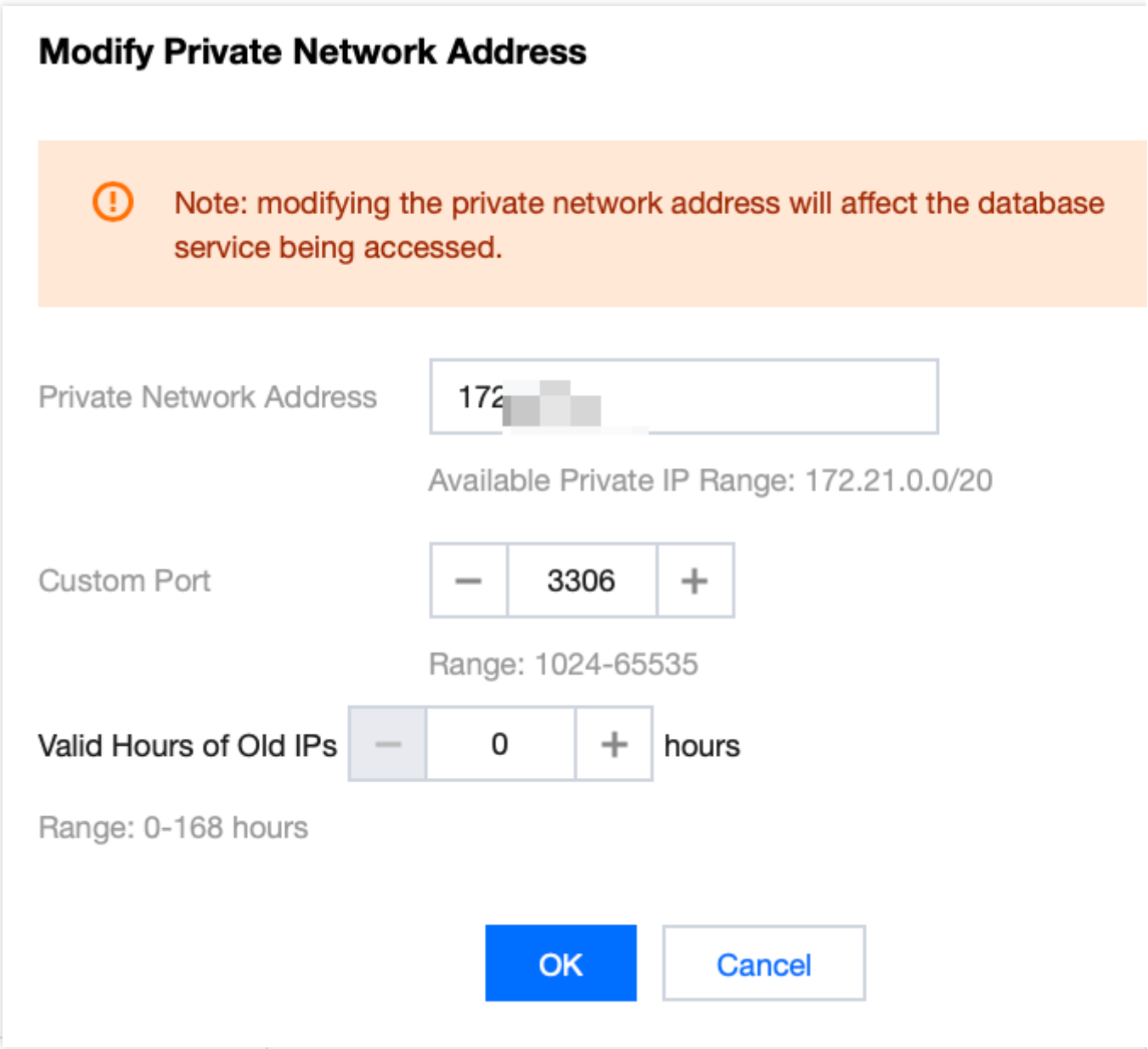
4. On the Database Proxy Management page, find the **Connection Address**. Then, find the target address under **Private Network Access Address** and click the

icon.

5. In the displayed dialog box, modify the private network address, and click on **OK**.

Note

Modifying proxy private network address es impacts the ongoing database operations. Therefore, it is recommended that you modify the address during low-traffic periods and ensure that your business has an automatic reconnection mechanism.



Parameter	Description
Private Network Address	You can customize the private network address within the available IP range.
Custom Port	You can customize the port values, ranging from 1024 to 65535.
Valid Hours of Old IPs	The system default recovery time is 0 hours, indicating that the old IP address will be immediately reclaimed after modification, subsequently disabling access to the database proxy via this address. The time can be customized, ranging from 0 to 168 hours.

Deleting Database Proxy Private Network Addresses

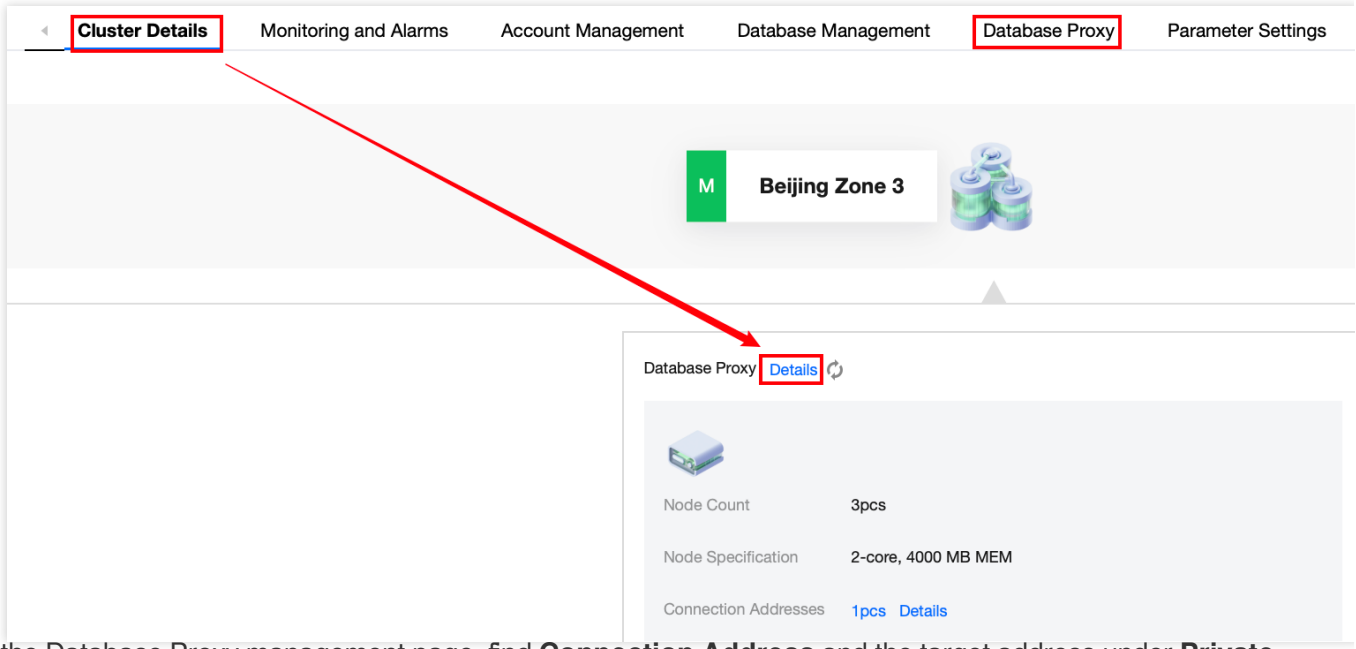
Note

If your database proxy has multiple proxy connection addresses, unnecessary addresses can be deleted. However, only one address cannot be deleted.

1. Log in to the TDSQL-C for MySQL console and select the **regions** to which the instance belongs.

2. In the cluster List, select the cluster with the proxy enabled and click on its ID to navigate to the cluster management page.

3. On the **Cluster Management** page, click **Cluster Details** > **Details** to open the Database Proxy Management page. Alternatively, you can directly enter the Database Proxy Management page by selecting **Database Proxy** on the **Cluster Management** page.



4. On the Database Proxy management page, find **Connection Address** and the target address under **Private Network Access Address**. Then, click **Disable** in the **Operation** column.

5. In the displayed dialog box, click **OK**.

Viewing and Changing the Access Policy

Last updated：2024-03-25 17:53:53

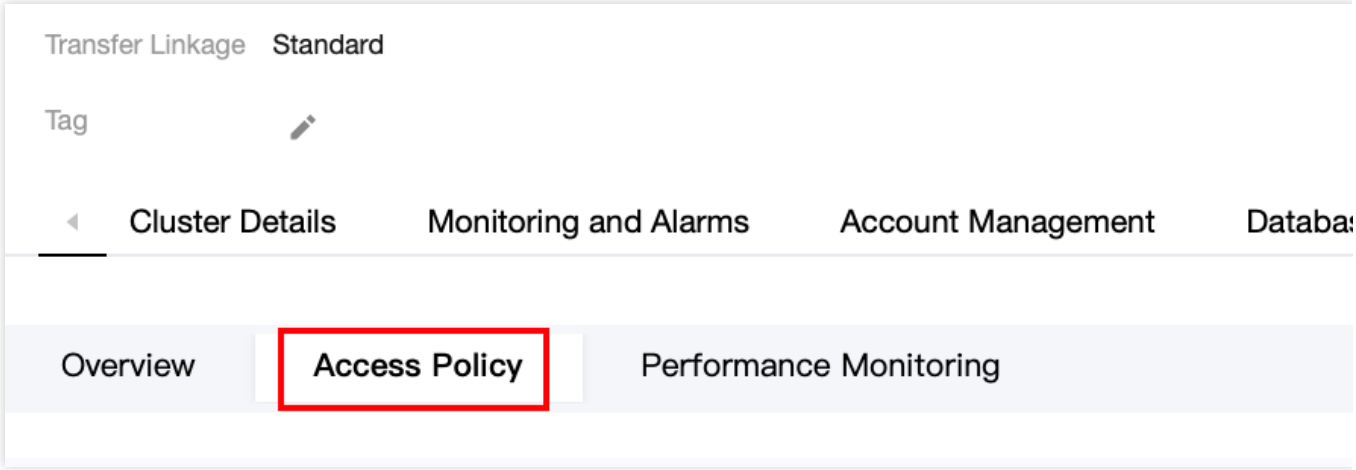
After database proxy is enabled for TDSQL-C for MySQL, a database proxy address will be added by default. You can also add addresses later to implement different business logics. The number of proxy addresses that can be created is the same as that of database proxy nodes. You can view and change the access policy of these addresses in the console.

Prerequisites

You have enabled data proxy. For details, see Enabling Database Proxy.

Viewing the Access Policy

1. Log in to the TDSQL-C for MySQL console. In the cluster list, select the cluster with the proxy enabled and click its ID to navigate to the Cluster Management page.
2. On the Cluster Management page, click **Cluster Details** > **Details** to open the Database Proxy Management page. Alternatively, you can directly enter the Database Proxy Management page by selecting **Database Proxy** on the Cluster Management page.
3. On the Database Proxy management page, choose **Database Proxy** > **Access Policy**.



Changing the Access Policy

1. Log in to the TDSQL-C for MySQL console. In the cluster list, select the cluster with the proxy enabled and click its ID to navigate to the Cluster Management page.

2. On the Cluster Management page, click **Cluster Details** > **Details** to open the Database Proxy Management page. Alternatively, you can directly enter the Database Proxy Management page by selecting **Database Proxy** on the Cluster Management page.

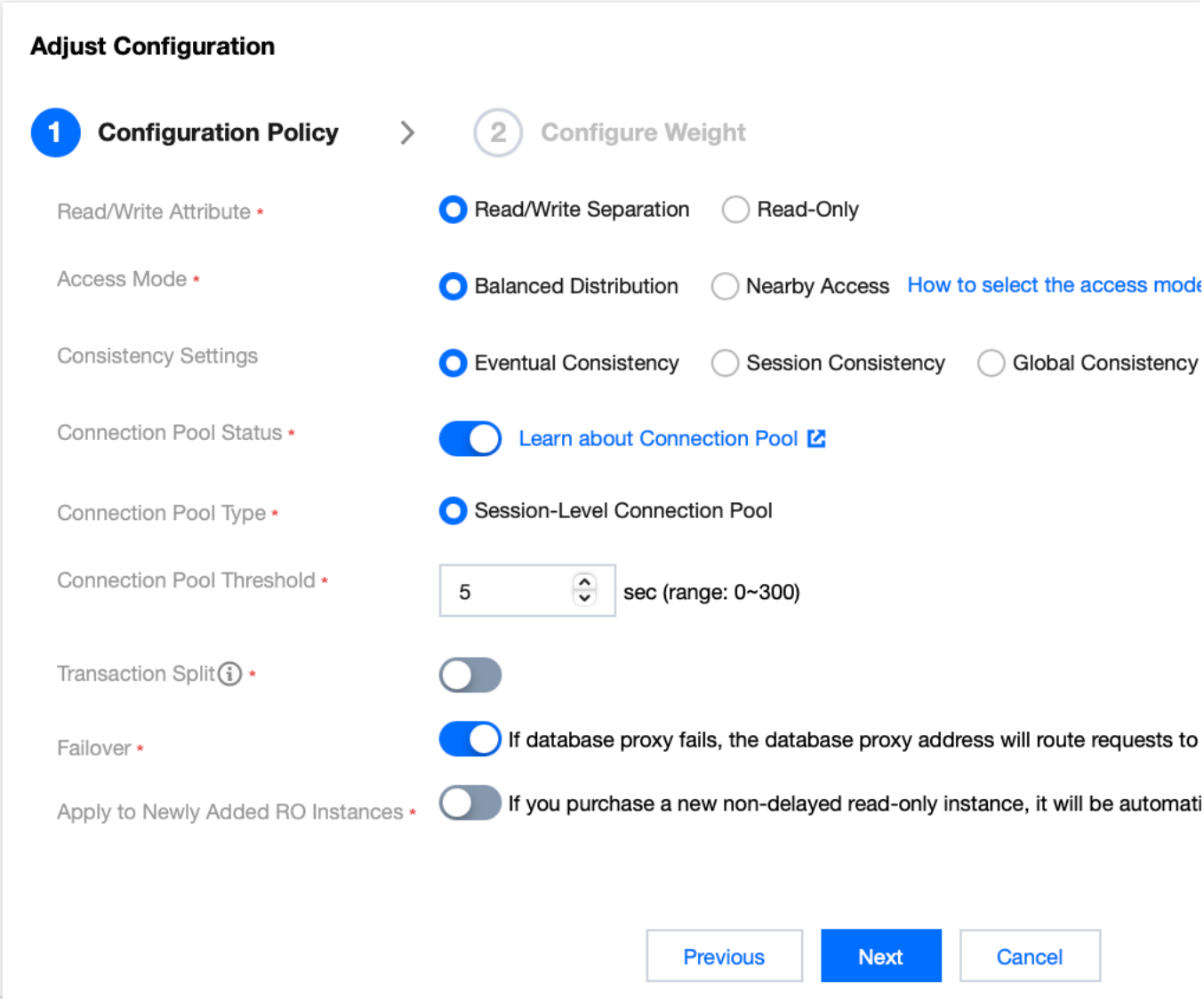
3. On the Database Proxy Management page, select **Access Policy**, find the target access policy, and click **Adjust Configuration**.

Note：

You can also find the target proxy address in **Database Proxy** > **Overview** > **Connection Address**. Then, click **Adjust Configurations** in the **Operation** column.

4. In the pop-up window, change the policy configuration and click **OK**.

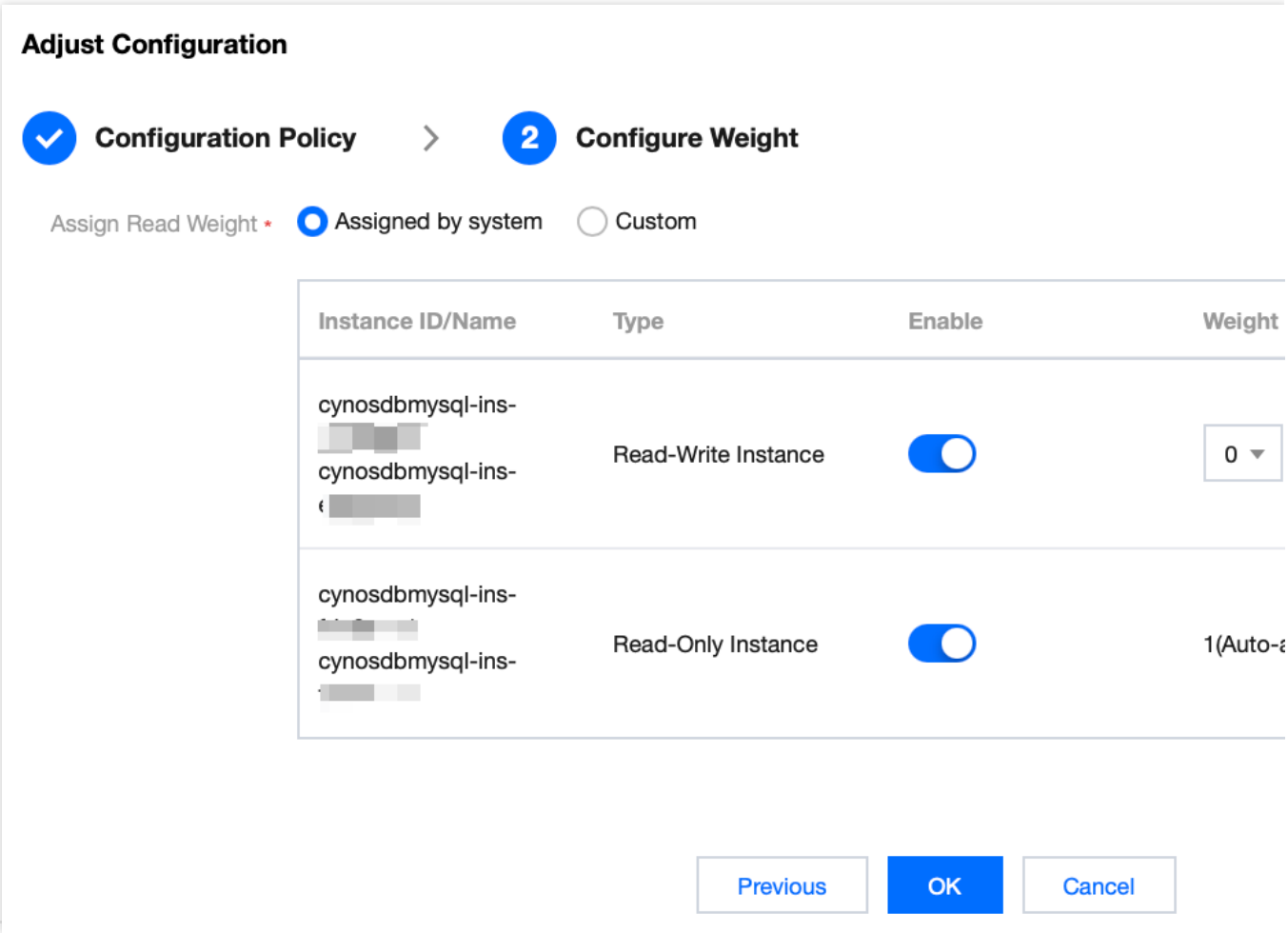
Step 1. Configure the policy



Parameter	Description
Read/Write Attribute	Choose the read/write attribute for this proxy address: Read/Write Separation or Read-Only. If you select Read/Write Separation, the feature will be enabled automatically.

Access Mode	There are two access modes available to manage the connection between the client application and the database: Balanced Distribution and Nearby Access. For details, see Access Mode.
Consistency Settings	Under the Read-Write Separation Attribute, three consistency levels are provided: Eventual Consistency, Session Consistency, and Global Consistency, to meet your consistency requirements in different scenarios. For more information, see Consistency Level.
Connection Pool Status	Connection pooling shares connections across different user requests to reduce the instance load caused by frequent creation of new connections. Enabling this feature allows you to choose a supported connection pool type, with Session-Level Connection Pool as the default.
Connection Pool Threshold	Configure the Connection Pool Threshold. The valid range is between 0 to 300 seconds.
Transaction Split	You can enable this feature to separate reads and writes in a transaction, directing them to different instances for execution. Read requests will be forwarded to read-only instances, effectively reducing the load on the read-write instance.
Failover (For instances with read/write separation)	You can enable this feature to ensure that in the event of a failure in the database proxy, the address will be routed to the read/write separation instance.
Apply to Newly Added RO Instances	You can enable this feature to automatically add newly purchased read-only instances to the database proxy.If Assign Read Weight is set to Assigned by system, newly purchased read-only instances will be assigned with the default weight based on their specification.If Assign Read Weight is set to Custom, when newly purchased read-only instances are added to the RO group, their weights will be 0 by default. You can change the weight on the Configure Weight page.

Step 2: Configure the Weight



Parameter	Description
Assign Read Weight	You can select Assigned by system or Custom. If Custom is selected, you can separately assign the weights of different instances. **Note：** The weight here refers to the distribution policy for read request (non-transactional) weights.

Rebalancing the Load

Last updated：2024-03-25 16:24:32

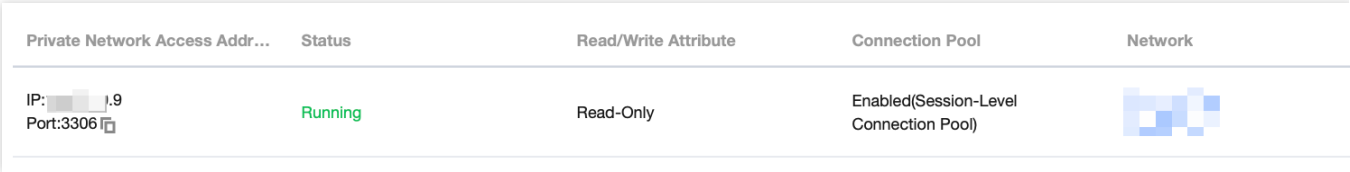
After enabling the database proxy, you can check the connection count or monitor the performance of each proxy node to identify any access imbalances. If there are a significant number of long-lived connections in your business, adding additional database proxy nodes may introduce new load imbalances. If there is an uneven distribution of connections among the nodes, you can manually perform a reload balancing operation through the console. This helps evenly distribute the connections and address any potential load imbalances.

Prerequisites

You have enabled the database proxy. For details, see Enabling Database Proxy.

Directions

1. Log in to the TDSQL-C for MySQL console. In the cluster list, select the cluster with the proxy enabled and click its ID to navigate to the Cluster Management page.
2. On the Cluster Management page, click **Cluster Details** > **Details** to open the **Database Proxy Management** page. Alternatively, you can directly enter the Database Proxy Management page by selecting Database Proxy on the Cluster Management page.
3. On the Database Proxy management page, navigate to **Overview** > **Connection Address** > **Proxy Address** and click **Rebalance Load** in the **Operation** column.



4. In the pop-up window, click **OK**.

Note：

During a reload balancing operation, sessions connected to the address will be disconnected, causing temporary service unavailability. It is recommended to restart the service during off-peak hours and ensure a reliable reconnection mechanism is in place.

Transaction Split Feature

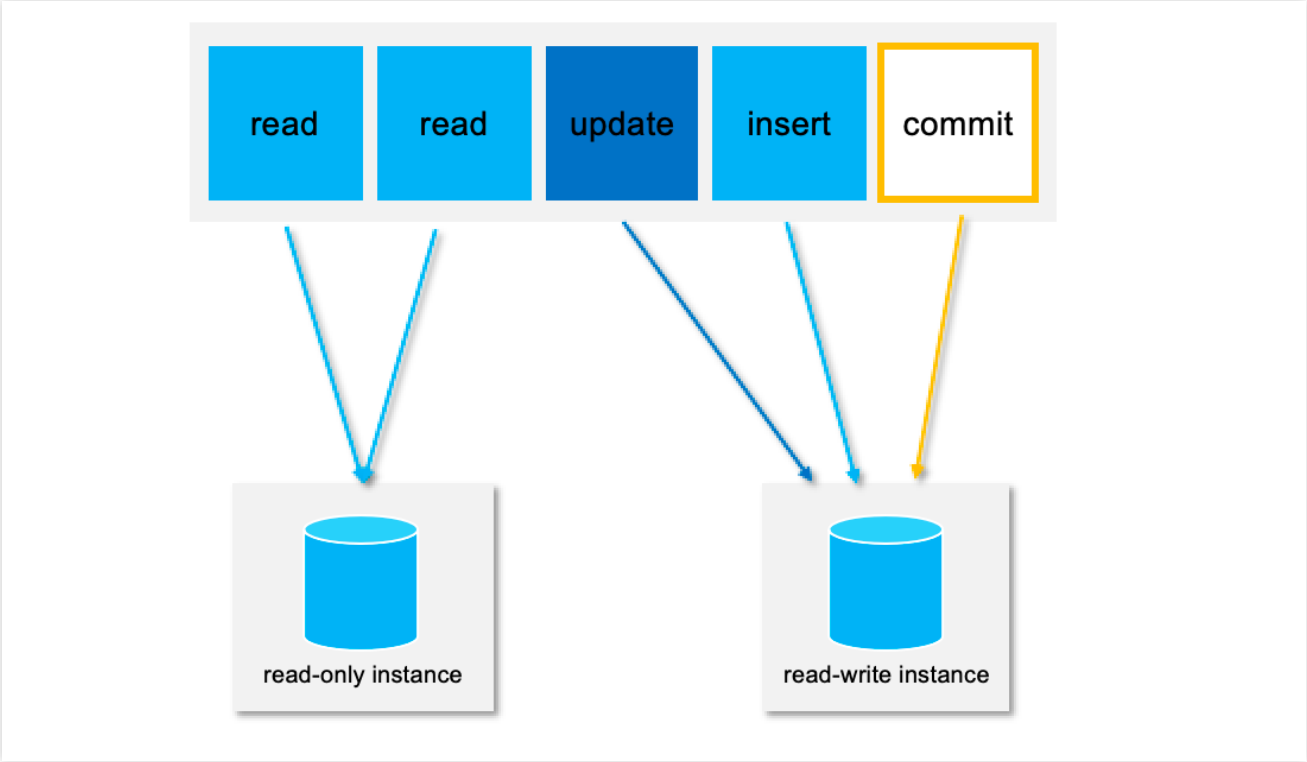
Last updated：2024-03-25 17:11:06

The TDSQL-C for MySQL database proxy provides the transaction split feature. This feature separates read and write operations in a transaction to different instances for execution and forwards read requests to read-only instances, thereby effectively reducing the load on the read-write instance.

Note：

Split is currently not supported by explicit transactions (such as BEGIN TRANSACTION or START TRANSACTION).

Enabling or disabling the transaction split feature will affect newly established connections only.



Background

By default, the TDSQL-C for MySQL database proxy sends all requests within a transaction to the read-write instance to ensure transaction correctness. However, certain frameworks may encapsulate all requests in non-autocommit transactions, leading to increased load on the read-write instance. In such cases, you can use the transaction split feature.

To enable or disable the transaction split feature, you can go to **Database Proxy** > **Overview** > **Connection Address**. Then, click Adjust Configurations in the **Operation** column.

Prerequisites

You have enabled database proxy. For details, see Enabling Database Proxy.

Directions

1. Log in to the TDSQL-C for MySQL console.In the cluster list, select the cluster with the proxy enabled and click its ID to navigate to the Cluster Management page.
2. On the Cluster Management page, click **Cluster Details** > **Details** to open the Database Proxy Management page. Alternatively, you can directly enter the Database Proxy Management page by selecting **Database Proxy** on the Cluster Management page.
3. On the Database Proxy Management page, select **Access Policy**, find the target access policy, and click **Adjust Configuration**.

Note：

You can also find the target proxy address in **Database Proxy** > **Overview** > **Connection Address**. Then, click **Adjust Configurations** in the **Operation** column.

4. In the Adjust Configuration window, enable or disable the transaction split feature and click **OK** to save the changes.

Access Mode

Last updated：2024-03-25 16:16:49

This document describes the access modes of the TDSQL-C for MySQL database proxy.

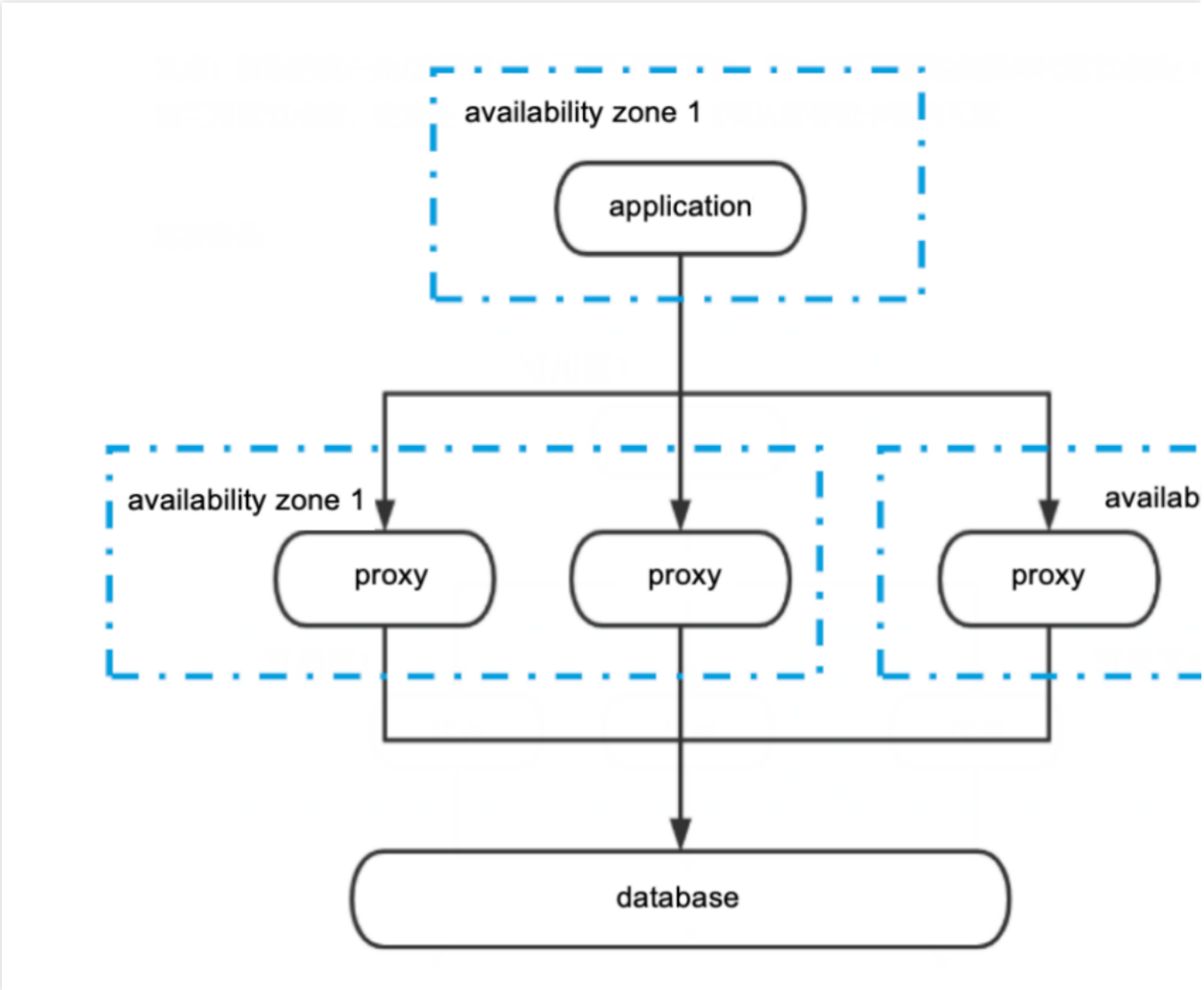
Prerequisites

You have enabled database proxy. For details, see Enabling Database Proxy.

Overview

There are two access modes available to manage the connection between the client application and the database: **Balanced Distribution** and **Nearby Access**. The following sections detail the rules, advantages, and disadvantages of the two access modes.

Balanced Distribution

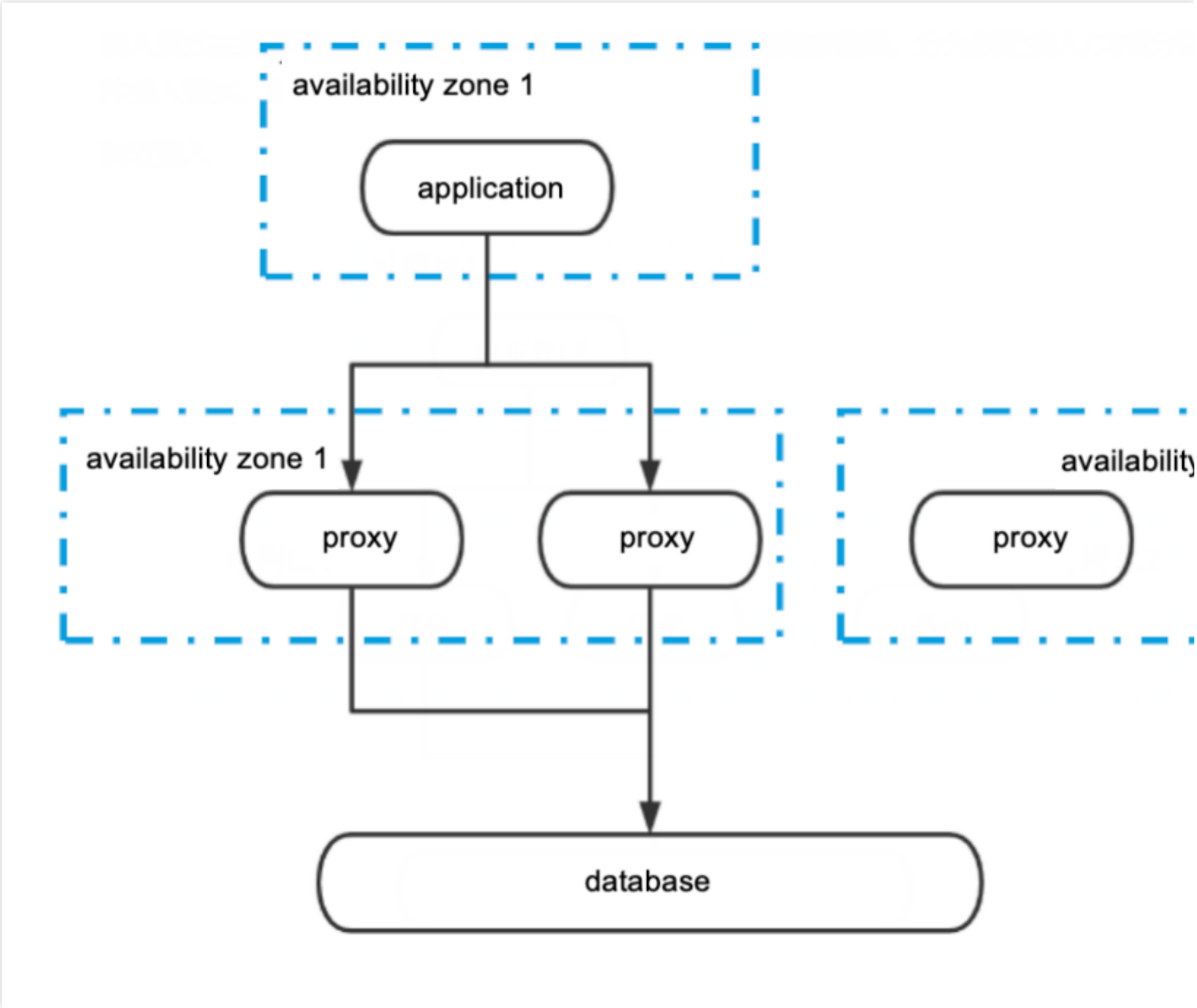


Rule: In the Balanced Distribution access mode, the client application connects to all available database proxy nodes. If the database proxy nodes are spread across multiple availability zones, application connections to the database are evenly distributed among all availability zone nodes.

Advantage: Traffic is evenly distributed, resolving the issue of excessive load on a single node.

Disadvantage: If there are proxy nodes in different availability zones, high request latency is prone to occur due to extended access paths.

Nearby Access



Rule: In the Nearby Access mode, the application will connect to the proxy node in the same availability zone or the one with the shortest access path. Even if the database proxy nodes are spread across multiple availability zones, the application will still connect the proxy node with the shortest access path.

Advantage: Low access latency and high speed.

Disadvantage: If your client application changes to an availability zone with fewer database proxy nodes than the original zone, it can lead to excessive load on the nodes and performance delays.

Changing Access Mode

The access mode can be changed. For details, see Viewing and Changing the Access Policy.

Adjusting Database Proxy Configuration

Last updated：2024-03-07 10:27:16

You can adjust the specification and node count of the database proxy in the console.

Prerequisite

You have enabled the database proxy. For more information, see Enabling Database Proxy.

Notes

The database proxy will be automatically upgraded to the latest version during configuration adjustment if its version is old.
If the selected database proxy does not reside in the same availability zone with the primary instance, a potential decline in write performance may be observed during the connection via the database proxy.
If the required number of proxy nodes exceed the purchase limit upon calculation, it is advisable to choose a higher specification proxy.
When the proxy specification is 2 cores and 4000MB of memory, the recommended number of proxies is 1/8th (rounded up) of the sum of the CPU core numbers of the main instance and the read-only instance.
When the proxy specification is 4 cores and 8000MB of memory, the recommended number of proxies is 1/16th (rounded up) of the sum of the CPU core numbers of the main instance and the read-only instance.
When the proxy specification is 8 cores and 16000MB of memory, the recommended number of proxies is 1/32 (rounded up) of the sum of the CPU core numbers of the main instance and the read-only instance.

Impact Description

Changing different configuration items has different impacts, certain modifications will not result in connection interruptions, while others may cause such disruptions. The specific changeable items and their corresponding impacts are detailed in the following table.

Scenario One: Changed Proxy Specification and Unchanged Availability Zone and Node Count

Proxy Specification	Availability Zone	Number of Nodes	Switch Time	Impact
Upgrade or downgrade	Unchanged	Unchanged	During the maintenance	There may be momentary interruptions. Please ensure your operations are equipped

			period	with a reconnection mechanism.	
			Upon completion of the upgrade		

Scenario Two: Changed Node Count and Unchanged Proxy Specification and Availability Zone

Proxy Specification	Availability Zone	Number of Nodes	Load Balancing Method	Switch Time	Impact
Unchanged	Unchanged	Raised	Automatic	During the maintenance period	There may be momentary interruptions. Please ensure your operations are equipped with a reconnection mechanism.
				Upon completion of the upgrade	
			Manual	During the maintenance period	It will not result in intermittent interruptions.
				Upon completion of the upgrade	
		Reduced	Automatic	During the maintenance period	Momentary interruptions may occur. Please ensure your operations are equipped with a reconnection mechanism.
				Upon completion of the upgrade	
			Manual	During the maintenance period	
				Upon completion	

				of the upgrade	

Scenario Three: Unchanged Proxy Specification and Availability Zone and Node Count Combination Changed

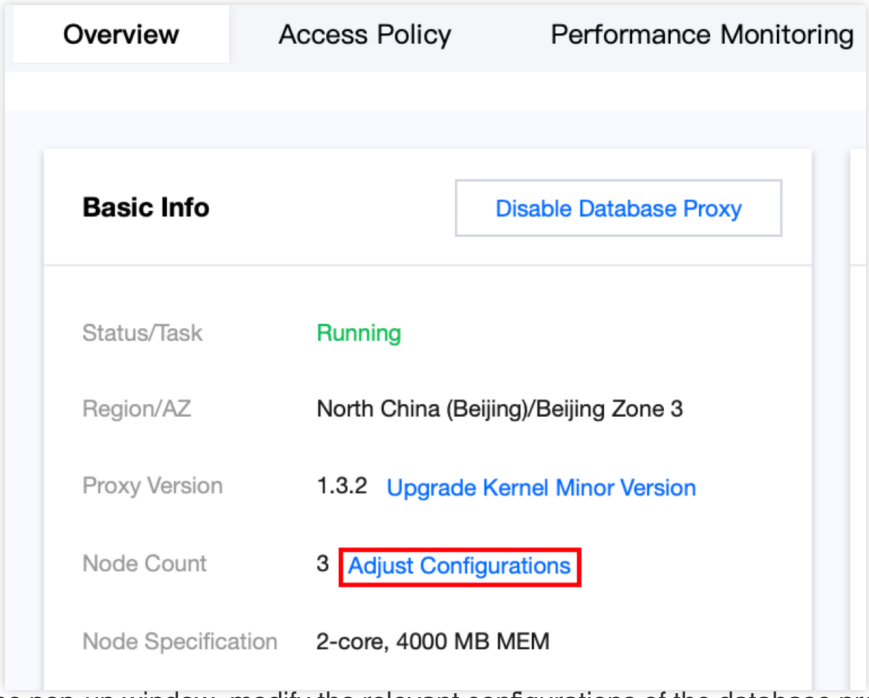
Proxy Specification	Availability Zone	Number of Nodes	Load Balancing Method	Switch Time	Impact
Unchanged	Raised	Raised	Automatic	During the maintenance period	There may be momentary interruptions. Please ensure your operations are equipped with a reconnection mechanism.
				Upon completion of the upgrade	
			Manual	During the maintenance period	It will not result in intermittent interruptions.
				Upon completion of the upgrade	
		Reduced	Automatic	During the maintenance period	Momentary interruptions may occur. Please ensure your business is equipped with a reconnection mechanism.
				Upon completion of the upgrade	
			Manual	During the maintenance period	
				Upon completion of the upgrade	
	Reduced	Raised	Automatic	During the	

				maintenance period
				Upon completion of the upgrade
			Manual	During the maintenance period
				Upon completion of the upgrade
		Reduced	Automatic	During the maintenance period
				Upon completion of the upgrade
			Manual	During the maintenance period
				Upon completion of the upgrade
	Changed	Raised	Automatic	During the maintenance period
				Upon completion of the upgrade
			Manual	During the maintenance period

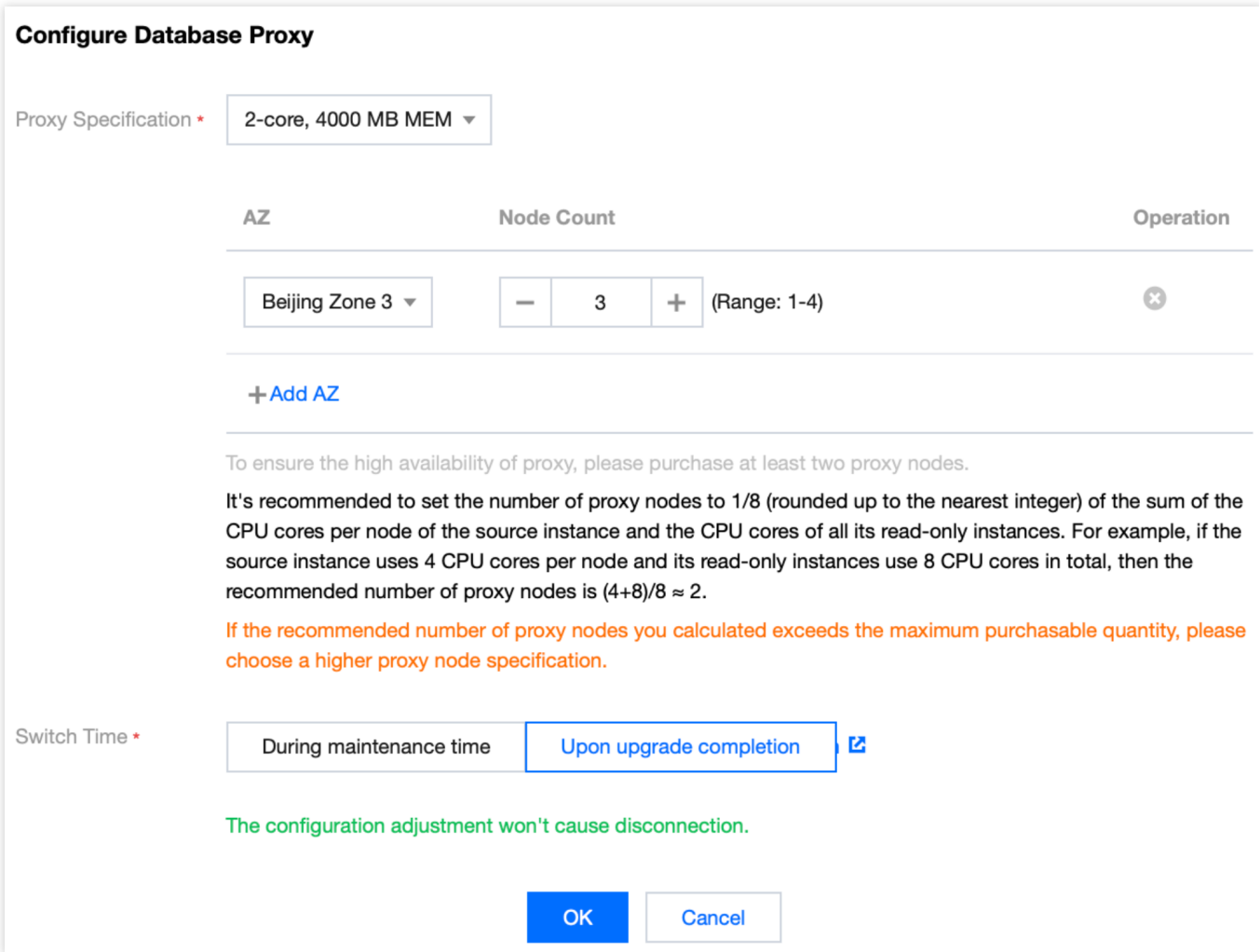
				Upon completion of the upgrade	
		Reduced	Automatic	During the maintenance period	
				Upon completion of the upgrade	
			Manual	During the maintenance period	
				Upon completion of the upgrade	

Directions

1. Log in to the TDSQL-C for MySQL console. In the cluster list, select the cluster with the proxy enabled, click on the cluster ID, and proceed to the cluster management page.
2. Navigate to the **Cluster Management** page > **Cluster Details** > **Database Proxy**, and click **Details**, or directly select the **Database Proxy** tab on the **Cluster Management** page to access the **Database Proxy Management** page.
3. On the **Overview** tab on the **Database Proxy** tab, Click **Adjust Configuration** after **Basic Information** > **Number of Nodes**. The output content is in markdown format.



4. In the pop-up window, modify the relevant configurations of the database proxy (proxy specification, availability zone, and number of nodes) as needed, then click **Confirm**.



5. In **Basic Info** on the **Database Proxy** tab, after the task status changes from **Upgrading** to **Running/Waiting to upgrade**, the configuration adjustment is completed.

Note:

If you select **Upon upgrade completion**, after the configuration is adjusted, the system will automatically switch to the new configuration.

If you select **During maintenance time**, after the configuration is adjusted, the system will switch the configuration during the specified maintenance time period.

If you select **During maintenance time** but need to switch the configuration earlier due to your business requirements, after the configuration is adjusted, you can go to **Database Proxy** > **Overview** > **Basic Info** > **Status/Task** and click **Execute now** next to **Running/Waiting to upgrade**.

If the CLB resettings are performed manually after configuration adjustment, once the configuration adjustment is complete, one can manually perform CLB resettings under the corresponding **Overview > Connection Address** on the database proxy page.

Switching Database Proxy Network

Last updated：2024-03-07 10:28:35

This document describes how to modify the database proxy network in the TDSQL-C for MySQL console.

Prerequisite

You have enabled the database proxy. For more information, see Enabling Database Proxy.

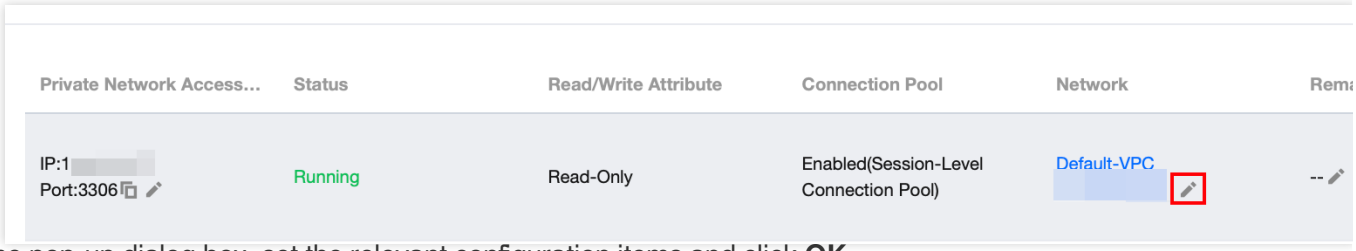
Notes

Changing the network may cause the change of the instance's database proxy IP. The old IP will be retained for 24 hours by default and up to 168 hours. Then, it will become invalid. Therefore, modify the IP on the client promptly.
If **Valid Hours of Old IP** is set to 0 hours, the IP is released immediately after the network is changed.
You can select only a VPC in the region where the cluster of TDSQL-C for MySQL resides, but you can choose a subnet in any AZ and view its IP range.

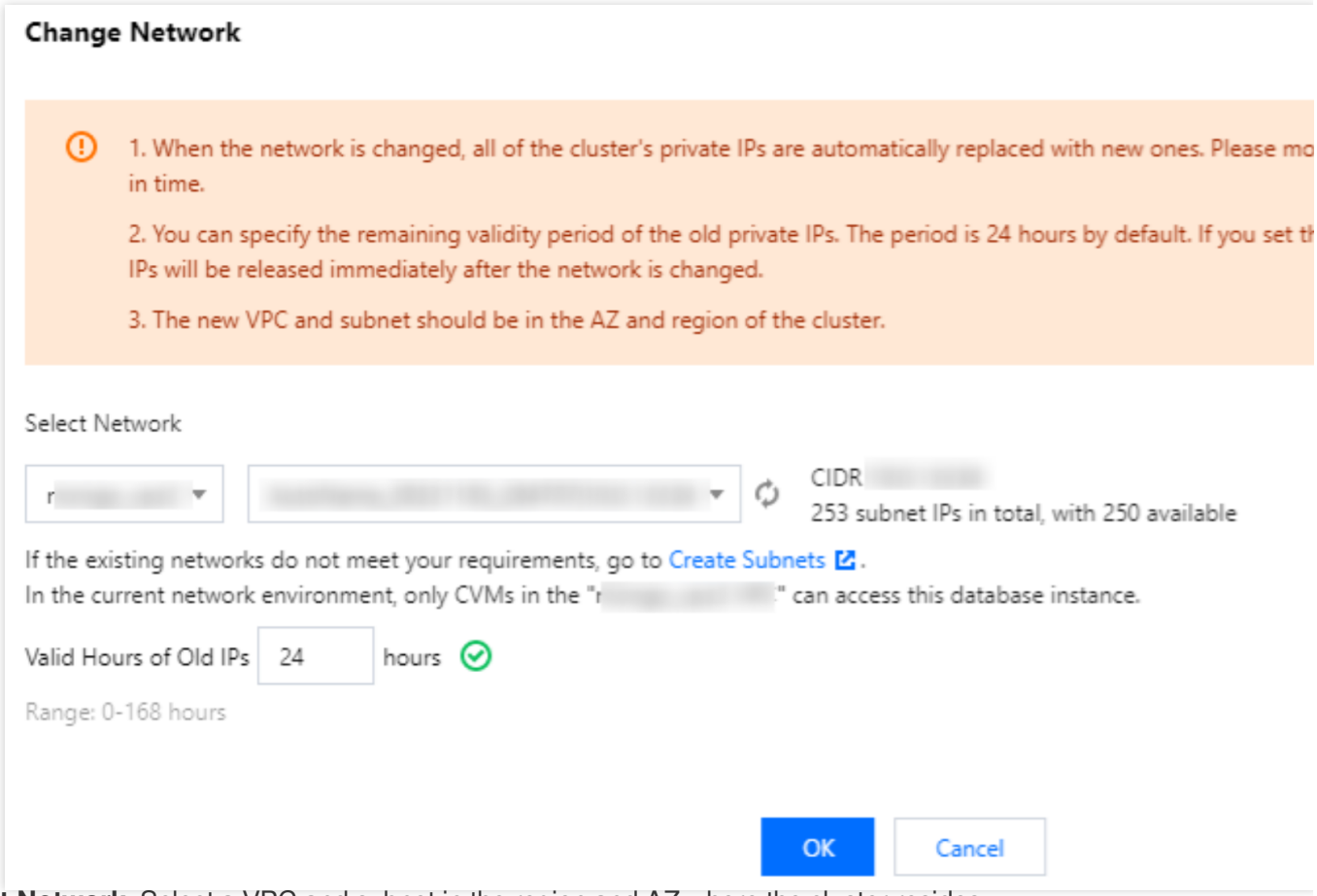
Directions

1. Log in to the TDSQL-C for MySQL console, select the region at the top of the page, and click the ID of the target cluster to enter the cluster management page.
2. Navigate to the **Cluster Management** page > **Cluster Details** > **Database Proxy**, and click **Details**, or directly select the **Database Proxy** tab on the **Cluster Management** page to access the **Database Proxy Management** page.
3. On the **Database Proxy** tab, click

under **Overview** > **Connection Address** > **Network**.



4. In the pop-up dialog box, set the relevant configuration items and click **OK**.



Select Network: Select a VPC and subnet in the region and AZ where the cluster resides.

Valid Hours of Old IP: Its range is 0–168 hours and default value is 24 hours.

5. After successfully changing the network, you can view the new network under **Connection Address**.

Viewing Database Proxy Monitoring Data

Last updated：2024-06-17 16:14:14

This document introduces how to view database proxy node monitoring data in the TDSQL-C for MySQL console.

Prerequisite

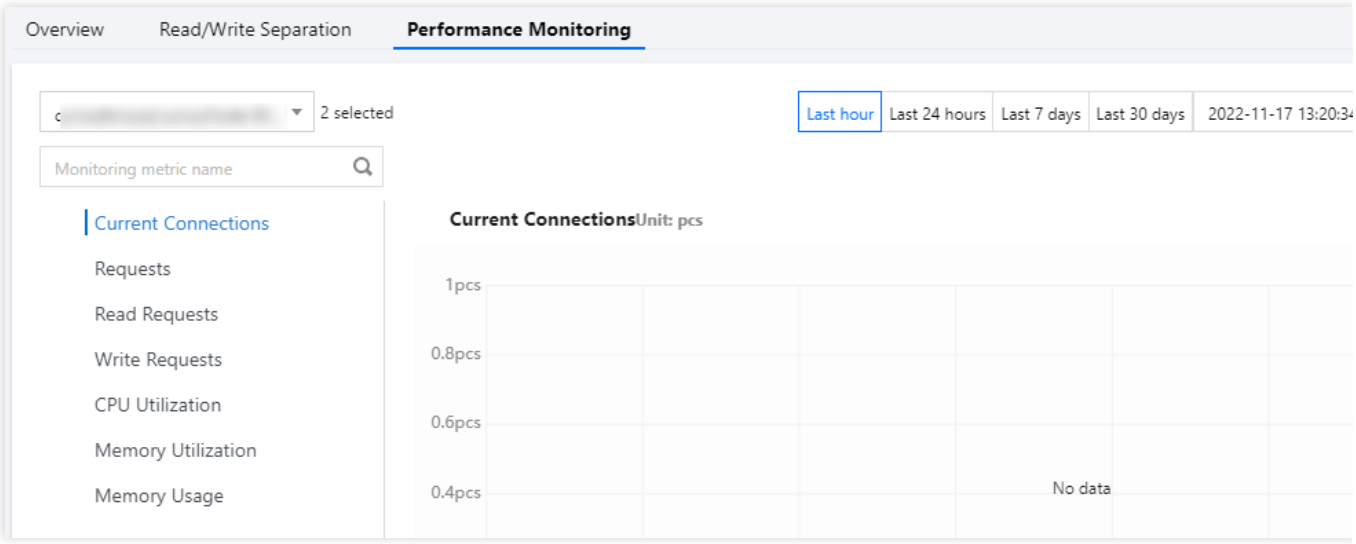
You have enabled the database proxy. For more information, see Enabling Database Proxy.

Database Proxy Node Monitoring Metrics

Monitoring Metrics (Chinese)	Monitoring Metrics (English)	Unit
Current Connections	Current Connections	Individual
Requests	Requests	Times/sec
Read Requests	Read Requests	Times/sec
Write Requests	Write Requests	Times/sec
CPU Utilization	CPU Utilization	%
Memory Utilization	Memory Utilization	%
Memory Usage	Memory Usage	MB
Proxy-to-Client Traffic per Second	Proxy-to-Client Traffic per Second	MB/s
Client-to-Proxy Traffic per Second	Client-to-Proxy Traffic per Second	MB/s

Directions

1. Log in to the TDSQL-C for MySQL console. In the cluster list, click the ID or **Manage** in the **Operation** column of the cluster with the proxy enabled to enter the cluster management page.
2. On the cluster management page, choose **Cluster Details** > **Database Proxy**, click **Details** , or directly click the **Database Proxy** tab on the cluster management page to enter the database proxy management page.
3. On the database proxy page, click **Performance Monitoring** , and click the node name to switch and view the monitoring data of each proxy node.



Disabling Database Proxy

Last updated：2023-01-03 17:21:17

This document describes how to disable the database proxy in the TDSQL-C for MySQL console.

Note:

After the database proxy is disabled, the database proxy address will be unavailable, but the read-write instance address is still available.

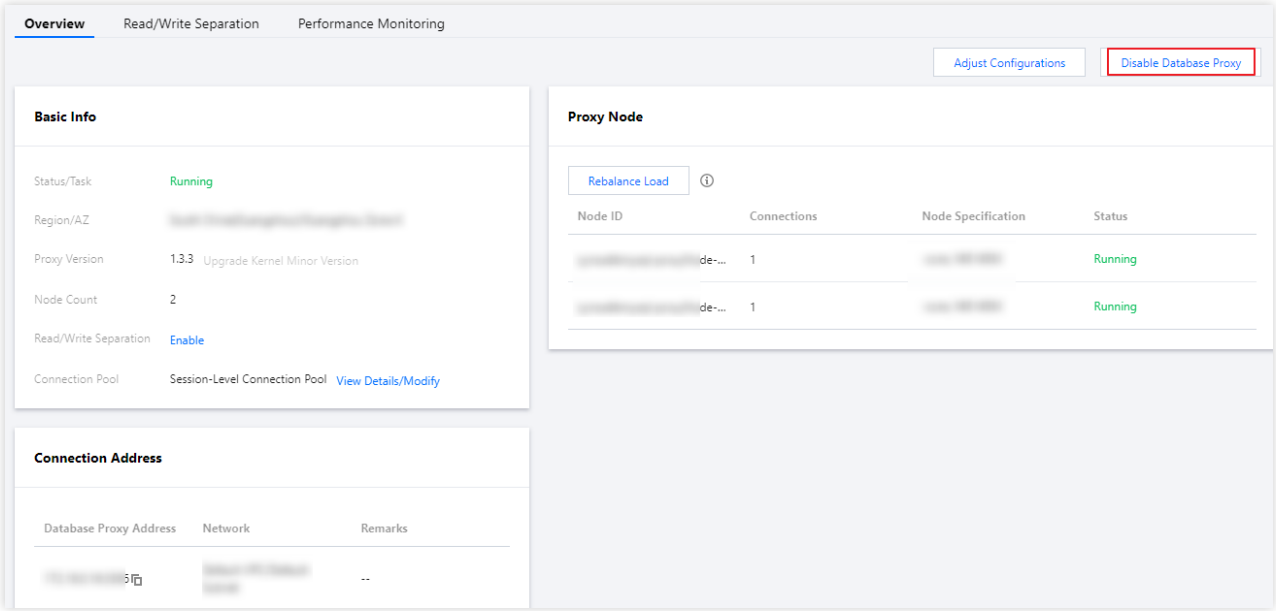
If you enable the database proxy again after disabling it, the database proxy access address will change.

Prerequisite

You have enabled the database proxy. For more information, see Enabling Database Proxy.

Directions

1. Log in to the TDSQL-C for MySQL console, and click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
2. On the cluster management page, select the **Database Proxy** tab and click **Disable Database Proxy** in the top-right corner.



3. In the pop-up dialog box, confirm that everything is correct and click **OK**.

Automatic Read/Write Separation Overview

Last updated：2023-09-12 14:48:55

This document describes the automatic read/write separation feature of the database proxy service in TDSQL-C for MySQL as well as its strengths and routing rules.

Automatic read/write separation

Currently, businesses of many users in the production environment have problems such as more reads and less writes and unpredictable business loads. In application scenarios with a large number of read requests, a single instance may not be able to withstand the pressure of read requests, which even may affect the businesses. To implement the auto scaling of read capabilities and mitigate the pressure on the database, you can create one or multiple read-only instances and use them to sustain high numbers of database reads. However, this solution requires that businesses can be transformed to support read/write separation, and the code robustness determines the quality of business read/write separation, which imposes high technical requirements and has low flexibility and scalability.
Therefore, after creating a read-only instance, you can enable a database proxy, configure access address policy, and configure the database proxy address in your application so as to automatically forward write requests to the source instance and read requests to the read-only instance. In addition to automatic read/write separation, the database proxy also provides better solutions to other business challenges as detailed below:
Scenarios where the load is unpredictable or fluctuates irregularly with obvious peaks and troughs
In internet business scenarios, business load and access pressure are often unpredictable and unstable, and there will be frequent great fluctuations. If the business uses a large number of non-persistent connections to access the database, it is easy to generate many new connections. In other words, the number of connections between the database and the application is likely to fluctuate as the business access pressure changes frequently.
Connection management for the database proxy allows you to efficiently reuse database connections to appropriately scale applications that handle unpredictable workloads. First, this feature allows multiple application connections to share the same database connection to effectively use database resources. Second, it allows you to adjust the number of open database connections to maintain database performance. Finally, it allows you to delete unusable application requests to guarantee the overall application performance and availability.
Scenarios where the application is frequently connected to and disconnected from the database
Applications built based on technologies such as serverless, PHP, or Ruby on Rails may frequently open and close database connections to process application requests.
The database proxy can help you maintain a database connection pool to prevent unnecessary pressure on data computing and the memory used to establish new connections.

Scenarios where the database access connection is idle for a long time and is not released
SaaS applications and traditional ecommerce applications may make database connections idle to minimize the response time for user reconnection. You can use the database proxy to retain idle connections and establish database connections as needed instead of excessively increasing the threshold or upgrading to higher database specifications to support most idle connections.
Scenarios where you want to improve the smoothness and stability of database PaaS service failover
With the database proxy, you can build applications that can tolerate active and passive database failures in an imperceptible manner with no need to write complex failure processing code. The database proxy will automatically route read traffic to new database instances while retaining the application connections.

Advantages

Read/write requests are automatically separated with a unified access address.
Native linkage support for improving the performance and reducing the maintenance costs.
You can flexibly set weights and thresholds.
Failover is supported, so that even if the database proxy fails, requests can access the source database normally.
When the read-write instance is switched, or its configuration is changed, or a read-only instance is added/removed, the database proxy can dynamically hot reload the configuration without causing network disconnections or restarts.

Read/Write Separation Routing Rules

Sending to the read-write instance

DDL statements such as `CREATE` , `ALTER` , `DROP` , and `RENAME` .
DML statements such as `INSERT` , `UPDATE` , and `DELETE` .
`SELECT FOR UPDATE` statement.
Statements related to temp tables.
Certain system function calls (such as `last_insert_id()`) and all custom function calls.
Statements related to `LOCK` .
Statements after transaction is enabled (including `set autocommit=0`)
Stored procedures.
Multiple statements concatenated by ";".
`KILL` (SQL statement, not command)
All queries and changes of user variables.

Sending to the read-only instance

Read (`SELECT`) statements outside transactions.

Sending to all instances

`show processlist` statement.

All changes of system variables (`SET` command).

`USE` command.

Enabling Database Proxy Read/Write Separation

Last updated：2023-01-03 17:21:17

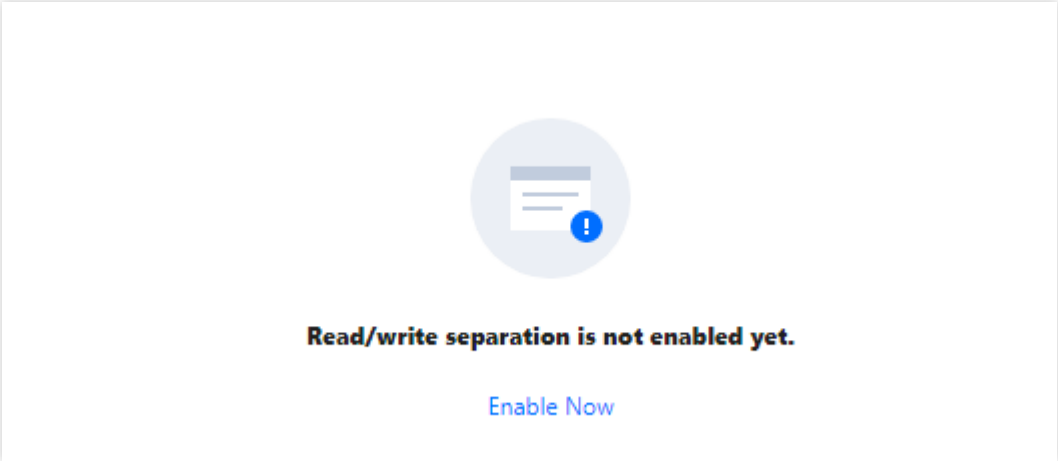
With the database proxy read/write separation feature, you can configure the database proxy address in your application, so that write requests are automatically forwarded to the read-write instance and read requests to each read-only instance.
This document describes how to enable/disable the read/write separation feature of TDSQL-C for MySQL.

Prerequisite

The instance is the read-write instance.
You have enabled the database proxy. For more information, see Enabling Database Proxy.
You have created a read-only instance. For more information, see Creating Read-Only Instance.

Enabling Read/Write Separation

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
3. On the **Database Proxy** tab on the cluster management page, select the **Read/Write Separation** tab and click **Enable Now**.



4. In the pop-up window, set the read/write configuration items and click **OK**.

Note:

Only the running read-write instance and read-only instances can be added to the database proxy.

Currently, remote or delayed read-only instances cannot be mounted to the database proxy.

Consistency Settings: Three consistency levels are available, that is, eventual consistency, session consistency, and global consistency.

Assign Read Weight: Assign read weights to instances. You can select **Assigned by system** or **Custom**. Each weight must be an integer between 0 and 100. The configured read weight assignment takes effect immediately for all connections.

The database proxy will assign read request traffic according to the set weights; for example, if weights of two read-only instances are 10 and 20 respectively, their read request traffic will be assigned at a ratio of 1:2.

The weight here refers to the read request weight only, as write requests are directly routed to the source database without participating in weight calculation; for example, if a client sends 10 write statements and 10 read statements, and the ratio of the source and read-only instance weights is 1:1, the read-write instance will receive 10 write statements and 5 read statements, and the read-only instance will receive 5 read statements only.

If you select **Assigned by system**, the system will automatically assign weights based on the instance's CPU and memory specification, and you can only set the weight of the read-write instance in this case.

If the weight of a read-only instance is 0, the database proxy will not connect to the instance. If its weight is changed from 0 to another value, the weight takes effect only for new connections.

Failover: Specifies whether to enable failover. We recommend you enable this option, so the database proxy can send read requests to the read-write instance if all read-only instances are exceptional.

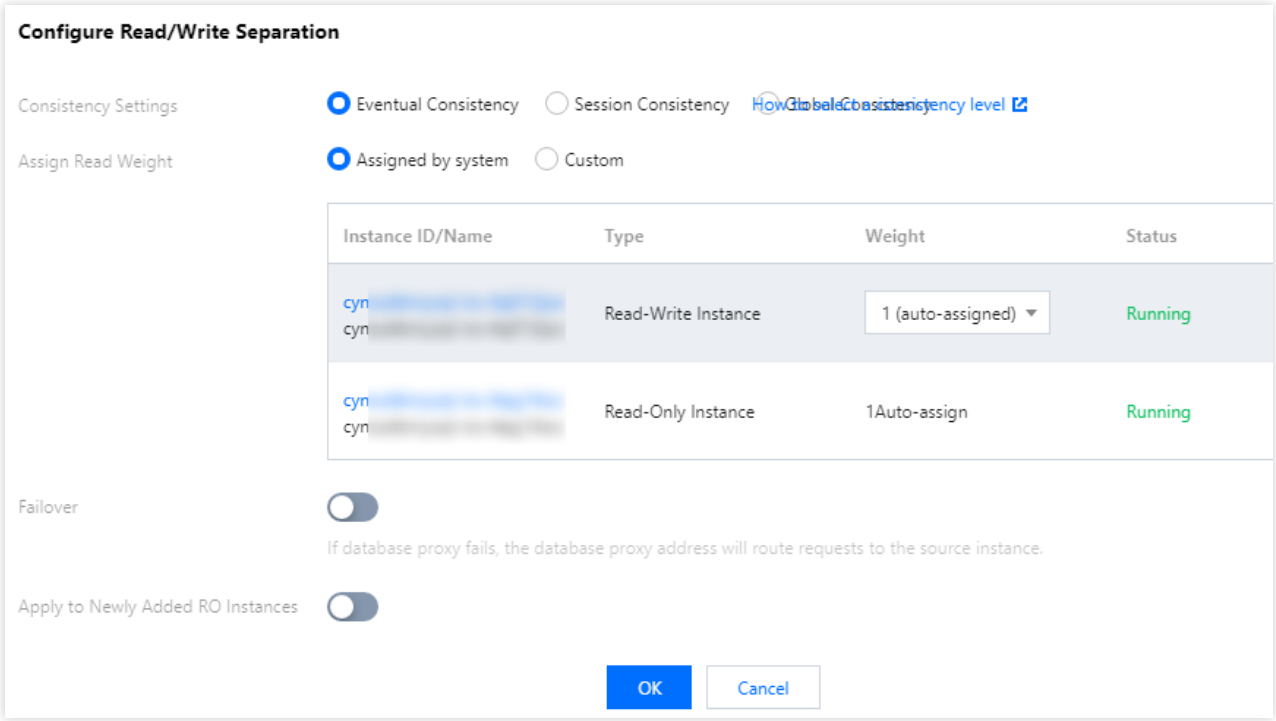
Note:

The configured failover capability takes effect only for new connections.

Apply to Newly Added RO Instances: Enable or disable this parameter. After it is enabled, if you purchase new read-only instances, they will be automatically added to the database proxy.

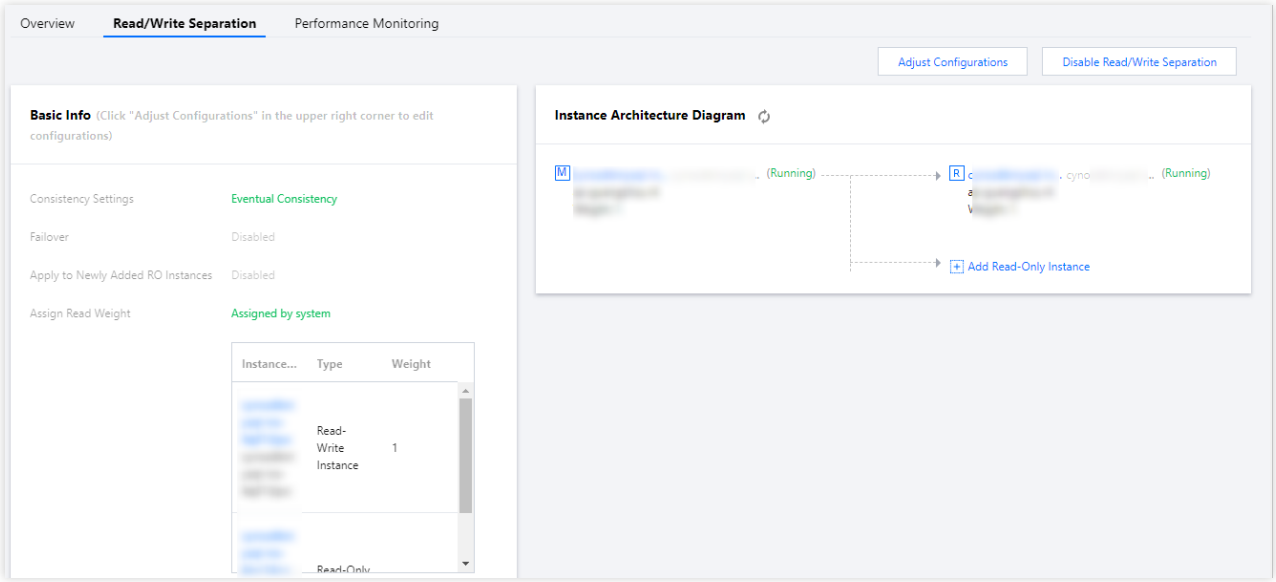
If **Assign Read Weight** is set to **Assigned by system**, newly purchased read-only instances will be assigned with the default weight based on their specification.

If **Assign Read Weight** is set to **Custom**, when newly purchased read-only instances are added to the RO group, their weights will be 0 by default, which can be modified in the configuration of the database proxy's read/write separation.



Page display

After the database proxy's read/write separation feature is enabled, on the **Read/Write Separation** tab, you can view the basic information and read/write separation architecture diagram. You can also click buttons on the right to adjust the configuration and disable the feature.



Disabling Read/Write Separation

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
3. On the **Database Proxy** tab on the cluster management page, select the **Read/Write Separation** tab and click **Disable Read/Write Separation** on the right.

Consistency Level

Last updated：2023-01-03 17:21:17

With the automatic read/write separation feature of TDSQL-C for MySQL, a connection will be established between TDSQL-C for MySQL and the application to parse every incoming SQL statement. If it is a statement such as `CREATE`, `ALTER`, `DROP`, or `RENAME`, the statement is directly sent to the read-write instance. If it is a non-transactional read (`SELECT`) statement, the statement is sent to a read-only instance, thereby implementing read/write separation. However, when the database load is high, for example, when a large batch of data is inserted, the delay will be long, resulting in a failure to read the latest data from the read-only instance.
When there is data update in the read-write instance, the related update will be applied to the read-only instance. The delay time of data sync is subject to the write workload. TDSQL-C for MySQL provides different consistency levels to ensure data consistency requirements when the business accesses the database.
The consistency levels are as follows:
Eventual consistency
Session consistency
Global consistency

Eventual Consistency

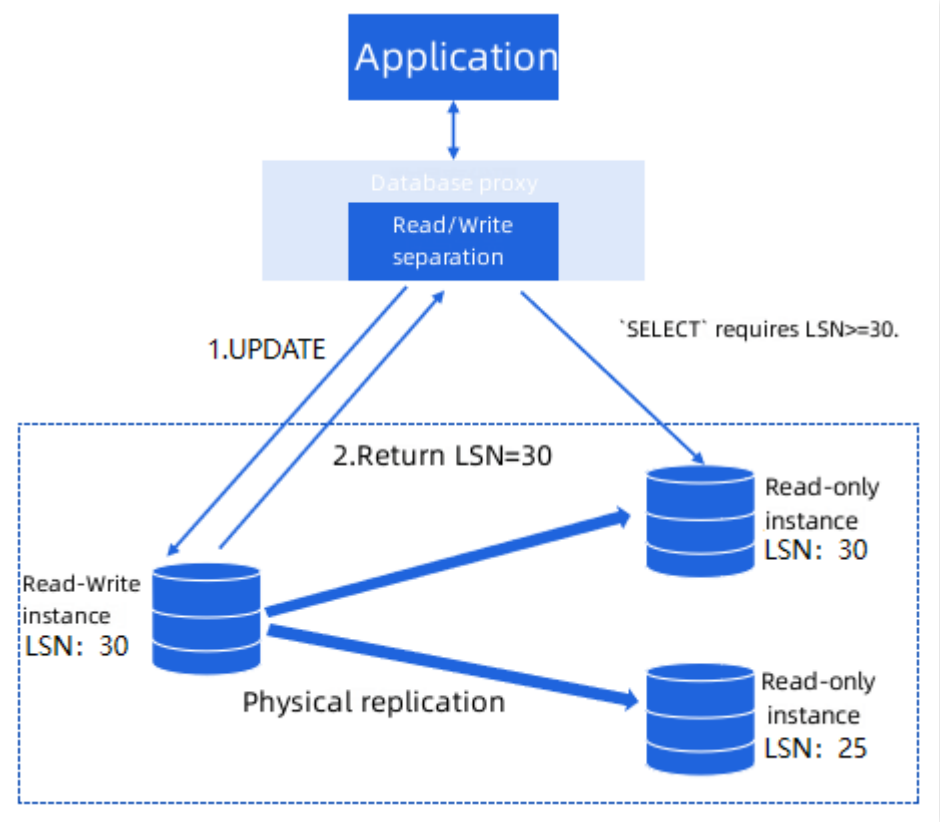
Feature overview
The database proxy of TDSQL-C for MySQL implements the automatic read/write separation feature. In the automatic read/write separation scenario, the eventual consistency is provided by default to ensure that the read-only instance can read the newly written data and eventually obtain all of them. However, the database proxy does not ensure that the updated data is immediately available. Delays in source/replica replication of the updated data will cause inconsistent results queried from different nodes.
Use cases
If you need to reduce the pressure on the read-write instance and route as many read requests as possible to the read-only instance, you can choose eventual consistency in scenarios that do not require high consistency.

Session Consistency

Feature overview
Some scenarios require high consistency, but eventual consistency causes incosistent query results. Usually, the business needs to be split, that is, requests with high consistency requirements are directly sent to the read-write instance while requests that can accept eventual consistency are sent to the read-only instance through read/write

separation. However, this will increase the pressure on the read-write instance, reduce the effect of read/write separation, and increase the burden of application development.

To solve the above problems, TDSQL-C for MySQL provides session consistency to guarantee monotonic reads within the same session, and data that has been updated before the read requests are executed can be queried. While the middle layer of the linkage of TDSQL-C for MySQL performs read/write separation, the middle layer traces log time points (that is, log sequence number, LSN for short) that have been applied by each node. When data is updated each time, TDSQL-C for MySQL records the updated time point as session LSN. When a new request arrives, TDSQL-C for MySQL compares the session LSN with the LSN of each current instance, and sends the request only to the instance whose LSN is greater than or equal to the session LSN, thereby ensuring session consistency.



In the above scenario, when the update is completed, replication is also synchronously performed while the result is returned to the client. When the next read request arrives, data replication between the read-write instance and the read-only instance may have been completed. In addition, most application scenarios feature more reads and less writes, so session consistency is guaranteed under this mechanism.

Use cases

It is suitable for scenarios with high consistency requirements. A higher consistency level of TDSQL-C for MySQL

indicates greater pressure on the primary database and lower cluster performance. Session consistency is recommended because it has little impact on performance and can meet the needs of most application scenarios.

Note:

After session consistency is enabled, if the replication delay between the read-write instance and the read-only instance is large, and the LSN of each read node is smaller than the session LSN, SELECT requests will be sent to the read-write instance. This will increase the pressure on the read-write instance and reduce the read/write performance of the entire cluster to a certain extent.

Global Consistency

Feature overview

Some scenarios have extremely high consistency requirements. In addition to logical causal dependencies within sessions, there are dependencies between sessions. For example, in scenarios where connection pools are used, requests from the same thread may be sent through different connections. For the database, these requests belong to different sessions. However, in terms of business logic, these requests have pre- and post-dependencies. In this case, session consistency cannot ensure consistency of query results. Therefore, TDSQL-C for MySQL provides global consistency to solve this problem.

Use cases

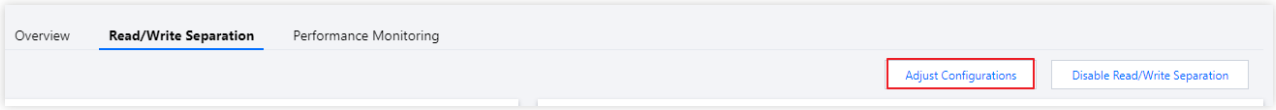
It is suitable for scenarios with extremely high consistency requirements. When the source-replica delay is high, global consistency may cause more requests to be routed to the read-write instance, thereby increasing the pressure on the read-write instance and prolonging the business delay. Therefore, it is recommended to choose global consistency in scenarios that feature more read requests and less write requests.

Setting a Consistency Level

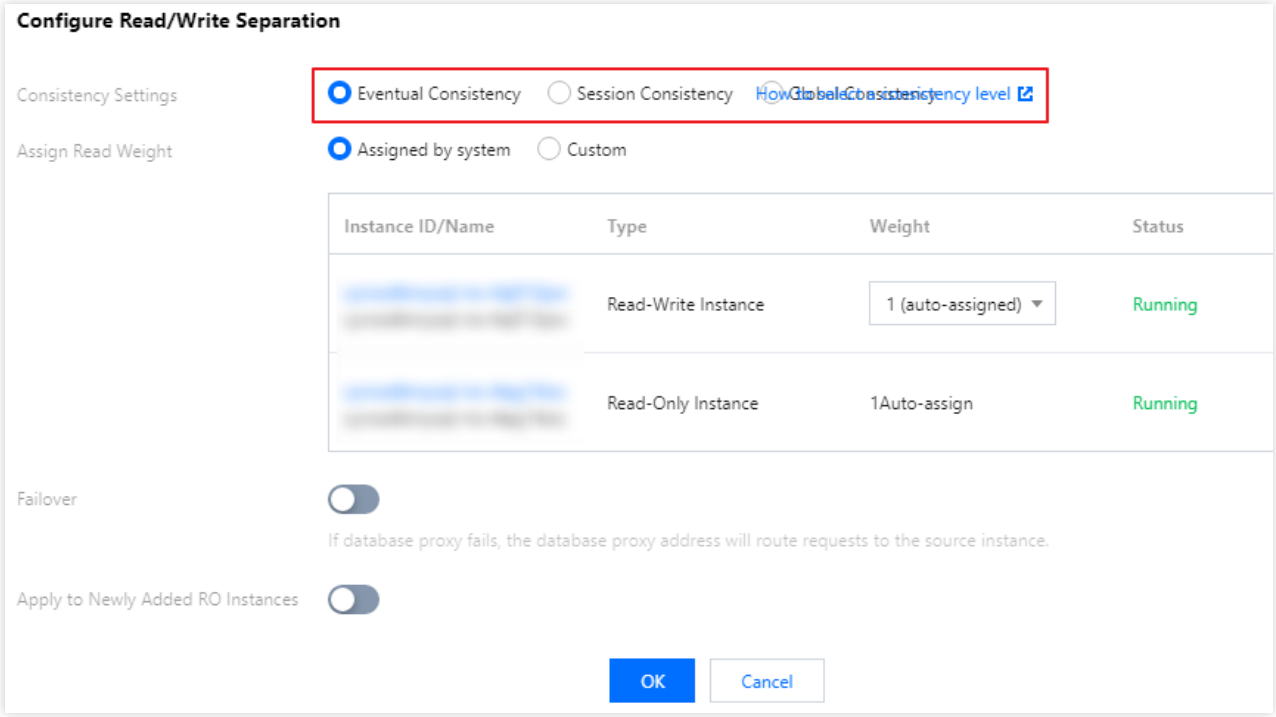
Note:

Before setting a consistency level, enable the read/write separation feature. You can set the session consistency level in the step of enabling read/write separation. If the feature has been enabled, but the consistency level needs to be modified, refer to the following steps.

1. Log in to the TDSQL-C for MySQL console.
2. In the cluster list, click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
3. On the cluster management page, select **Database Proxy** > **Read/Write Separation** > **Adjust Configurations*.



4. In the configuration window, select a consistency level and click **OK**.



Connection Pool Feature
Connection Pool Overview

Last updated：2023-02-08 09:46:36

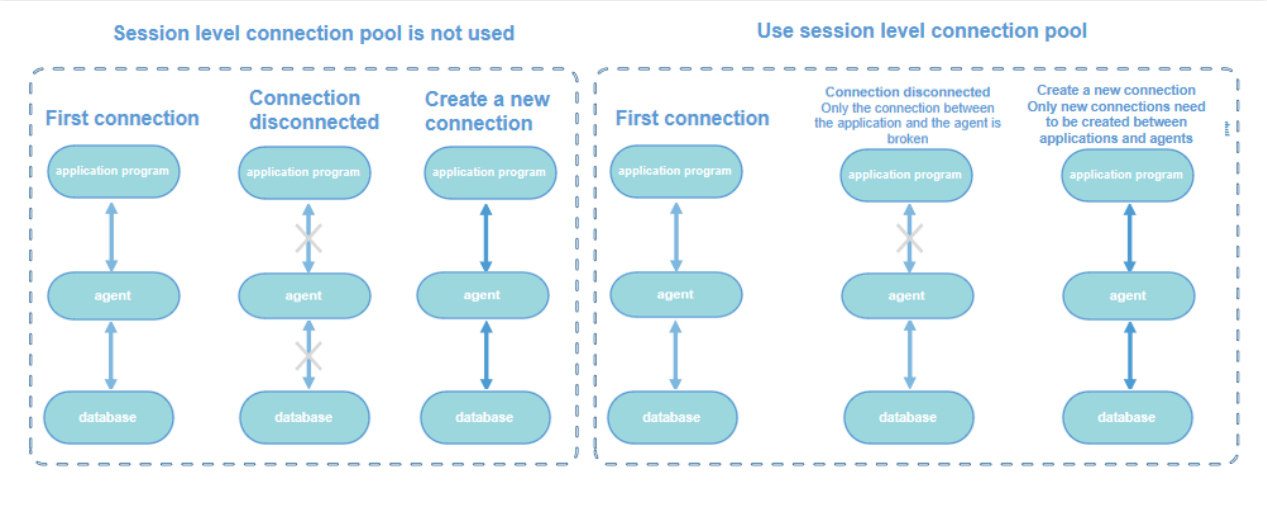
This document describes the connection pool feature. The TDSQL-C for MySQL database proxy supports the session-level connection pool feature. It can effectively solve the problem of excessively high database instance loads caused by frequent establishments of new non-persistent connections.

Prerequisites

You have enabled the database proxy. For more information, see Enabling Database Proxy.

Background

Session-level connection pool



The session-level connection pool is applicable to non-persistent connection scenarios.
It is used to reduce the instance load caused by frequent establishments of new non-persistent connections. If a client connection is closed, the system will determine whether the current connection is idle, and if so, the system will put it into the proxy connection pool and retain it for a short period of time (which is five seconds by default and can be customized as instructed in Setting Session-Level Connection Pool).

When the client initiates a new connection, if the connection pool has an available connection, the connection can be used directly to reduce the database connection overheads; otherwise, a new connection will be established as usual.

Note:

The session-level connection pool does not reduce the number of concurrent connections to the database, but it can reduce the overheads of the main thread of TDSQL-C for MySQL by lowering the speed for connecting the application to the database, so as to better process business requests. However, idle connections in the connection pool will occupy your connection quantity temporarily.

The session-level connection pool cannot be used to solve the problem of connection heap caused by a large number of slow SQL queries, which must first be solved by yourself.

Notes

Currently, the connection pool feature does not support setting different permissions for different IPs under the same account, as that may cause a permission error during connection reuse. For example, if `mt@test123` has `database_a` permissions, while `mt@test456` does not, a permission error may occur when you enable the connection pool.

The connection pool feature is for the database proxy instead of the client. You don't need to use the connection pool of the database proxy if your client already supports it.

Setting Session-Level Connection Pool

Last updated：2023-01-03 17:21:17

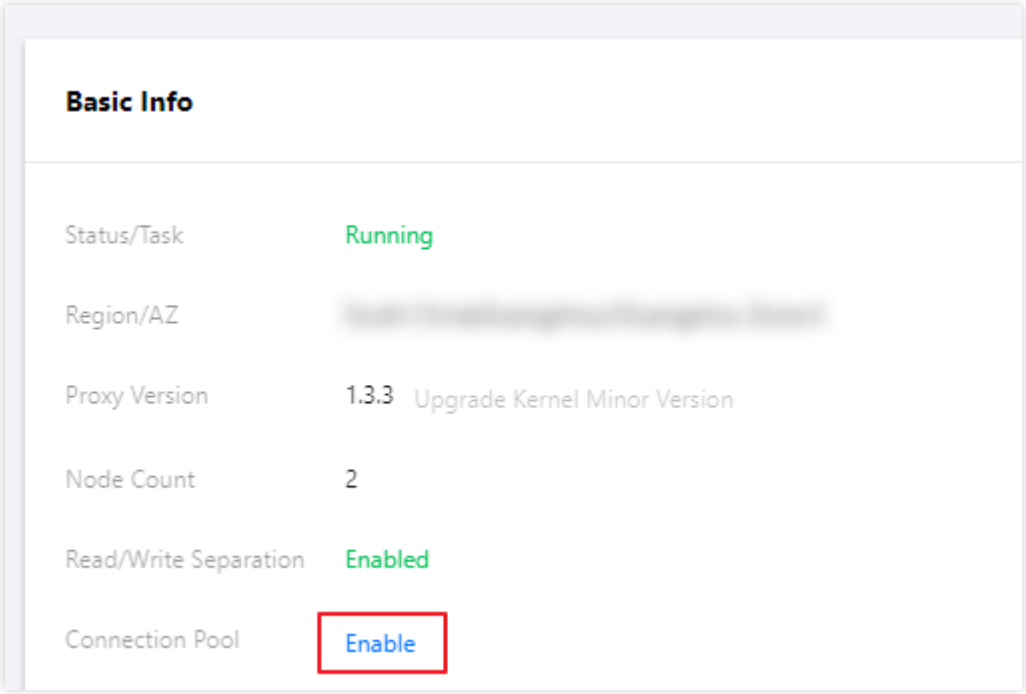
This document describes how to configure the session-level connection pool.

Database Proxy Disabled

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
3. On the **Database Proxy** tab on the cluster management page, click **Enable Now**.
4. In the pop-up dialog box, enable the connection pool feature.

Database Proxy Enabled

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
3. On the **Database Proxy** tab on the cluster management page, click **Enable** after **Connection Pool** at the bottom of the overview.



4. In the pop-up dialog box, set the relevant configuration items and click **OK**.

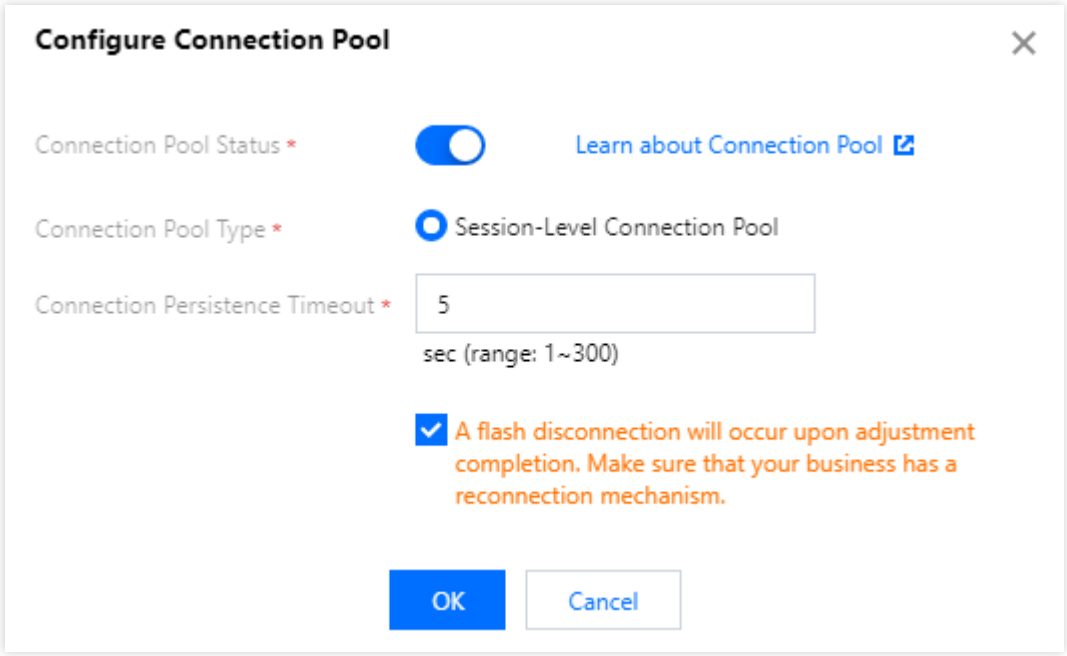
Connection Pool Status: Enable connection pool.

Connection Pool Type: Session-level connection pool is supported.

Connection Persistence Timeout: The time threshold for idle connections to be retained in the connection pool of the proxy, which can be 1–300 seconds.

Note:

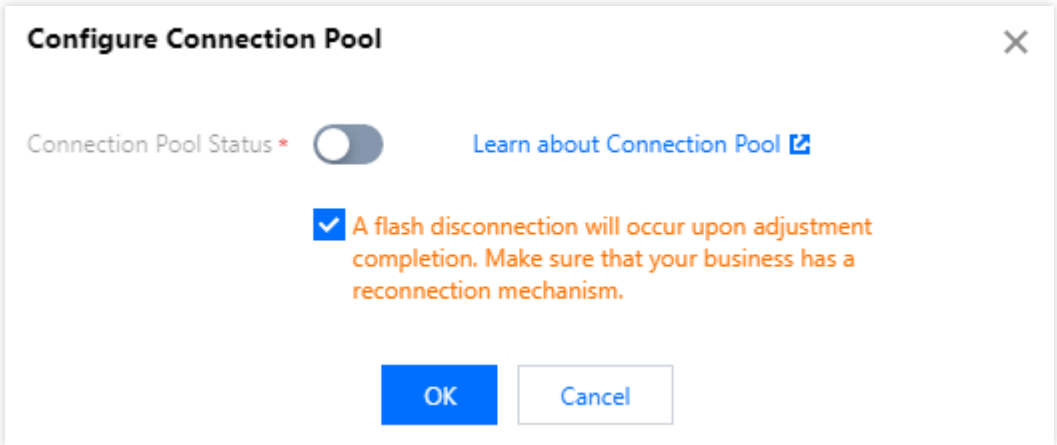
A flash disconnection will occur upon adjustment completion. Make sure that your business has a reconnection mechanism.



5. After the connection pool is successfully enabled, you can view the details and edit the configuration in **Overview** > **Basic Info** on the **Database Proxy** tab.

Disabling Connection Pool

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the ID or **Manage** in the **Operation** column of the target cluster to enter the cluster management page.
3. On the **Database Proxy** tab on the cluster management page, click **View/Edit** next to **Connection Pool** at the bottom of the overview.
4. In the pop-up dialog box, toggle off the **Connection Pool Status** option and click **OK**.



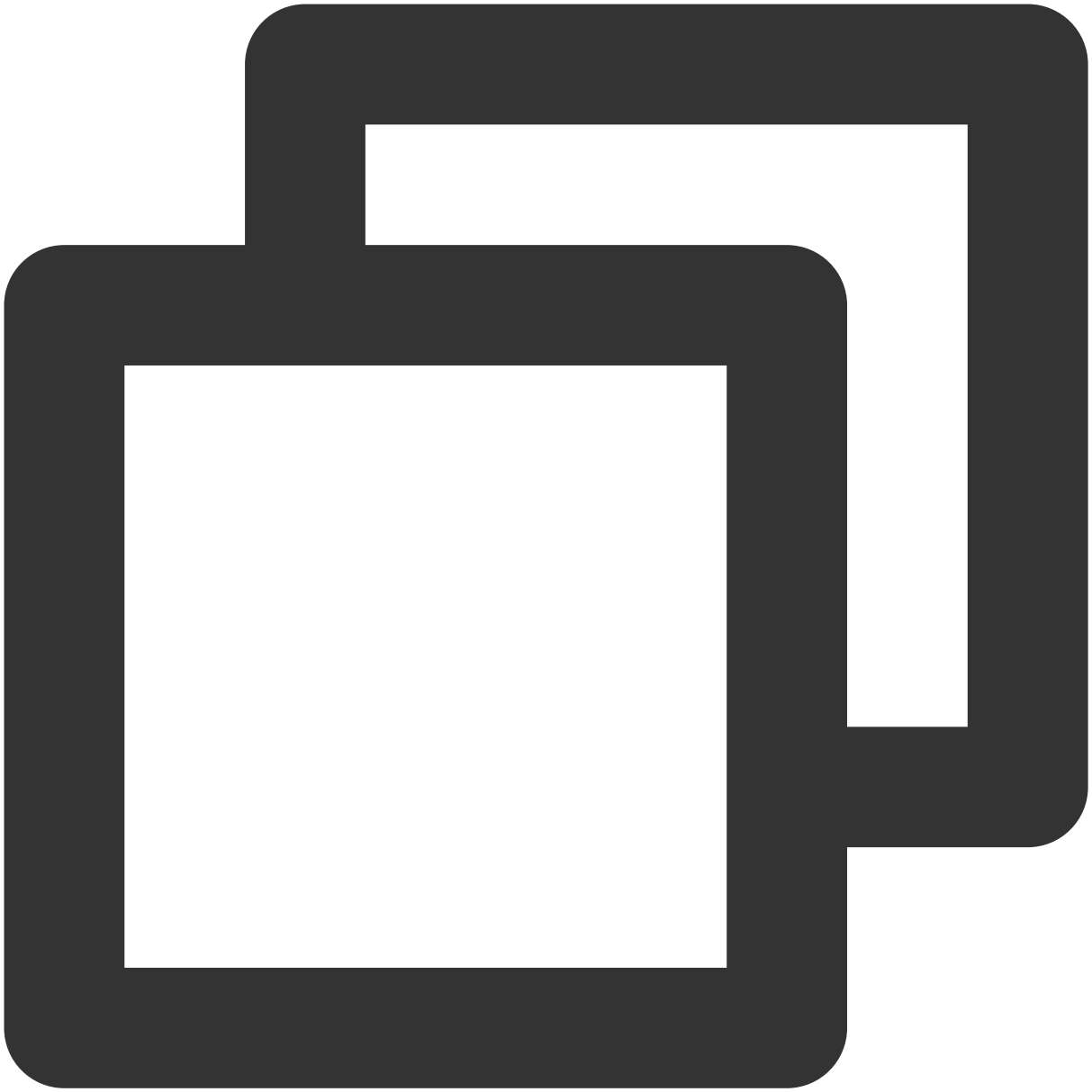
Other Features
Hint Syntax Usage

Last updated：2023-01-03 17:21:17

This document describes how to use the hint syntax on the database proxy.
The hint syntax can be used to forcibly execute SQL requests on the specified instance. A hint has the highest routing priority, and it is not subject to consistency and transaction constraints. You need to evaluate whether it is required in your business scenario before using it.

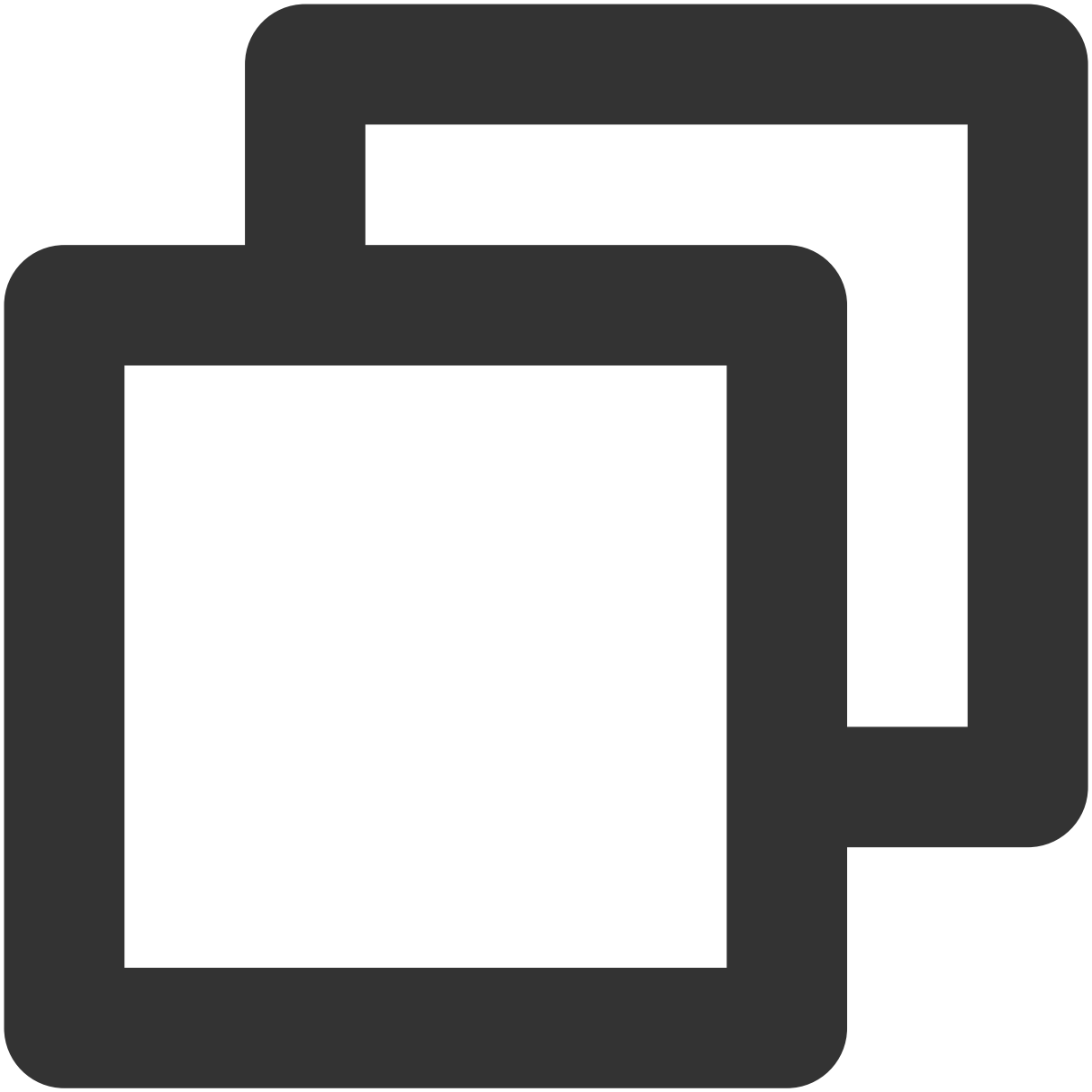
Note:

When using the TDSQL-C for MySQL command line tool to connect and use the `HINT` statement, you need to add the `-c` option in the command; otherwise, the hint will be filtered out by the tool.
If you use the hint syntax through the database proxy, `PREPARE` is currently not supported but will be in future versions.
Currently, three types of hints are supported:
Assign to the read-write instance for execution:



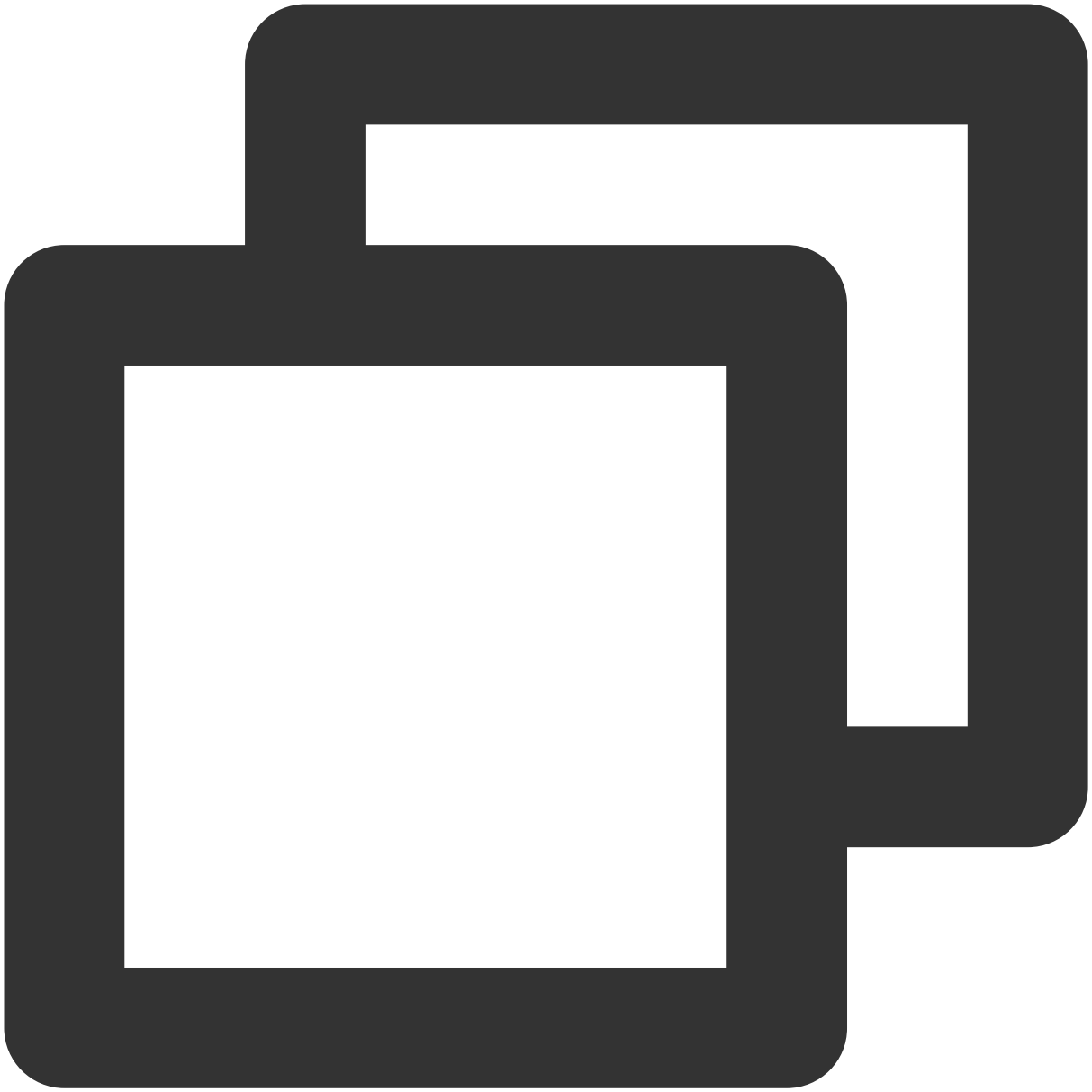
```
/* to master */
Or
/*FORCE_MASTER*/
```

Assign to a read-only instance for execution:



```
/* to slave */
Or
/*FORCE_SLAVE*/
```

Assign to a specified instance for execution:



```
/* to server server_name*/
```

`server_name` can be a short ID, such as `/* to server test_ro_1 */` .

Account Management
Account Overview

Last updated：2023-11-01 17:18:13

Before using the TDSQL-C for MySQL, you need to have 2 accounts, namely the Tencent Cloud console account and the database account. These two accounts are used to access the console and your database respectively. This document describes the information of the two accounts.

Tencent Cloud Console Account

Before using Tencent Cloud services, you need to register a Tencent Cloud account first. Then you can log in to the Tencent Cloud website or console with the account. For detailed directions, see Account Registration.

Database Account

When TDSQL-C for MySQL cluster is created, the system will retain the default account. In addition, you can also create a business account as needed to allocate and manage databases conveniently.

Default account

Default System Account	Host	MySQL 5.7	MySQL 8.0	Description
root	%	✓	✓	Admin account, which has permissions to perform all operations.
mysql.sys	localhost	✓	✓	The account used to manage and access the system's built-in sys database, through which you can quickly understand the metadata information of the system.
mysql.session	localhost	-	✓	The account used by internal plugins to access the server.
mysql.infoschema	localhost	-	✓	The account used to manage and access the system's built-in `information_schema` database.

Warning：

To avoid database problems, we recommend that you do not delete the default accounts.

Non-default account

Besides the default account created by the system, you can create other database accounts in the console as needed. For more information, see Creating Account.

Console Account
Account Registration

Last updated：2023-01-04 10:20:38

Before using TencentDB service, you need to sign up for a Tencent Cloud account. Then you can log in to the Tencent Cloud console with the account to purchase and use the TDSQL-C for MySQL product.

Step 1: Go to the registration page

Click to enter the Tencent Cloud registration page.

Step 2: Register an account

You can register for Tencent Cloud with the following options:

Option	Description
WeChat	Register for Tencent Cloud by scanning the QR code on WeChat. After successful registration, you can scan the code to log in to Tencent Cloud next time.
Email	Register for Tencent Cloud with your email address. This makes it simple for you to manage enterprise account.
QQ	Register for Tencent Cloud with your existing QQ account. After successful registration, you can use it to quickly log in to Tencent Cloud next time.
WeChat Official Account	Register for Tencent Cloud with your existing WeChat Official Account.
WeCom	Register for Tencent Cloud with your existing WeCom account

For more information, see Signing Up.

Related Operations

Identity Verification

Identity Verification

Last updated：2023-01-04 10:12:50

After registering a Tencent Cloud account, you need to complete identity verification or industry information.

Warning:

Identity directly affects the ownership of accounts and resources. If the enterprise users verify their identity with personal information, future personnel changes or account disputes may affect the business of the enterprise user and even cause economic losses. Therefore, before identity verification, you must confirm whether the resources you purchase and use in Tencent Cloud are owned by individuals or enterprises.

Verification Method

Individual Identity Verification Guide

Enterprise Identity Verification Guide

Database Account
Creating Account

Last updated：2023-08-24 09:38:34

Besides the default account created by the system, you can create other database accounts in the console as needed.

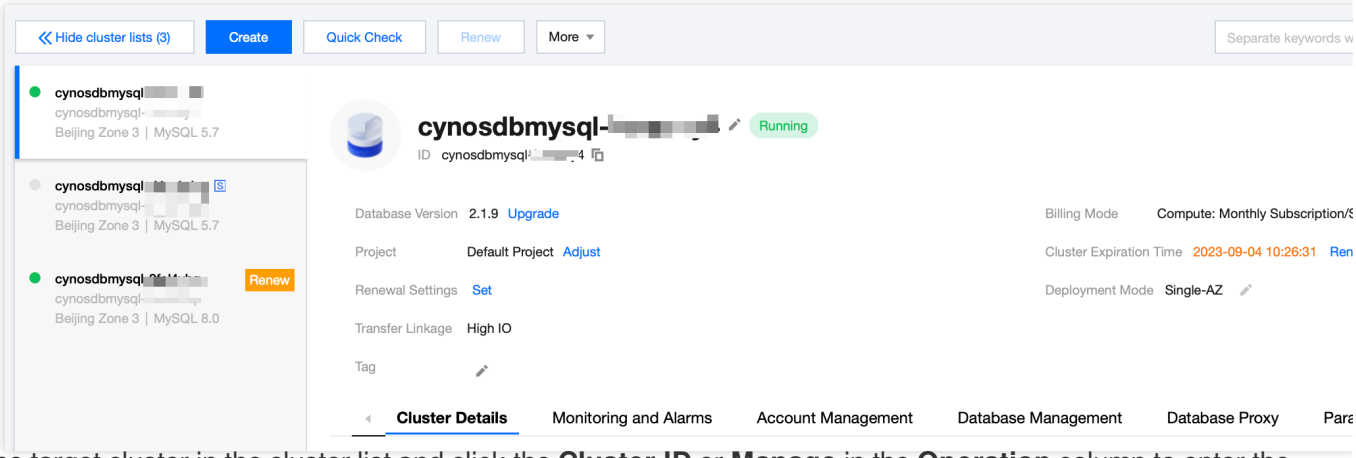
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode:

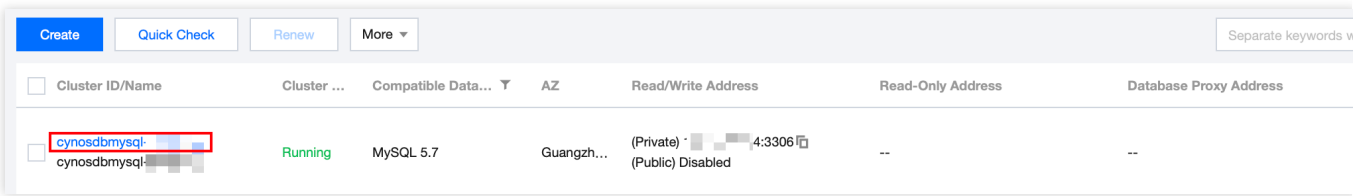
Tab view

List view

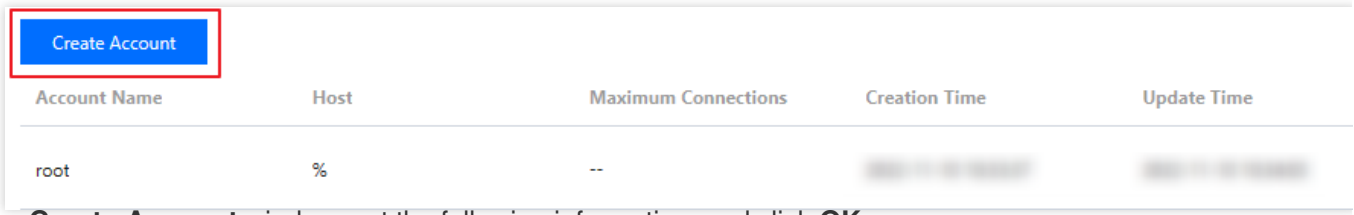
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



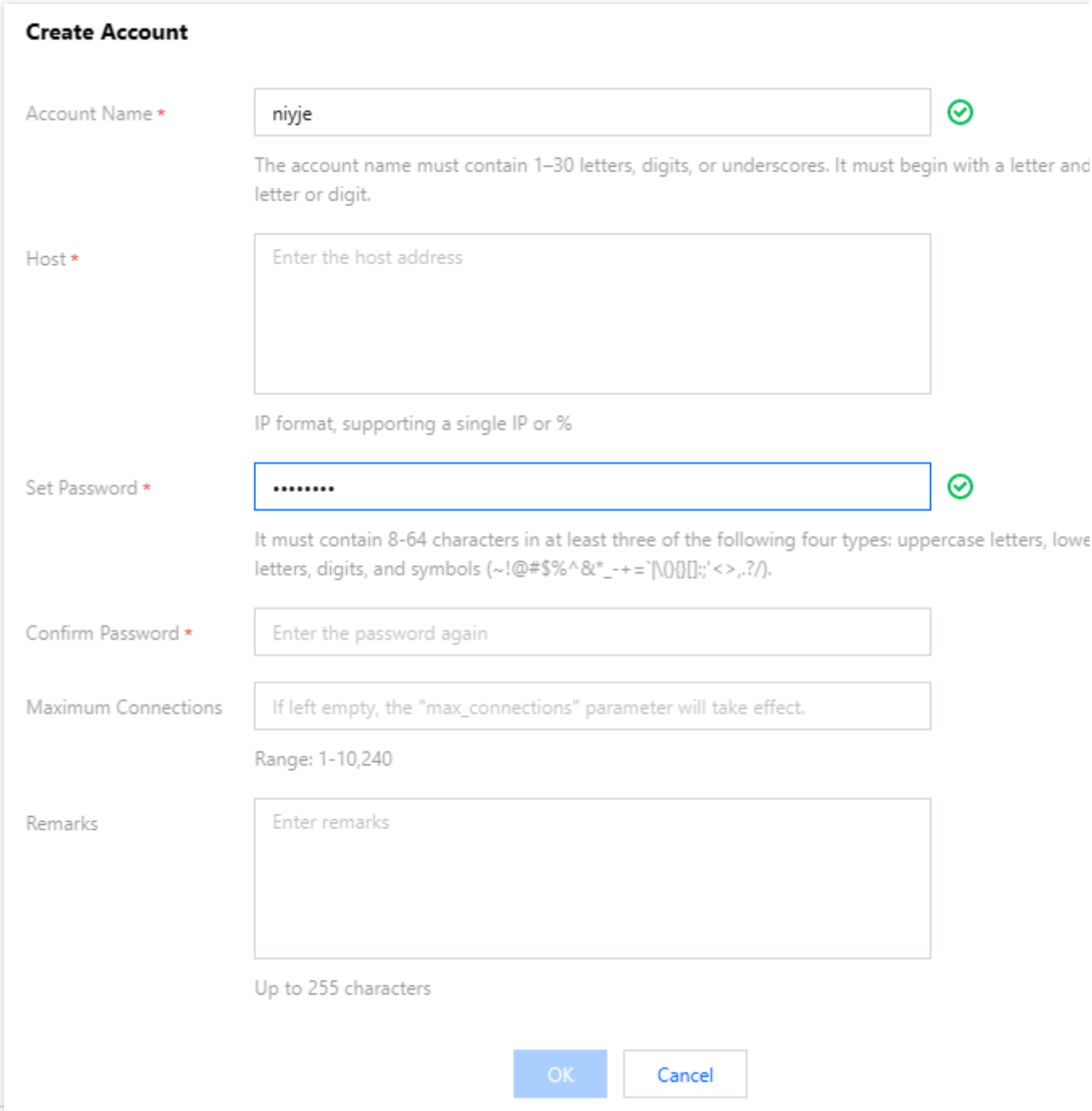
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Account Management** tab and click **Create Account**.



4. In the **Create Account** window, set the following information, and click **OK**.



Parameter	Description
Account Name	The database account name can contain 1-30 letters, digits, and underscores (_) and must begin with a letter and end with a letter or digit.

Host	The host address for database access can be an IP and contain `%` (indicating not to limit the IP range). Example 1: Enter `%` to indicate no limit on the IP range, that is, clients at all IP addresses are allowed to use this account to access the database. Example 2: Enter `10.5.10.%` to indicate that clients with an IP range within `10.5.10.%` are allowed to use this account to access the database.
Set Password	The password can contain 8–64 characters in at least three of the following character types: uppercase letters, lowercase letters, digits, and special symbols `~!@#$%^&*_-+=|\\(){}[]:;'<>,.?/` .
Confirm Password	Enter the same account password again.
Maximum Connections	Enter the maximum number of connections for the account. Valid range: 1-10240. If you don't enter a value, the maximum number of connections will be 10240.
Remarks	Remarks can contain up to 255 characters.

5. After the database account is created successfully, you can manage it in the database account list with the operations, such as permission modification, password reset, and account setting modification.

Configuring Custom Password Strength Overview

Last updated：2022-11-14 16:38:12

This document describes the custom password strength feature of TDSQL-C for MySQL.

Feature overview

Passwords are the most important means for protecting database security. As more data security regulations are introduced, there are higher requirements for the database password strength. TDSQL-C for MySQL supports the custom password strength feature to protect your database security and meet your needs for compliance with applicable regulations.
You can configure this feature in the console to enable password strength for all password-related operations. This helps protect your passwords from leakage or other risks. The feature offers the following configuration items:

Parameter	Description
Min Number of Uppercase and Lowercase Pair	Default value: `1`. Value range: 1–50.
Min Number of Digits	Default value: `1`. Value range: 1–50.
Min Number of Symbols	Default value: `1`. Value range: 1–50.
Min Password Length	Default value: `8`. Value range: 8–256.
Non-Compliant Dictionary	If the password strength level is **STRONG**, this parameter is configurable. Each non-compliant word can contain 4–100 letters.
Password Strength Level	You can select **MEDIUM** or **STRONG** as the strength level. MEDIUM: The feature under this setting will check the length, digits, letters, and symbols. STRONG: The feature under this setting will check the length, digits, letters, symbols, and non-compliant word dictionary.
Modify	You can modify the feature parameters to flexibly adjust the password strength settings.

Parameters	
Parameter Sync	Parameter sync and batch disablement features are provided, so you can batch apply the configuration in multiple clusters at a time.

After the custom password strength feature is enabled, you must meet the defined password strength requirements when setting passwords during operations such as account creation, password resetting, and account cloning.
When you connect to the database and use the command line to perform operations, if the custom password strength feature is enabled, all statements involving password setting will be restricted, such as CREATE USER, ALTER USER, and SET PASSWORD. When you use such statements to set or change an account password, the password must meet the defined password strength requirements.

Prerequisites

You have created a TDSQL-C for MySQL cluster.

Version limits

The custom password strength feature is supported by the following versions:
MySQL 5.7 on kernel minor version 2.0.21 or later and 2.1.7 or later.
MySQL 8.0 on kernel minor version 3.1.7.
You can use this feature only after upgrading the kernel to the above versions. For detailed directions, see Upgrading Kernel Minor Version.

Mutually exclusive tasks

A password strength customization task cannot be executed while a cluster is running another task, such as isolation, rollback, creation, minor version upgrade, or parameter modification. The two tasks are mutually exclusive and will be performed sequentially.

Relevant operations

You can grant a sub-account the permission to use the custom password strength feature as instructed in Granting Sub-User Feature Permissions.
You can enable/disable this feature as instructed in Enabling/Disabling Custom Password Strength Feature.

After the custom password strength feature is enabled, you can modify the custom password strength and specific parameters as well as configure parameter sync as instructed in Modifying Parameters and Configuring Parameter Sync.

Granting Sub-User Feature Permissions

Last updated：2022-09-19 15:47:05

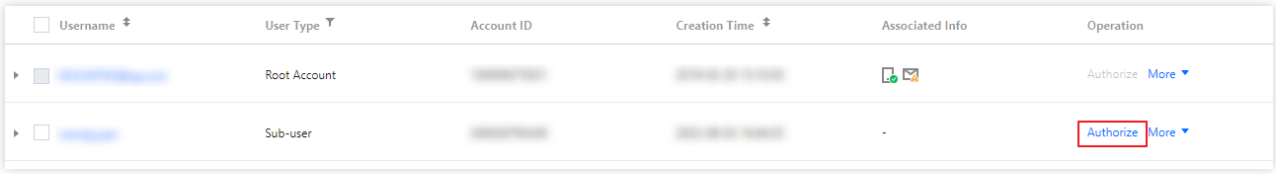
This document describes how to use the root account to grant a sub-user the permissions of the custom password strength feature.

Overview

Your root account has all permissions of the custom password strength feature with no additional settings needed. By default, sub-users don't have permissions of this feature. Therefore, you need to create policies to allow the target sub-user to use the feature.
Cloud Access Management (CAM) is a web service provided by Tencent Cloud that helps you securely manage access to the resources under your Tencent Cloud account. CAM allows you to create, manage, or terminate users or user groups and control who is allowed to use your Tencent Cloud resources through identity and policy management. When using CAM, you can associate a policy with a user or user group to allow or forbid them to use specified resources to complete specified tasks. For more information on CAM policies, see Element Reference.

Directions

1. Log in to the CAM console with the root account, locate the target sub-user in the user list, and click **Authorize**.



2. Set the following permissions based on the account's needs. Note that no matter whether the sub-user needs to use the custom password strength feature, we recommend you always add `cynosdb:DescribeClusterPasswordComplexity` to the sub-user, so that they can view the detailed password strength settings configured by the root account in the console and set compliant passwords accordingly.

CAM permission description

Permission	Purpose

cynosdb:DescribeClusterPasswordComplexity	Displays the detailed settings and enablement status of the custom password strength feature
cynosdb:CloseClusterPasswordComplexity	(Batch) disables the custom password strength feature
cynosdb:CopyClusterPasswordComplexity	Syncs the custom password strength parameters to other clusters under the current account
cynosdb:ModifyClusterPasswordComplexity	Modifies the detailed settings of the custom password strength feature
cynosdb:OpenClusterPasswordComplexity	Enables the custom password strength feature

Enabling/Disabling Custom Password Strength Feature

Last updated：2022-09-19 15:47:05

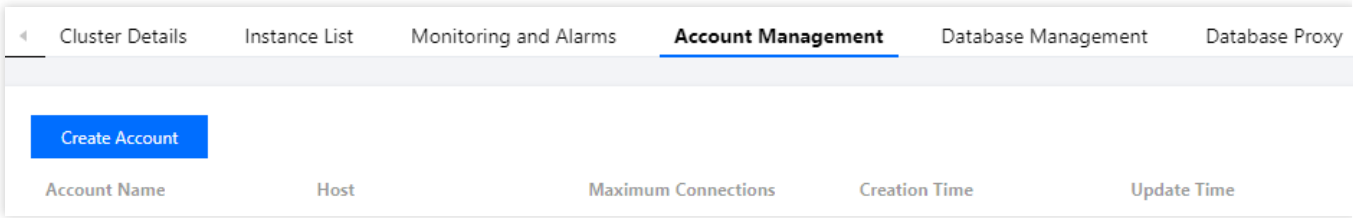
You can use the custom password strength feature with the root account or a sub-account granted with the corresponding CAM permissions. This document describes how to enable and disable the custom password strength feature.

Version limits

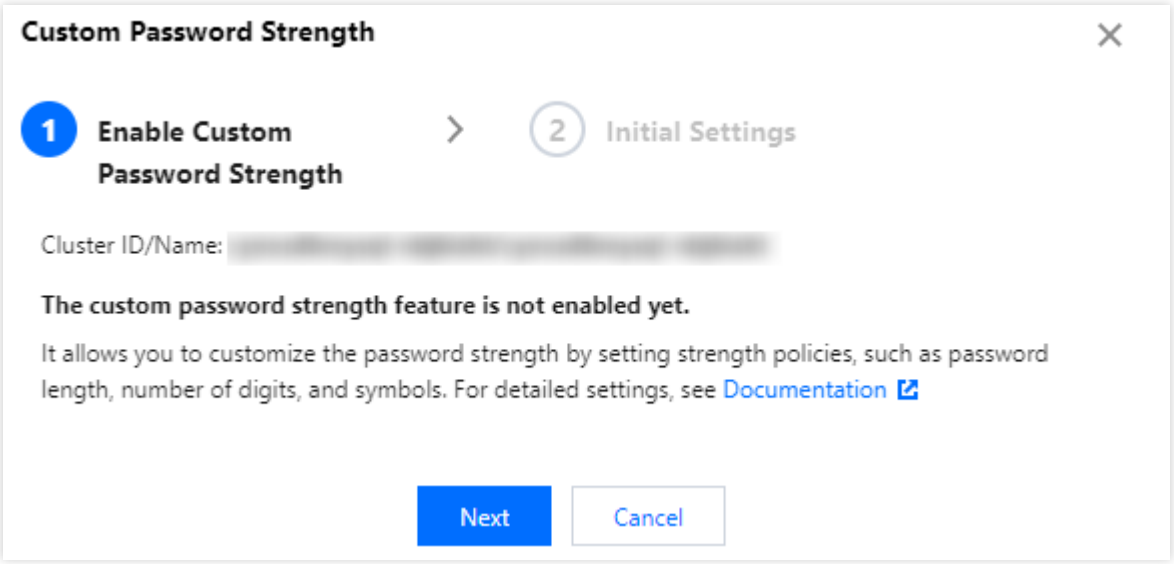
The custom password strength feature is supported by the following versions:
MySQL 5.7 on kernel minor version 2.0.21 or later and 2.1.7 or later.
MySQL 8.0 on kernel minor version 3.1.7.
You can use this feature only after upgrading the kernel to the above versions. For detailed directions, see Upgrading Kernel Minor Version.

Enabling custom password strength feature

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
3. On the cluster management page, select **Account Management** and enable **Custom Password Strength** on the right.



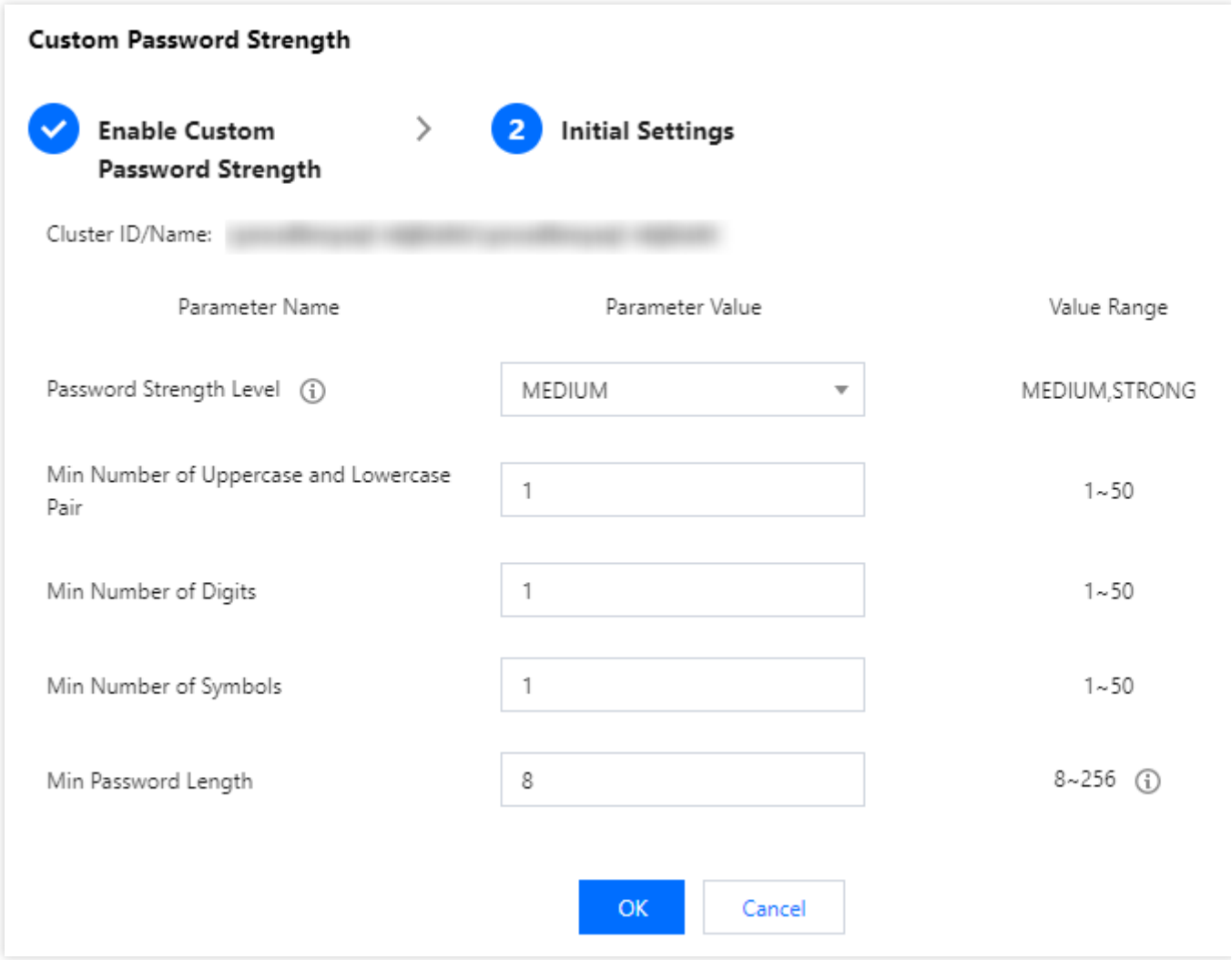
4. In the pop-up window, click **Next** to enter the initial settings page.



5. Complete the following settings and click **OK**.

In the initial settings of the custom password strength feature, two scenarios will be displayed for different password strength levels.

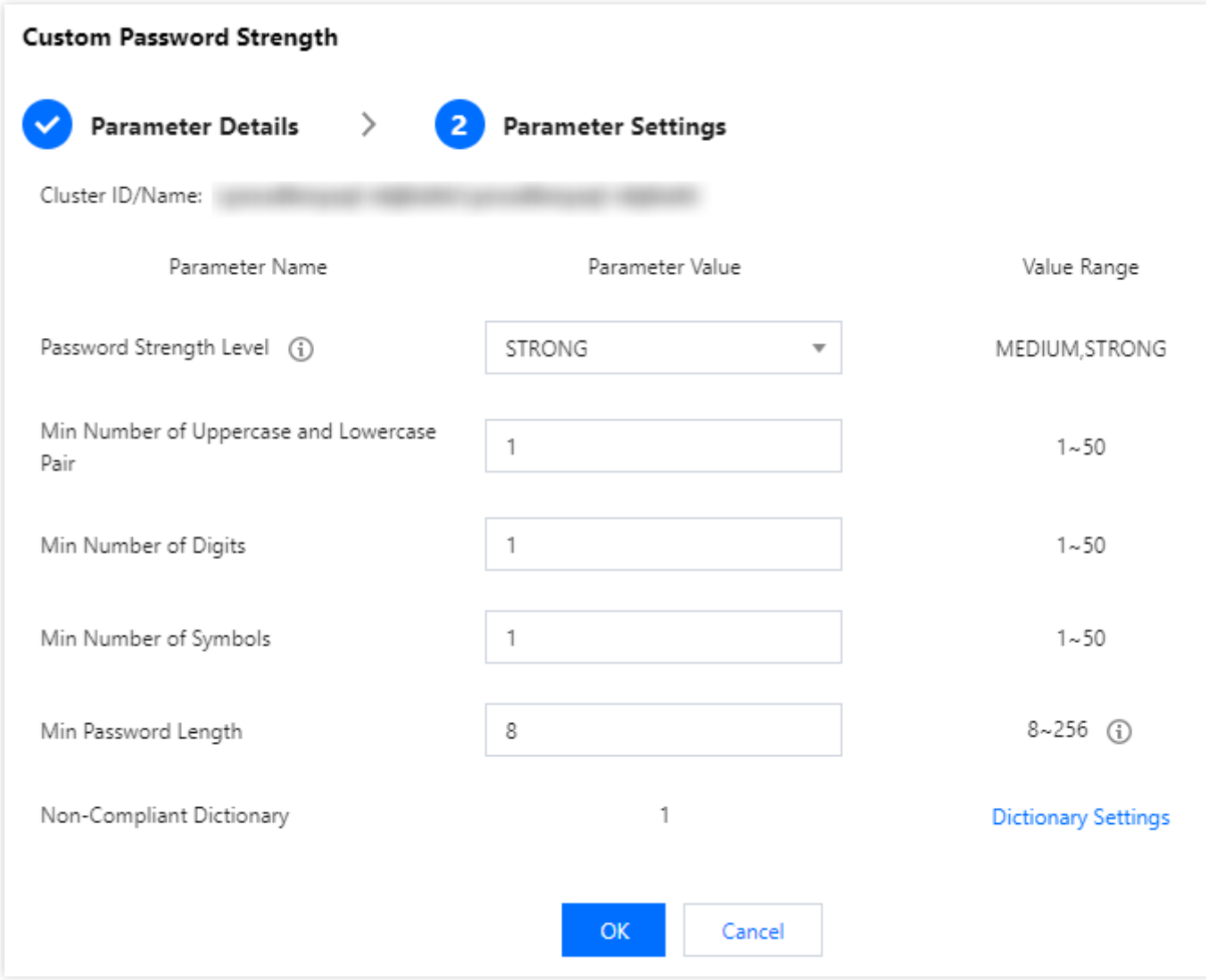
Scenario 1: The password strength level is **MEDIUM**.



Parameter	Description
Password Strength Level	You can select MEDIUM or STRONG as the strength level. MEDIUM: The feature under this setting will check the length, digits, letters, and symbols. STRONG: The feature under this setting will check the length, digits, letters, symbols, and non-compliant word dictionary.
Min Number of Uppercase and Lowercase Pair	The minimum number of pairs of uppercase and lowercase letters that the password must contain. For example, if this parameter is set to `2`, the password must contain at least two uppercase letters and two lowercase letters. Default value: `1`. Value range: 1–50.
Min	The minimum number of digits that the password must contain. Default value: `1`. Value range: 1–

Number of Digits	50.
Min Number of Symbols	The minimum number of special symbols that the password must contain. Default value: `1`. Value range: 1–50.
Min Password Length	The minimum length of the password. Default value: `8`. Value range: 8–256. This parameter equals to the number of digits + number of symbols + 2 * number of letters, and it must be greater than or equal to 8 for security of your password. If the final value after the sum of above parameters is greater than 8, it will be used as the minimum of the range.

Scenario 2: The password strength level is **STRONG**.

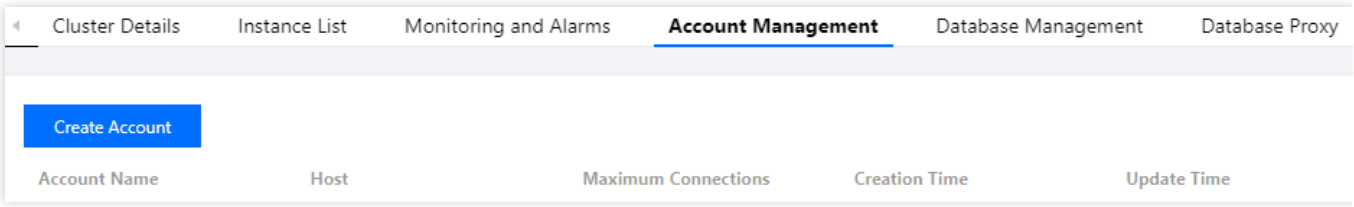


Parameter	Description

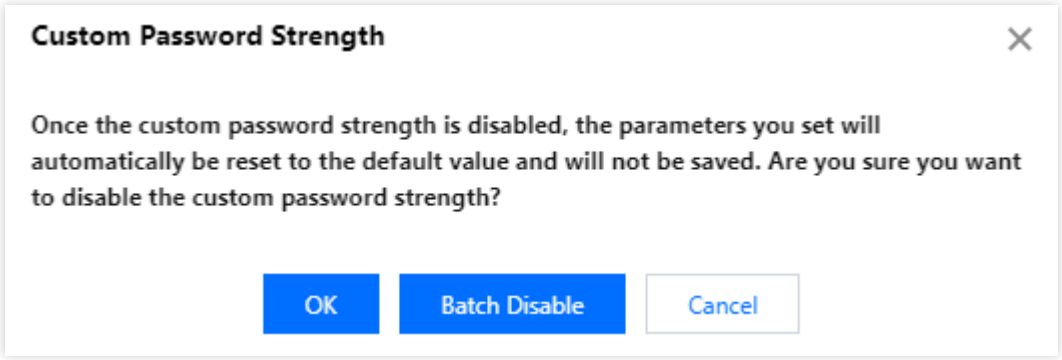
Password Strength Level	You can select MEDIUM or STRONG as the strength level. MEDIUM: The feature under this setting will check the length, digits, letters, and symbols. STRONG: The feature under this setting will check the length, digits, letters, symbols, and non-compliant word dictionary.
Min Number of Uppercase and Lowercase Pair	The minimum number of pairs of uppercase and lowercase letters that the password must contain. For example, if this parameter is set to `2`, the password must contain at least two uppercase letters and two lowercase letters. Default value: `1`. Value range: 1–50.
Min Number of Digits	The minimum number of digits that the password must contain. Default value: `1`. Value range: 1–50.
Min Number of Symbols	The minimum number of special symbols that the password must contain. Default value: `1`. Value range: 1–50.
Min Password Length	The minimum length of the password. Default value: `8`. Value range: 8–256. This parameter equals to the number of digits + number of symbols + 2 * number of letters, and it must be greater than or equal to 8 for security of your password. If the final value after the sum of above parameters is greater than 8, it will be used as the minimum of the range.
Non-Compliant Dictionary	If the password strength level is STRONG, this parameter is configurable. You can click Dictionary Settings for configuration. Each non-compliant word can contain 4–100 letters. After configuration, the system will check passwords for non-compliant words during password verification. If any non-compliant word (case-insensitive) is detected, the verification will fail.

Disabling custom password strength feature

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
3. On the cluster management page, select **Account Management** and disable **Custom Password Strength** on the right.



4. In the pop-up window, click **OK**.



5. In the pop-up window, you can click **Batch Disable** to select other clusters with the custom password strength feature enabled and batch disable the feature.

Note:

Once the custom password strength feature is disabled, the parameters you set will be automatically reset to the default value. If you enable the feature again, you need to configure these parameters again.

You can select up to 20 clusters in one batch disablement operation.

Modifying Parameters and Configuring Parameter Sync

Last updated：2024-03-07 10:32:03

After the custom password strength feature is enabled, you can modify the feature parameters in the current cluster and sync them to other clusters under your account. This document describes how to modify such parameters and configure parameter sync.

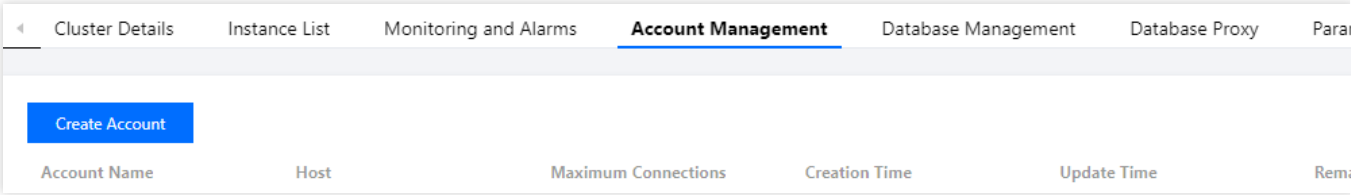
Prerequisites

You have enabled the custom password strength feature as instructed in Enabling/Disabling Custom Password Strength Feature.

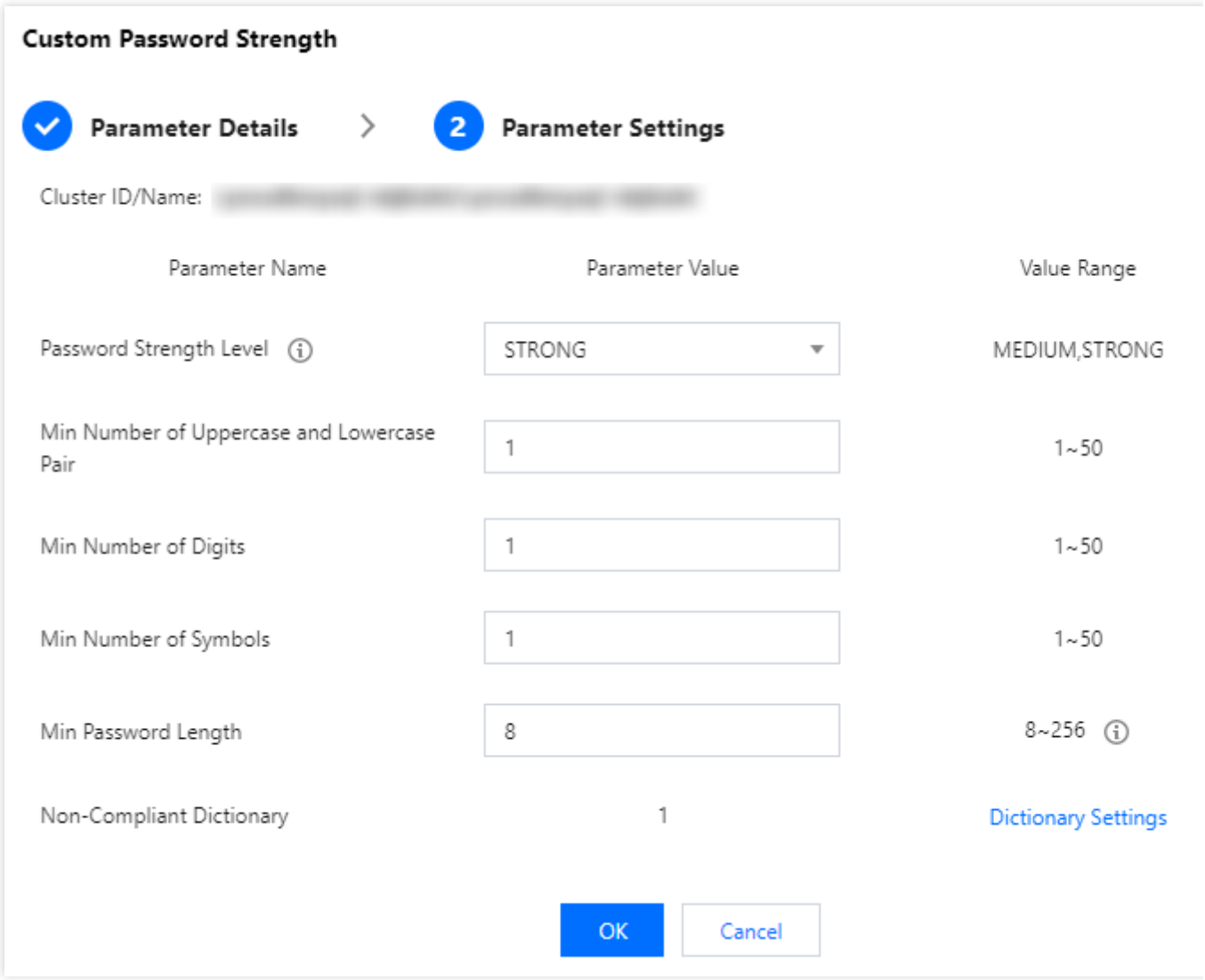
Modifying Custom Password Strength Parameters

Note:
The password check strength levels are divided into MEDIUM and STRONG, with the Prohibited Word Dictionary parameter only appearing when the strength is set to STRONG.

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
3. On the cluster management page, select **Account Management** and click **Strength Settings** on the right.



4. In the pop-up window, click **Next** to enter the **Parameter Settings** page.
5. Within the parameter modification interface, establish a new parameter configuration, then click **OK**.



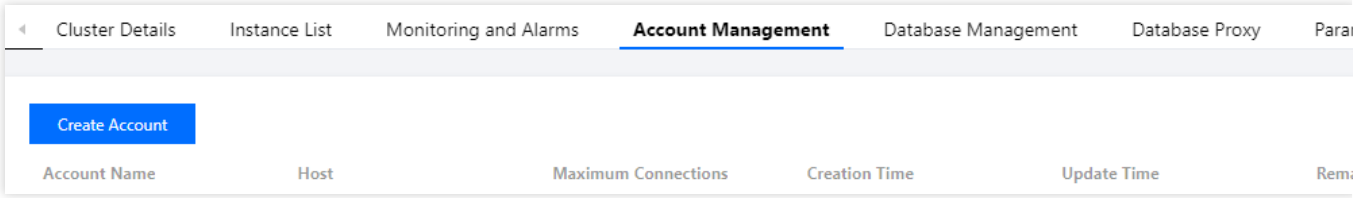
Setting Parameter Sync

Note：

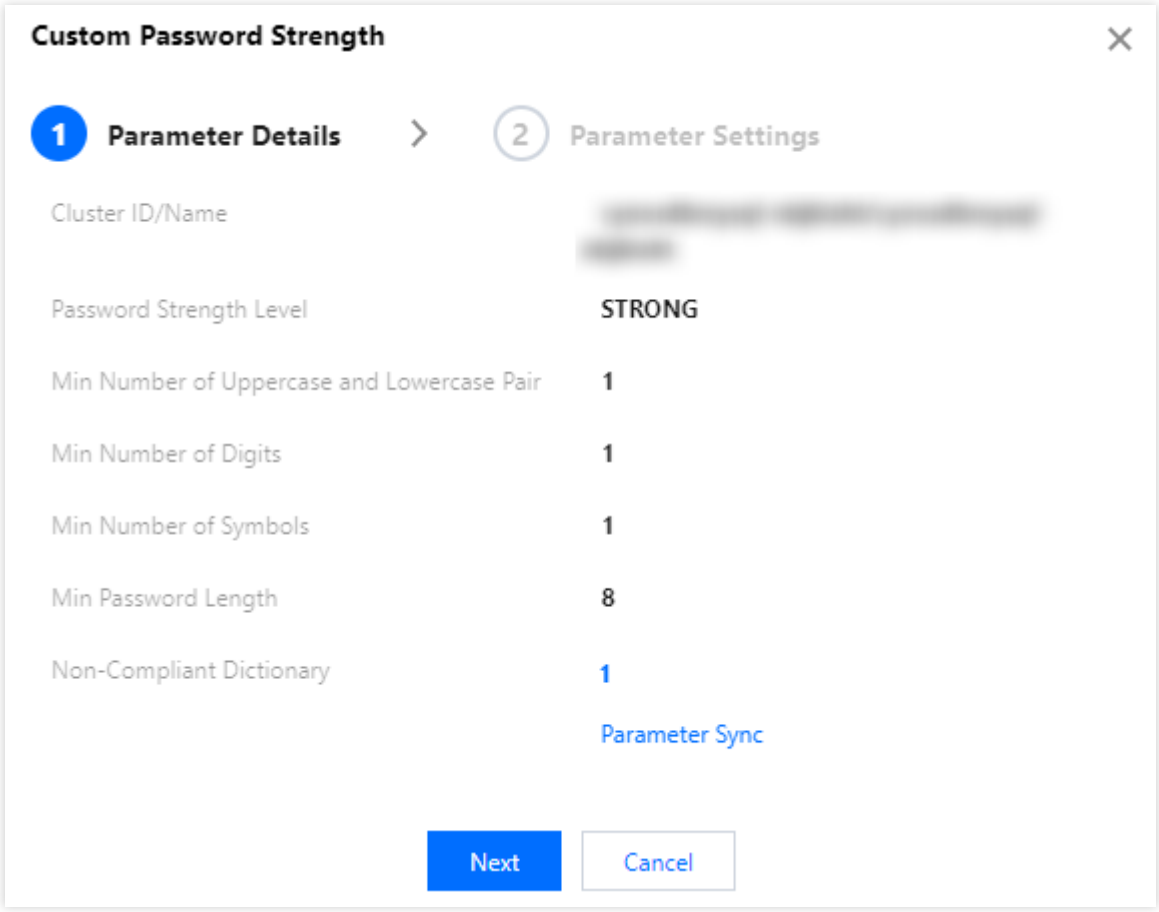
If you are logging into the console using a sub-account, ensure that this sub-account is authorized to use the parameter synchronization function for custom password strength. For detailed authorization function names and operations, please refer to Granting Sub-User Feature Permissions.

After enabling the custom password strength feature in a cluster, you can use the parameter sync feature to sync the configured custom password strength parameters, including **Non-Compliant Dictionary**, to other clusters in the same region under the current Tencent Cloud account. If this feature is disabled in a target cluster, it will be enabled to sync the parameters.

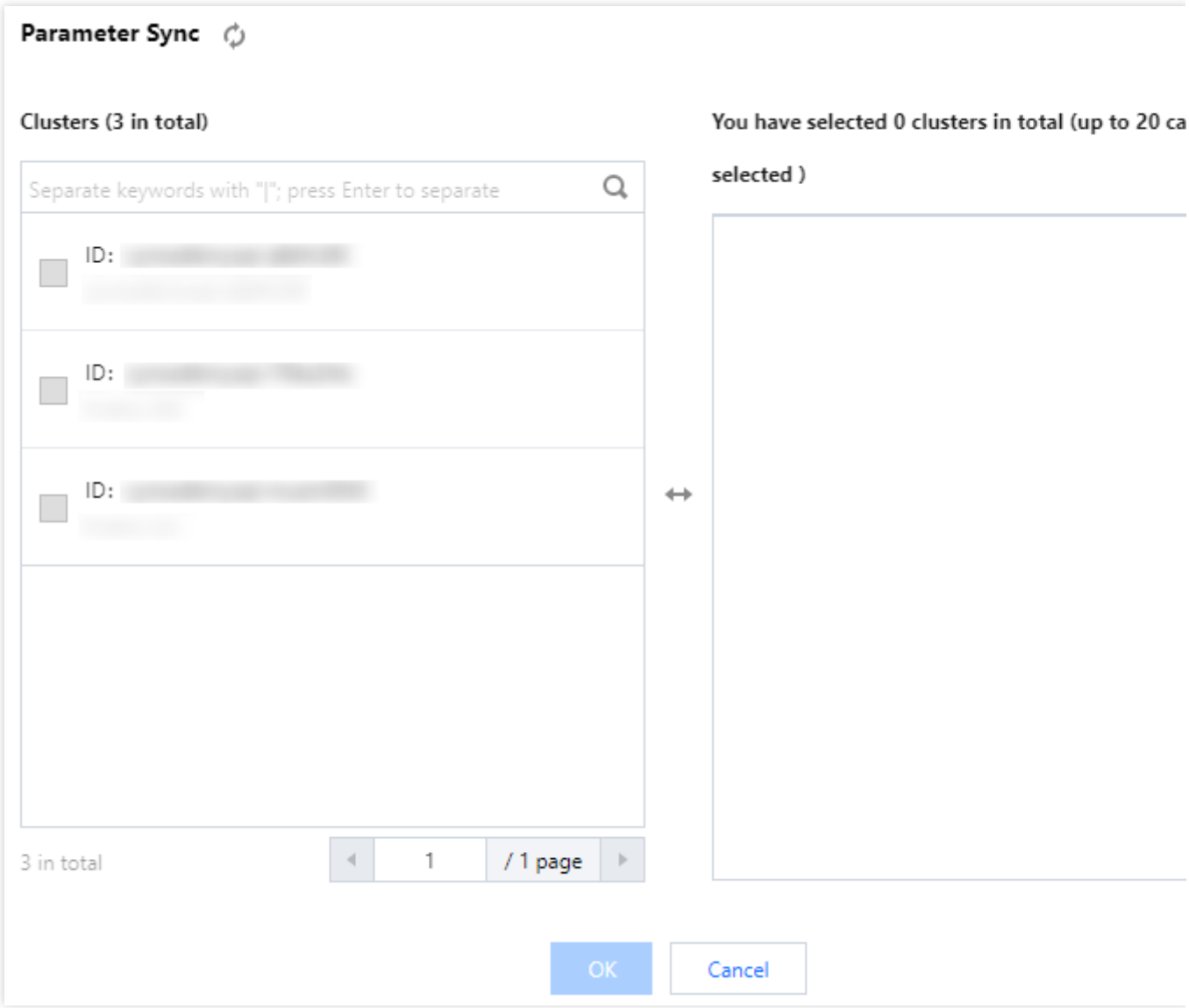
1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
3. On the cluster management page, select **Account Management** and click **Strength Settings** on the right.



4. On the parameter modification page, click **Parameter Sync**.



5. On the **Parameter Sync** page, select or deselect one or multiple clusters and click **OK**.



Resetting Password

Last updated：2023-08-24 09:41:02

This document describes how to reset the password of the database admin account in the console.

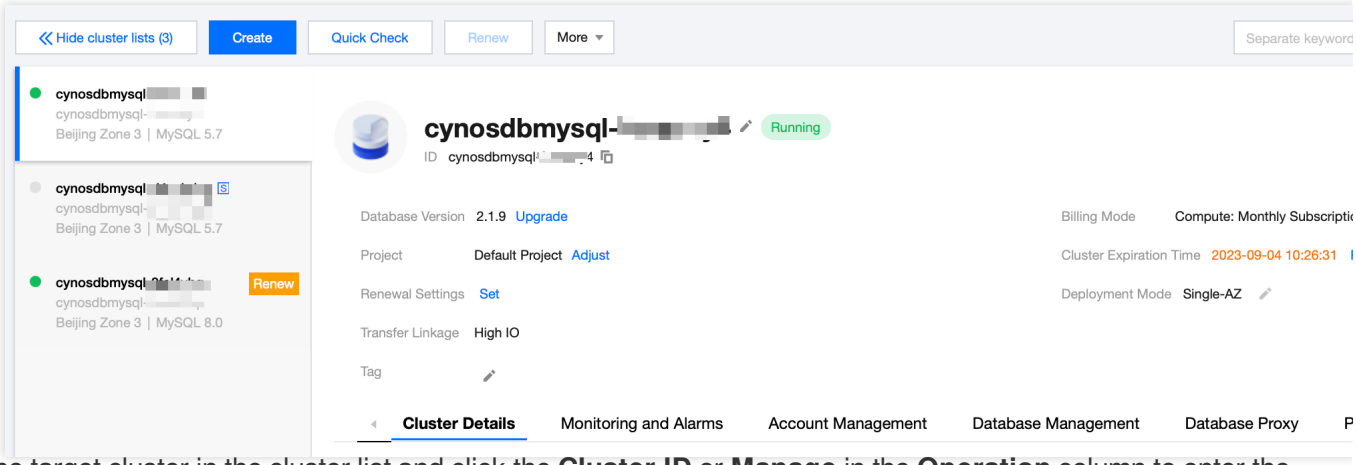
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

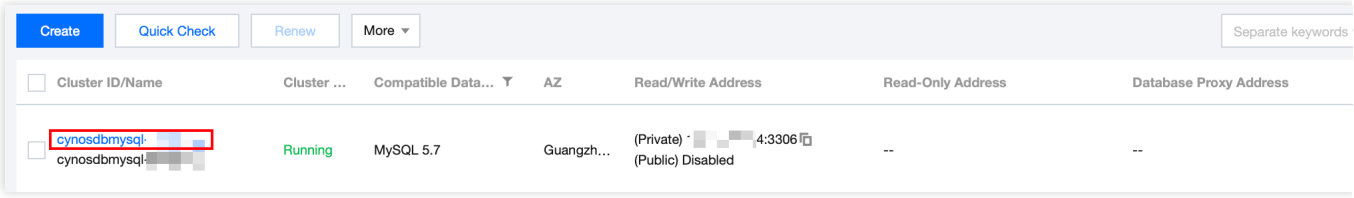
Tab view

List view

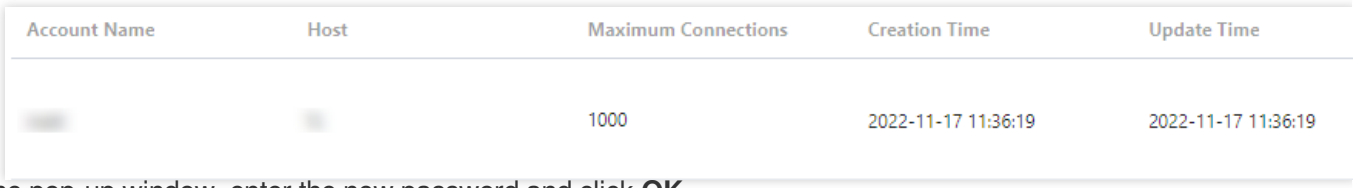
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Account Management** tab and click **Reset Password** in the **Operation** column.



Account Name	Host	Maximum Connections	Creation Time	Update Time
[redacted]	[redacted]	1000	2022-11-17 11:36:19	2022-11-17 11:36:19

4. In the pop-up window, enter the new password and click **OK**.

Note：

Password must contain 8-64 characters in at least three of the following four types: uppercase letters, lowercase letters, digits, and symbols (`~!@#$%^&*_-+=|\\(){}[]:;'<>,.?/`).

Modifying Account Permission

Last updated：2023-08-24 09:44:56

You can manage the permissions of existing database accounts in the console. Specifically, you can grant or revoke global/object-level privileges for those accounts.

Account Permission Description

Permission	Description
ALTER	Permission for modifying structure of the database table, including modifying/adding/deleting table fields, and adding/deleting the table index
ALTER ROUTINE	Permission for modifying a stored process and function
CREATE	Permission for creating a database, table, view, stored procedure, and function
CREATE ROUTINE	Creates routine objects such as stored procedures and functions
CREATE TEMPORARY TABLES	Permission for creating a temp table
CREATE USER	Permission for creating a user and assigning permission for it
CREATE VIEW	Permission for creating a view
DELETE	Permission for deleting data from the specified table
DROP	Permission for deleting a database, table, view, stored procedure, and function
EVENT	Permission for creating, modifying, and deleting an event
EXECUTE	Permission for executing the stored procedure and function
GRANT OPTION	Permission for granting a permission to the specified user.
INDEX	Permission for creating and deleting an index
INSERT	Permission for inserting or writing new data into the table
LOCK TABLES	Permission for locking the specified data table for read and write operations
PROCESS	Permission for viewing information of all executing threads and processes
REFERENCES	Permission for creating or deleting foreign key constraint in the current database

RELOAD	Permission for reloading system configuration files, restarting or shutting down a database
REPLICATION CLIENT	Permission for viewing and managing replication process status and parameters
REPLICATION SLAVE	Permission for replicating data from the Master server
SELECT	Permission for viewing data in the specified table
SHOW DATABASES	Permission for displaying all databases in the current server
SHOW VIEW	Permission for displaying views
TRIGGER	Permission for creating and managing a trigger
UPDATE	Permission for updating data in the specified table

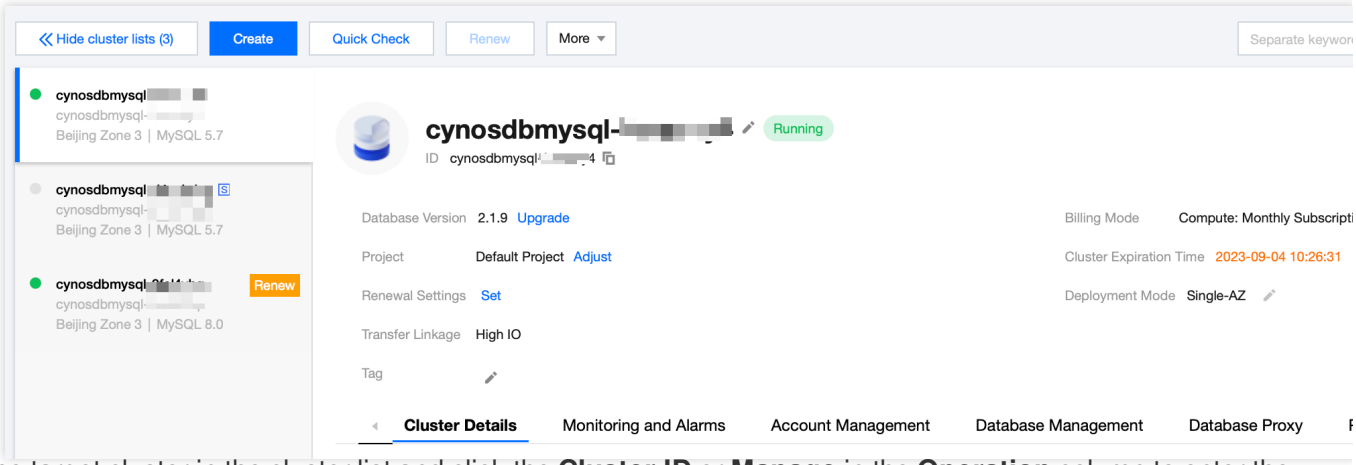
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode:

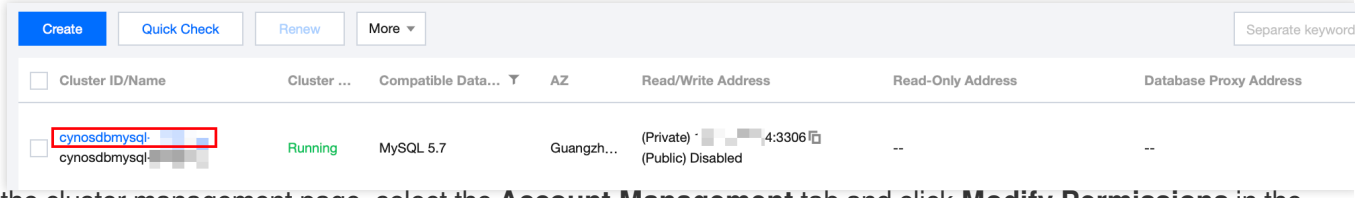
Tab view

List view

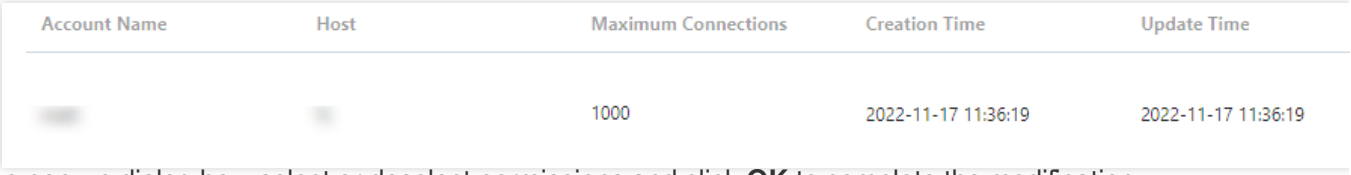
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



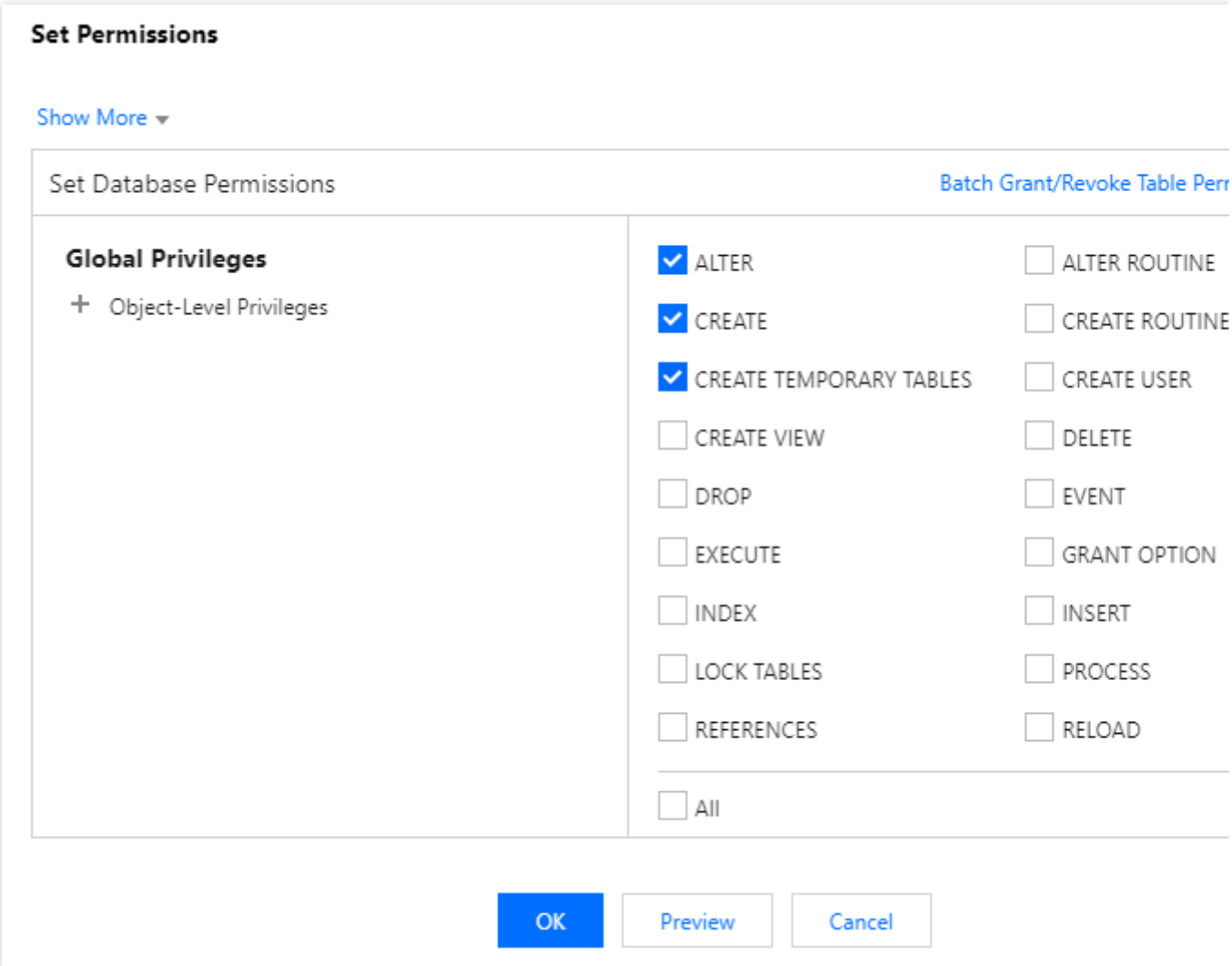
3. On the cluster management page, select the **Account Management** tab and click **Modify Permissions** in the **Operation** column.



4. In the pop-up dialog box, select or deselect permissions and click **OK** to complete the modification.

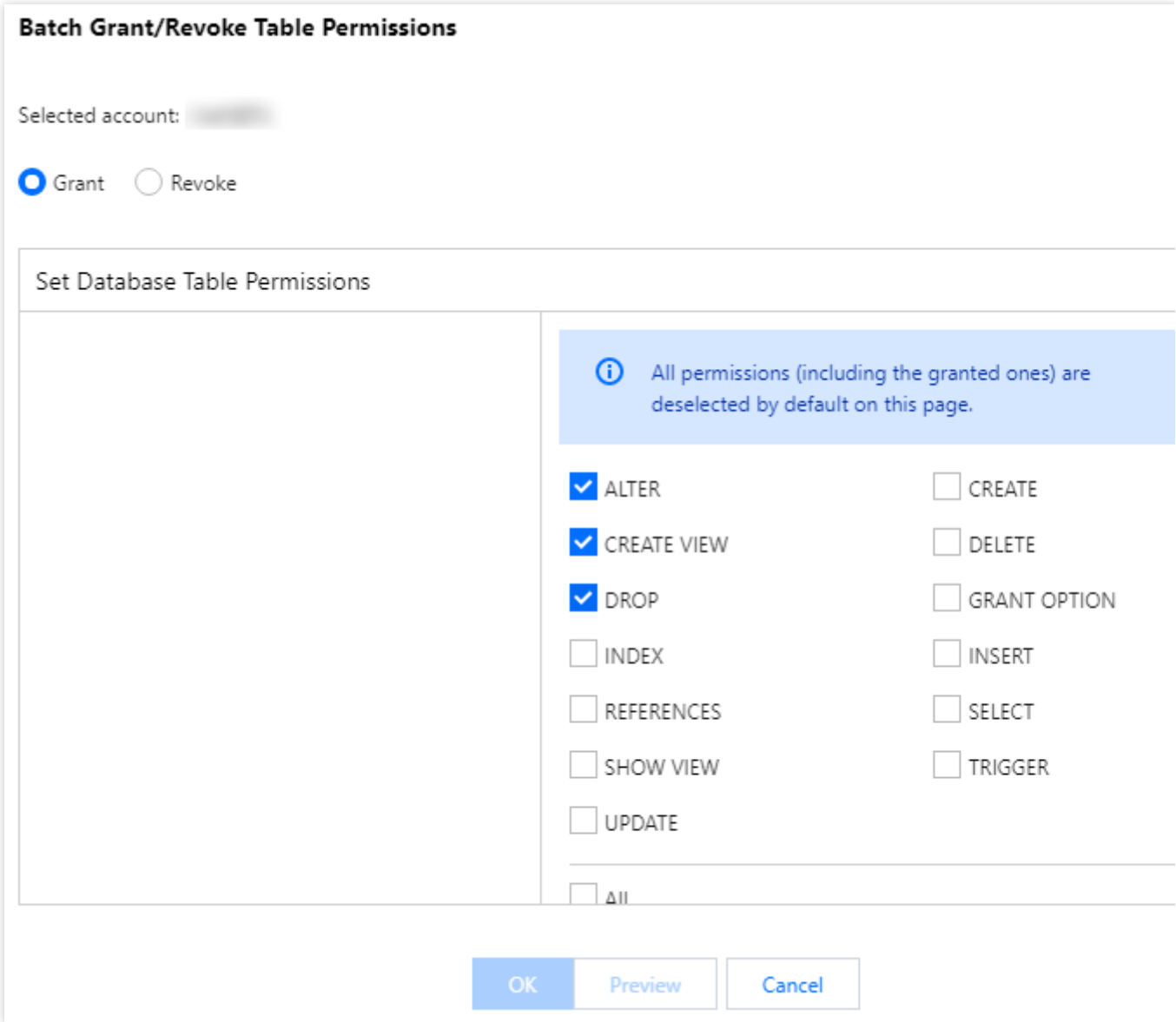
Global Privileges: Grant permissions to all databases in the instance.

Object-Level Privileges: Grant permissions to certain databases in the instance.

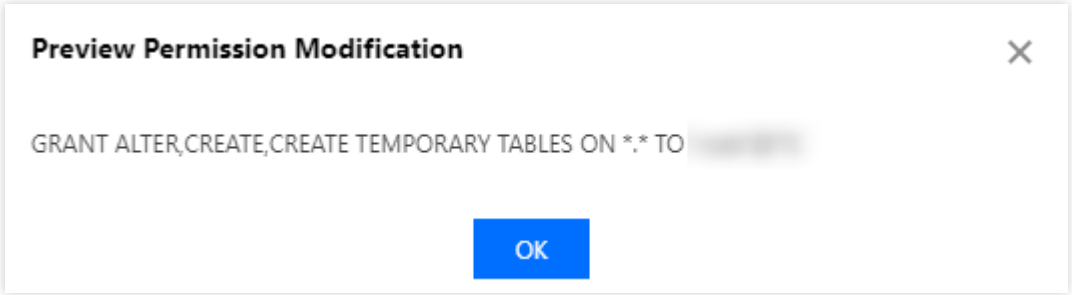


Note：

To batch grant or revoke table permissions, click **Batch Grant/Revoke Table Permissions**, select **Authorize** or **Revoke** in the pop-up window, select multiple target tables and permissions for authorization, and click **OK**.



To preview account permission modification, you can click **Preview**.



Modifying Authorized Host Address

Last updated：2023-08-24 09:42:48

You can modify the host address authorized by a database account in the console. By doing so, you can control the access to the database and enhance database connection security.

Directions

Note：

You can't modify the host address of the `root` account.

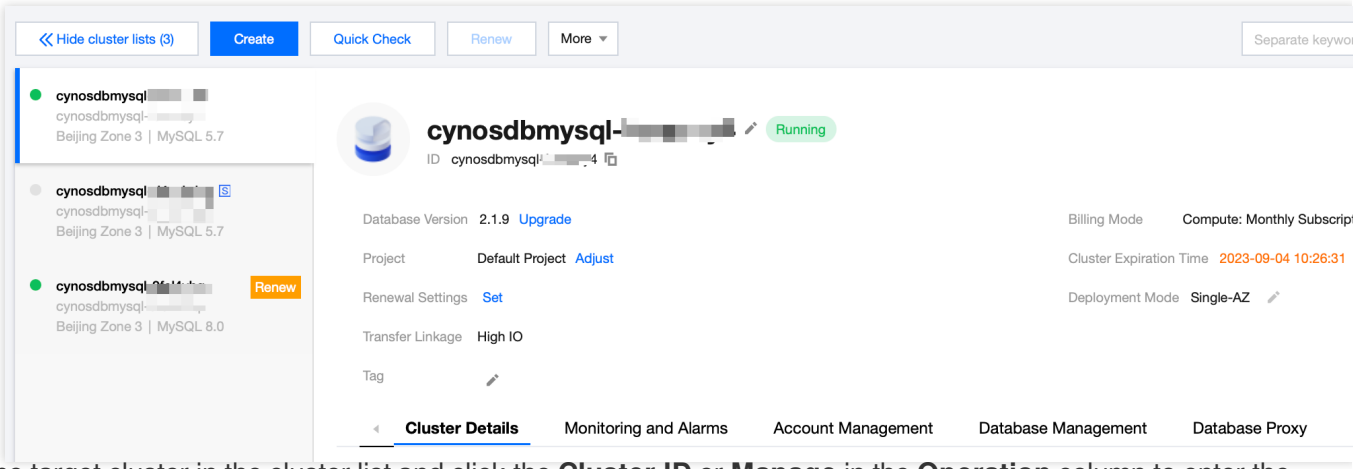
1. Log in to the TDSQL-C for MySQL console.

2. Select a region at the top and proceed according to the actually used view mode.

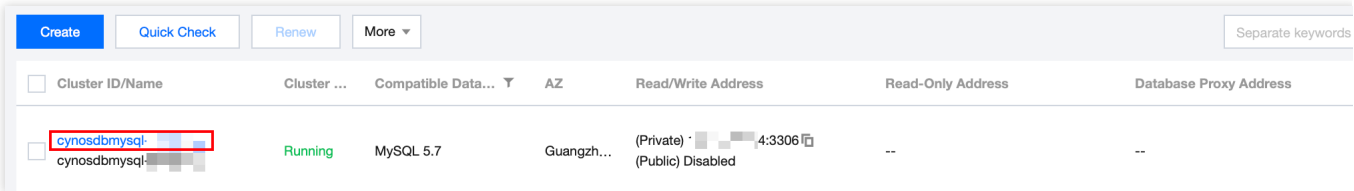
Tab view

List view

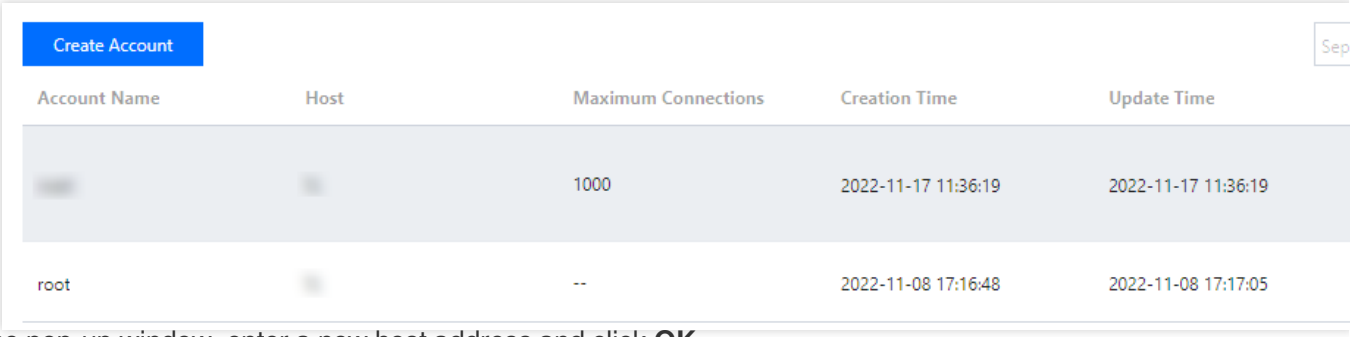
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Account Management** tab and select **More** > **Modify Host** in the **Operation** column.



4. In the pop-up window, enter a new host address and click **OK**.

Note：

The host address can be an IP or `%` (indicating no limit on the IP range).

Example 1: Enter `%` to indicate no limit on the IP range, that is, clients at all IP addresses are allowed to use this account to access the database.

Example 2: Enter `10.5.10.%` to indicate that clients with an IP range within `10.5.10.%` are allowed to use this account to access the database.

Modifying Account Connections

Last updated：2023-08-22 15:43:12

TDSQL-C for MySQL allows you to modify the maximum connections of each account to the database in the console. By doing so, you can prevent a single account from using up all connections.

This document describes how to modify the account connections in the console.

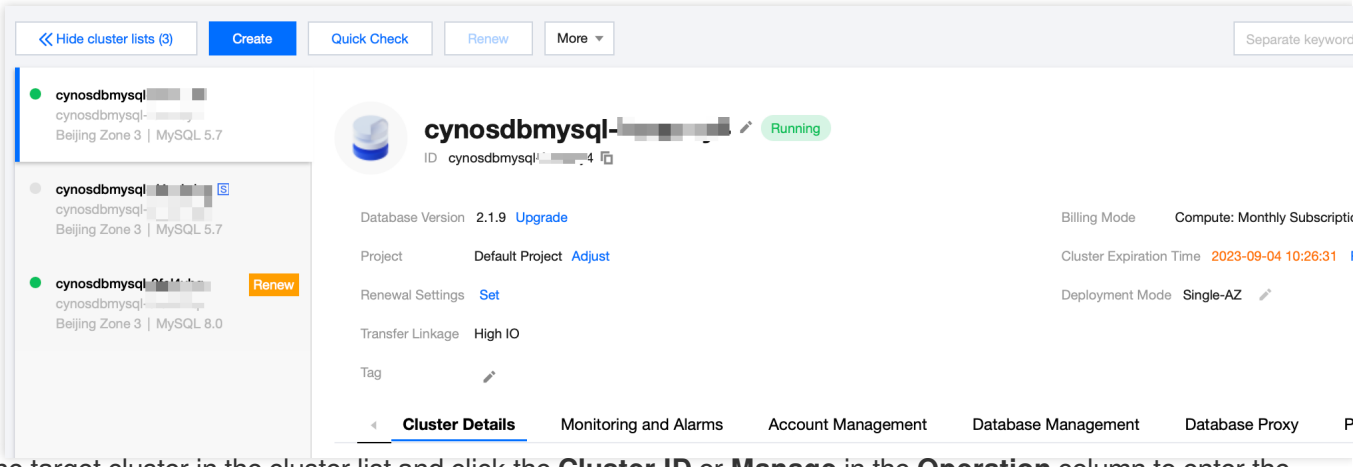
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

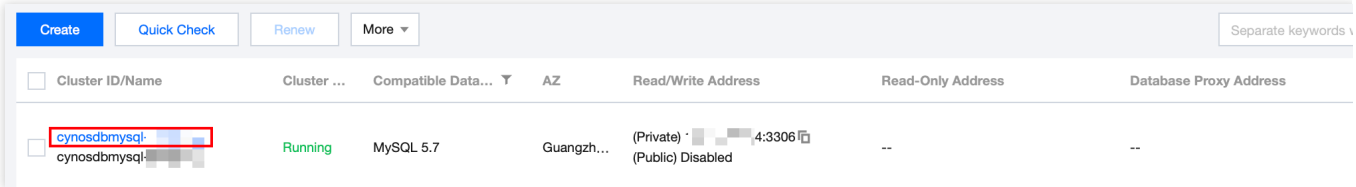
Tab view

List view

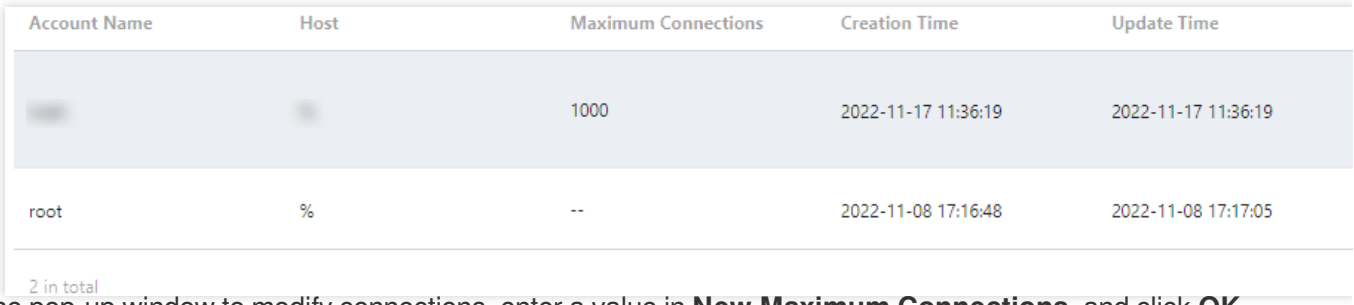
Click the target cluster in the cluster list on the left to enter the cluster management page.



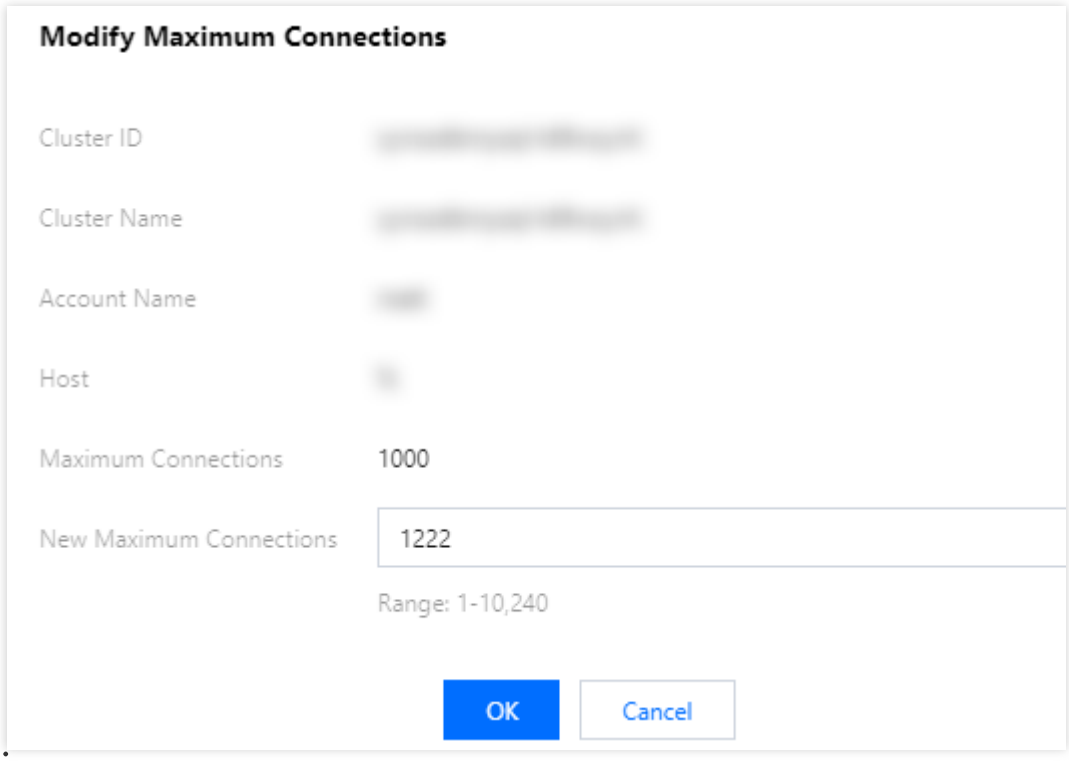
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. Click **Account Management**, find the target account in the list, select **More** > **Modify Maximum Connections** in the **Operation** column.



4. In the pop-up window to modify connections, enter a value in **New Maximum Connections**, and click **OK**.



Note：

Valid range of connections: 1-10240. If you don't enter a value, the maximum number of connections will be 10240.

Modifying Account Remarks

Last updated：2024-01-02 11:48:30

This document describes how to modify the remarks of a database account in the console to allocate and manage the account resources.

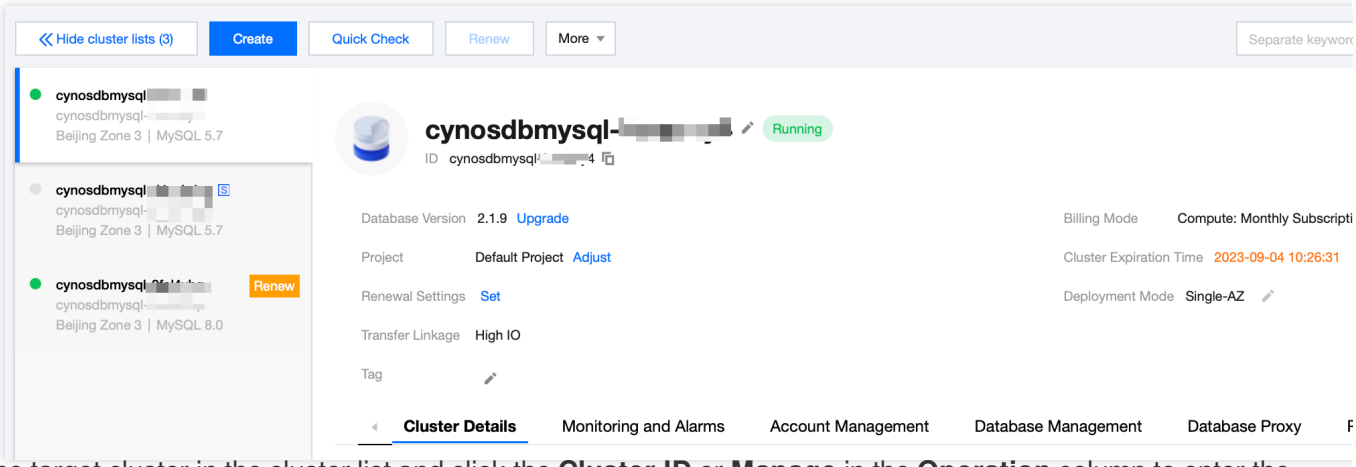
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

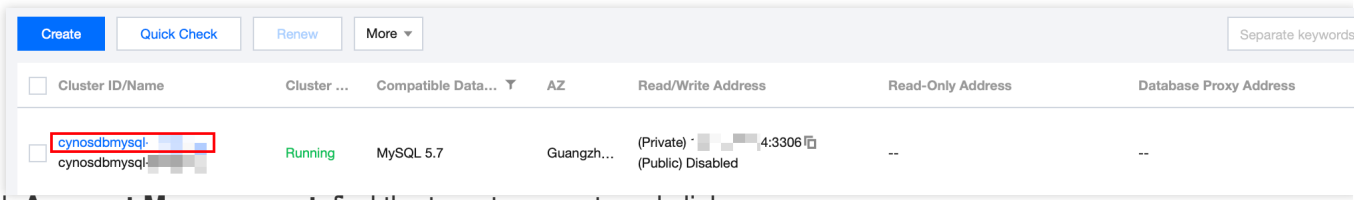
Tab view

List view

Click **Target Cluster** in the cluster list on the left to enter the cluster management page.

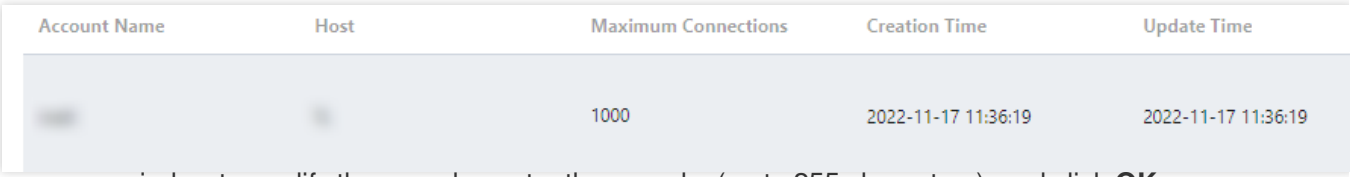


Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.

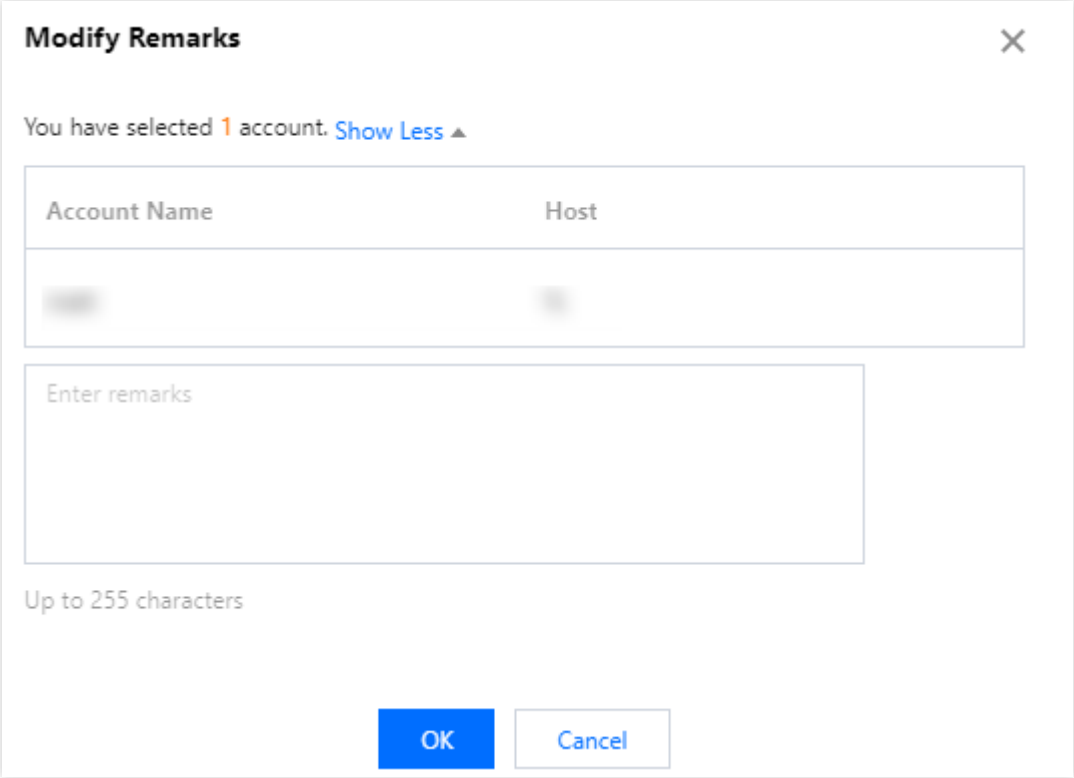


3. Click **Account Management**, find the target account, and click

after the remarks.



4. In the pop-up window to modify the remarks, enter the remarks (up to 255 characters), and click **OK**.



Cloning Account

Last updated：2023-08-22 15:55:51

This document describes how to clone an account in the console. The cloned account will inherit the account name, host, remarks, and account permission information of the original account.

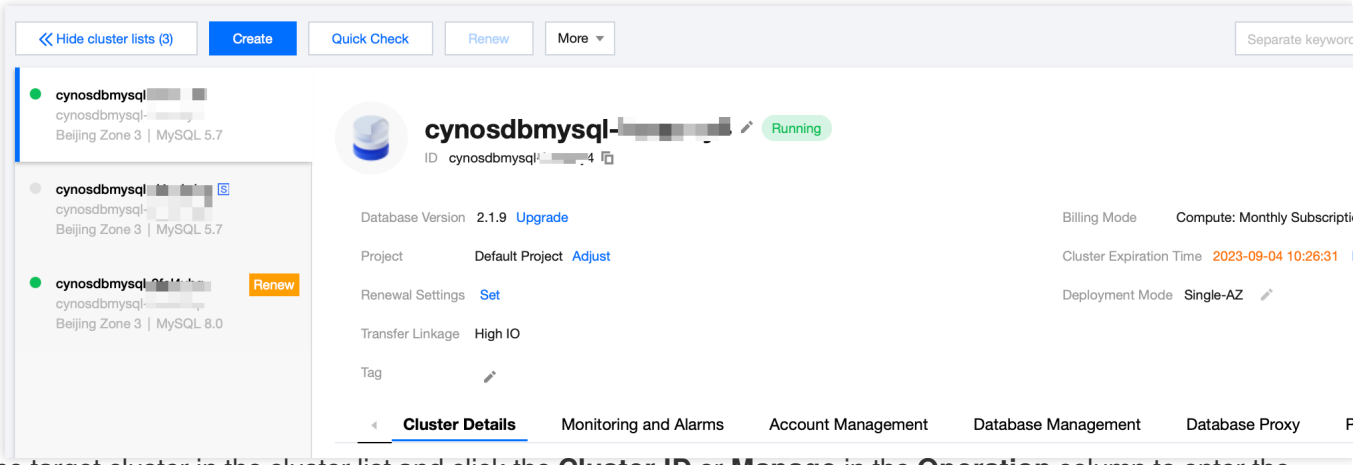
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

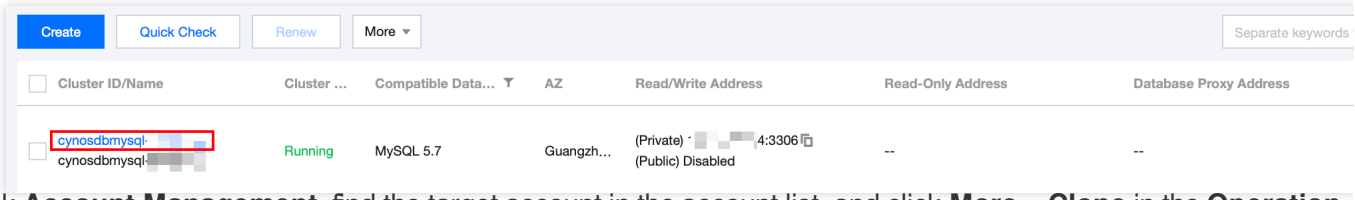
Tab view

List view

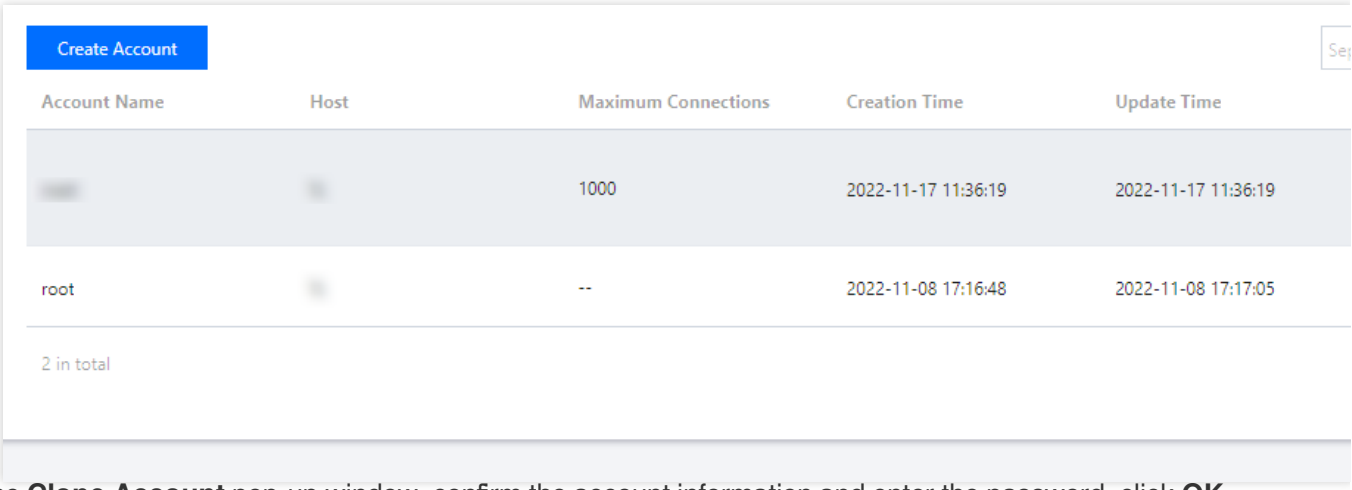
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. Click **Account Management**, find the target account in the account list, and click **More** > **Clone** in the **Operation** column.



4. In the **Clone Account** pop-up window, confirm the account information and enter the password, click **OK**.

Deleting Account

Last updated：2023-08-24 09:46:24

To disable an existing database account, you can delete it in the console.

Note：

A database account cannot be restored once deleted.

To prevent an unintentional deletion from disrupting business, you need to make sure that the database account to be deleted is no longer in use by any applications.

Directions

Note：

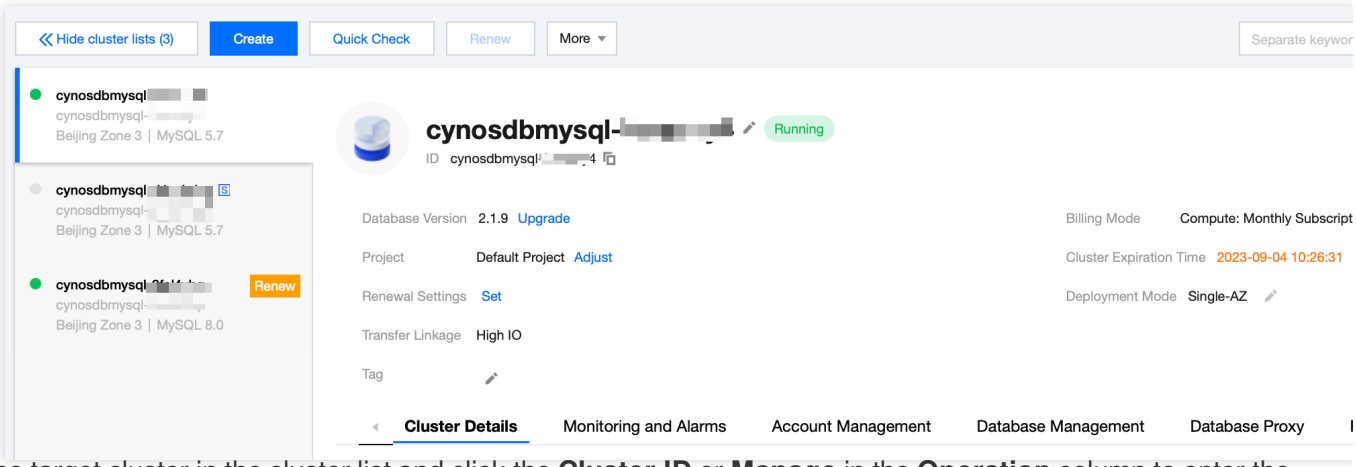
The root account can't be deleted.

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

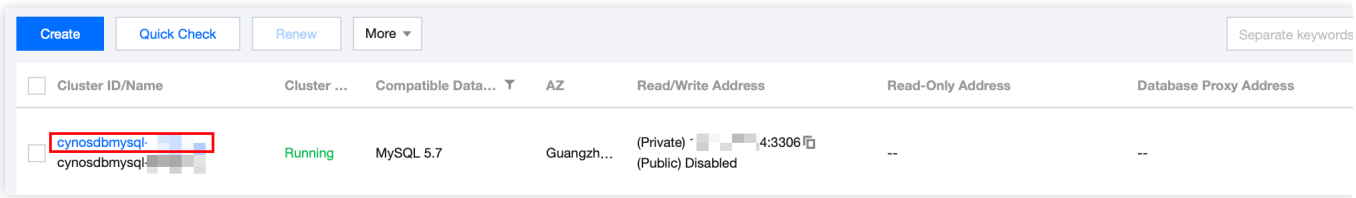
Tab view

List view

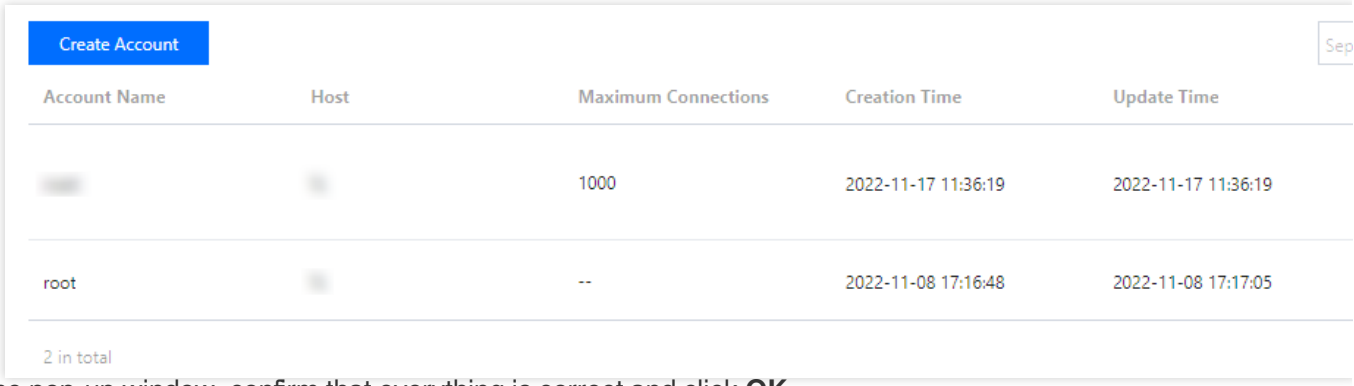
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, find the target account in the account list on the **Account Management** tab, click **More** > **Delete Account** in the **Operation** column.



4. In the pop-up window, confirm that everything is correct and click **OK**.

Database Management
Creating Database

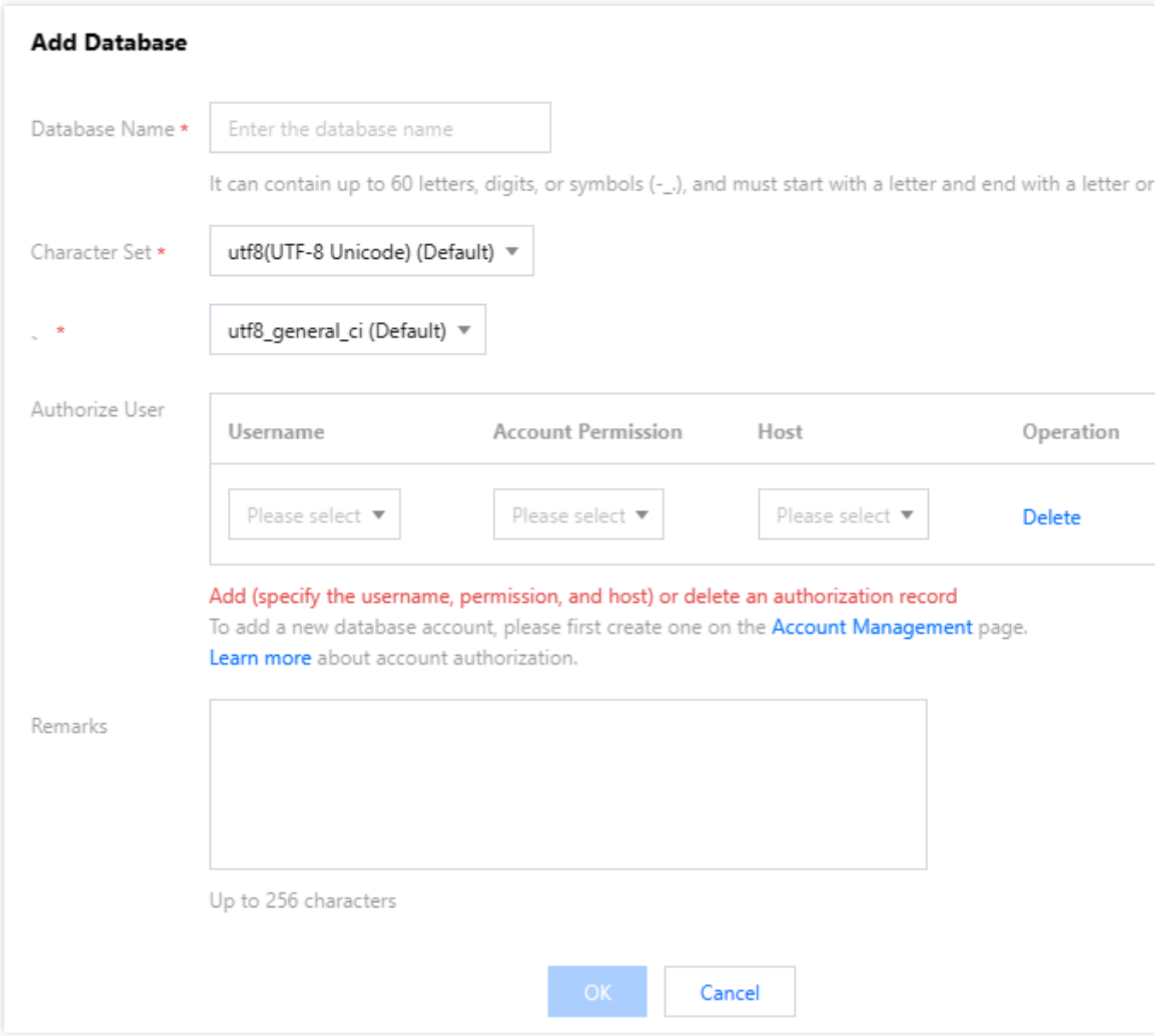
Last updated：2023-11-22 14:30:41

This document describes how to create a database and manage database account permissions on the **Database Management** tab in the TDSQL-C for MySQL console.

Directions

1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list or **Manage** in the **Operation** column to enter the cluster management page.
2. On the cluster management page, select the **Database Management** page and click **Create Database**.

3. In the pop-up window, configure the following parameters and click **OK**:



Parameter	Description
Database Name	Enter the database name, which can contain up to 64 letters, digits, hyphens, and underscores and must start with a letter and end with a letter or digit.
Character Set	Set the character set supported by the database. For more information, see MySQL documentation.
Collation	Set the sorting rule of the database. For more information, see MySQL documentation.
Authorize User	Click Add and select the account to be authorized, permissions to be granted, and host information. You can also delete the authorization record.
Remarks	Enter remarks for the database, which can contain up to 256 characters.

Account authorization details

You can grant an account the read-only, read-write, DML, DDL, and read-only & index permissions of the created database. The permissions and corresponding SQL statements are as follows:

Permission	Authorization Details	Authorization SQL Statement
Read-only	SELECT LOCK TABLES SHOW VIEW	``GRANT SELECT, LOCK TABLES, SHOW VIEW ON `database`.* TO 'account'@'%'``
Read-write	ALL PRIVILEGES	``GRANT ALL PRIVILEGES ON `database`.* TO 'account'@'%'``
DML	SELECT INSERT UPDATE DELETE CREATE TEMPORARY TABLES LOCK TABLES EXECUTE SHOW VIEW EVENT TRIGGER	``GRANT SELECT, INSERT, UPDATE, DELETE, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, SHOW VIEW, EVENT, TRIGGER ON `database`.* TO 'account'@'%'``
DDL	CREATE DROP INDEX ALTER CREATE TEMPORARY TABLES LOCK TABLES CREATE VIEW SHOW VIEW CREATE ROUTINE	``GRANT CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE ON `database`.* TO 'account'@'%'``

	ALTER ROUTINE	
Read-only & index	SELECT INDEX LOCK TABLES SHOW VIEW	``GRANT SELECT, INDEX, LOCK TABLES, SHOW VIEW ON `database`.* TO 'account'@'%'``

Modifying Database Account Permission

Last updated：2023-11-20 14:54:48

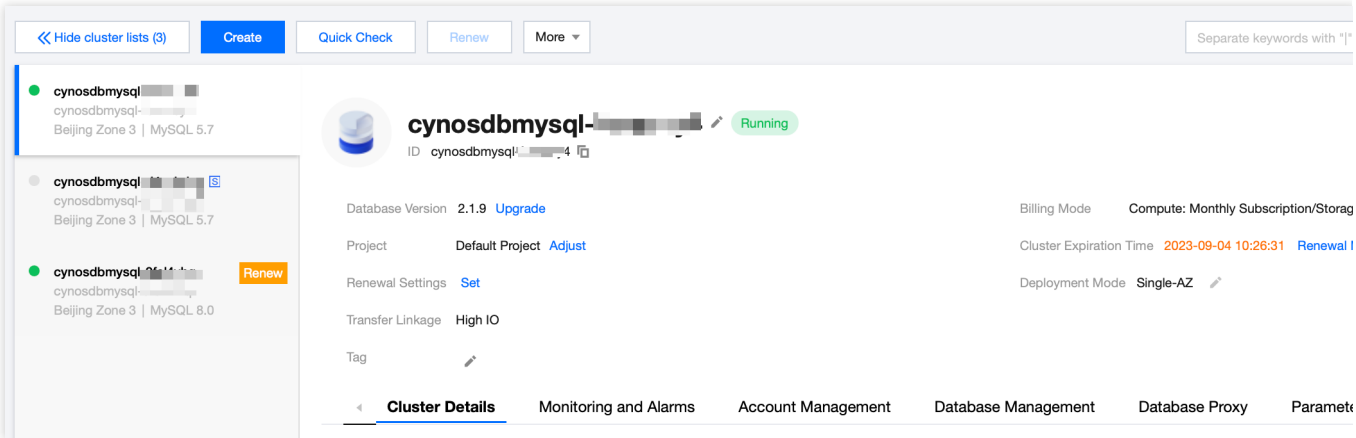
TDSQL-C for MySQL allows you to grant and modify database account permissions. As the root account has the read-write permissions of all databases in a TDSQL-C for MySQL cluster, you cannot modify the permissions of the root account. You can grant other accounts the read-write or read-only permissions for created databases on the database management page.

Prerequisites

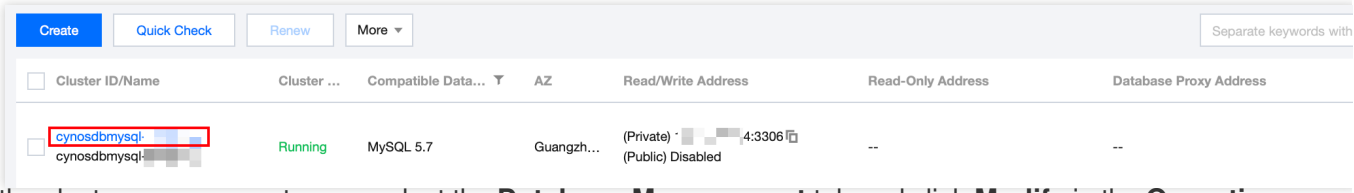
You have created an account other than the root account in your TDSQL-C for MySQL cluster as instructed in Creating Account.
You have created at least one database in your TDSQL-C for MySQL cluster as instructed in Creating Database.

Directions

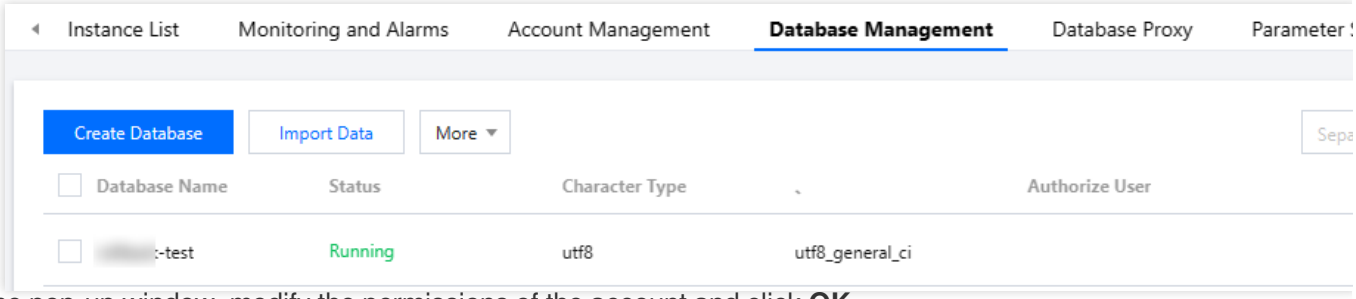
1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.
Tab view
List view
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



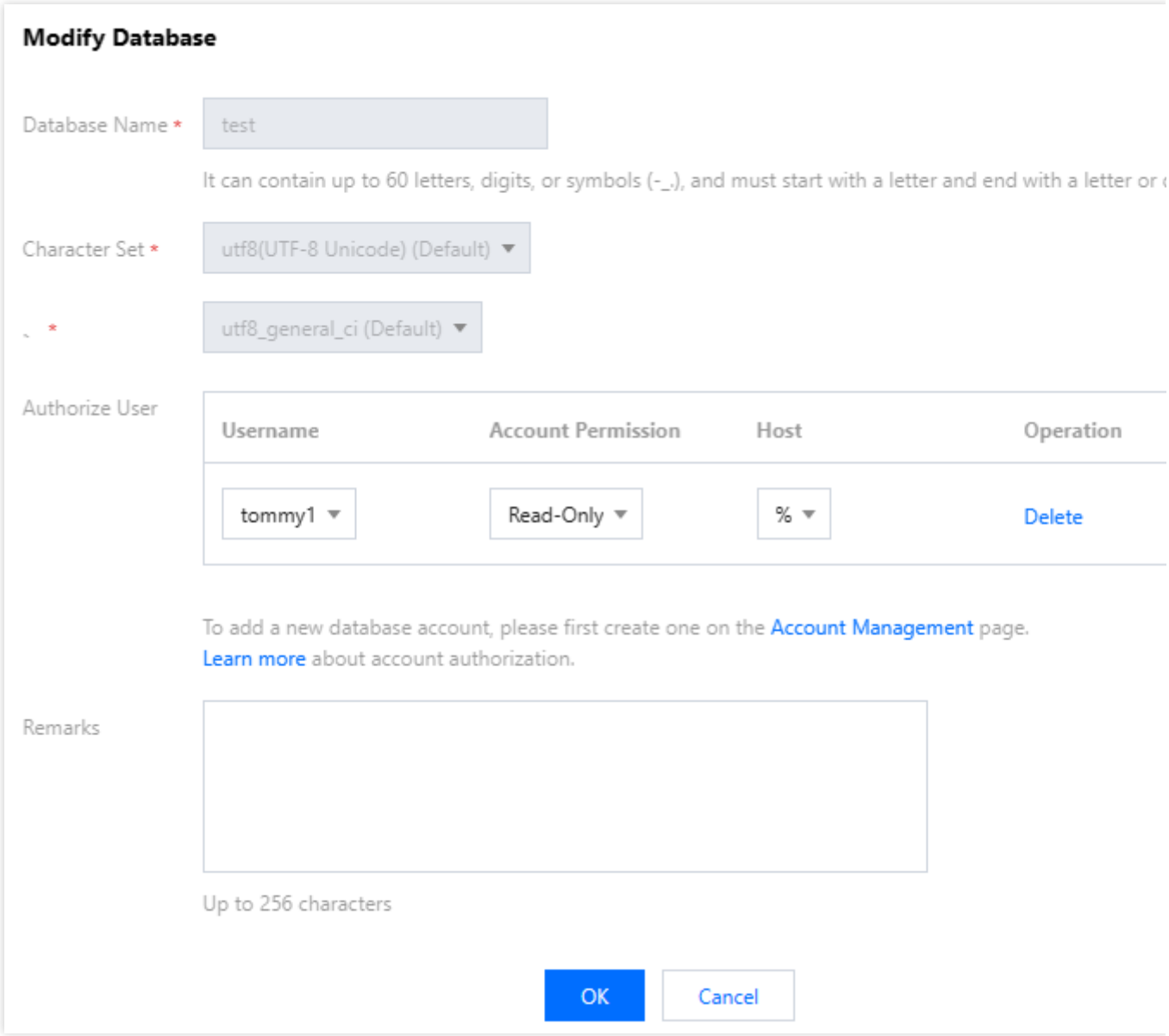
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Database Management** tab and click **Modify** in the **Operation** column of the target database.



4. In the pop-up window, modify the permissions of the account and click **OK**.



Note:

For more information on account authorization, see Creating Database > Account authorization details.

Deleting Database

Last updated：2023-09-12 14:56:41

You can delete databases created in a TDSQL-C for MySQL cluster on the **Database Management** tab in the console.

Prerequisites

You have created at least one database in your TDSQL-C for MySQL cluster as instructed in Creating Database.

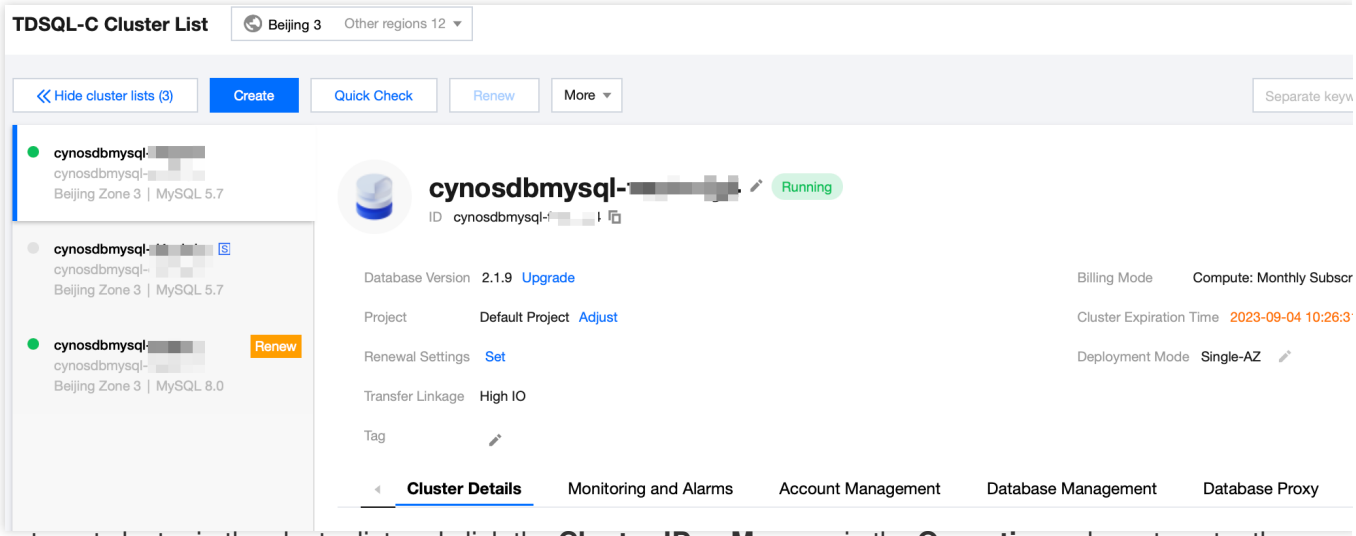
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

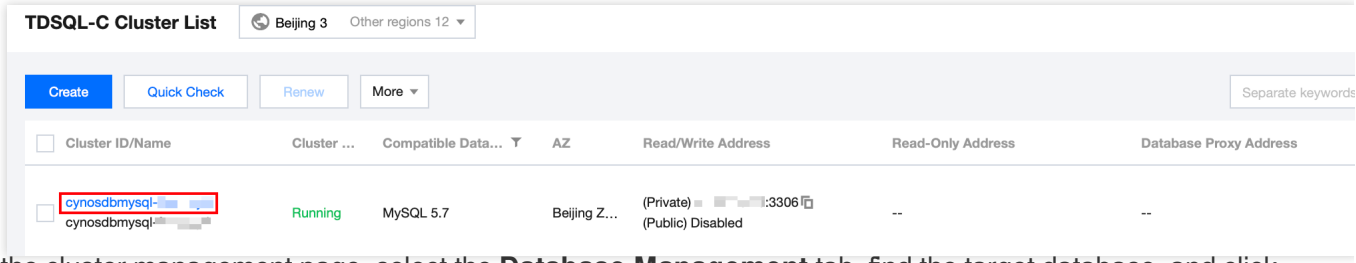
Tab view

List view

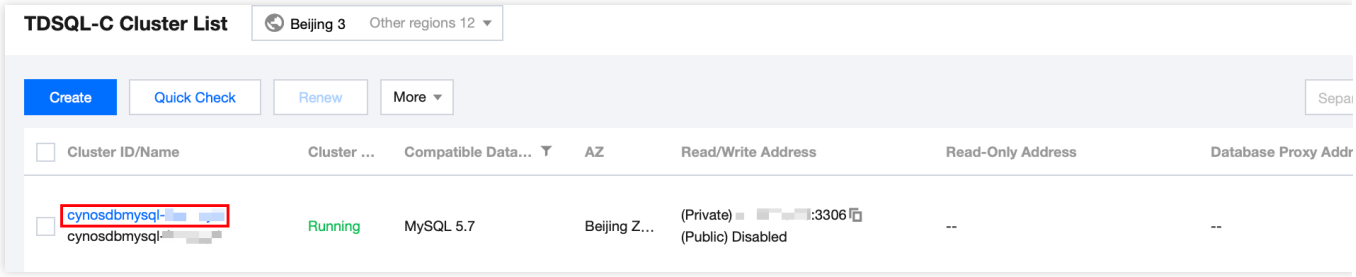
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.

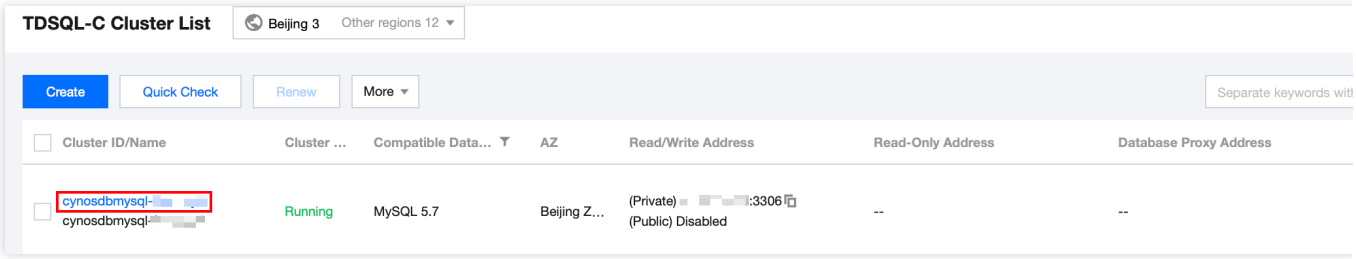


3. On the cluster management page, select the **Database Management** tab, find the target database, and click **Delete** in the **Operation** column.



Note：

You can also batch delete databases by selecting them and clicking **More** > **Batch Delete** above the list.



4. In the pop-up window, confirm that everything is correct and click **OK**.

Database Management Tool
DMC Overview

Last updated：2023-02-07 11:56:37

Database Management Center (DMC) is a one-stop Tencent Cloud database management tool. Services supported include database/table-level operations, real-time monitoring, instance session management, SQL window, and data management.

Description

Data management
It supports database operations such as creating and deleting databases/tables, modifying table structures, and maintaining table data.
Database instance session management
It displays multidimensional instance session information to help you easily view and perform relevant operations.
Database monitoring
It supports multidimensional monitoring of database status, InnoDB row operations, database connections, and traffic.
SQL window
It provides a SQL compiler and can save frequently used SQL templates and custom SQL statements. It can also display and export SQL statement execution results.

Benefits

Ease of use
DMC aims to provide easy-to-use database management services. You can use the convenient operation interface to perform database operations such as creating databases/tables.
Visualization
You do not need to separately install drivers or applications to use DMC. Instead, you just need to log in to the DMC console to manage, monitor, and manipulate databases in multiple dimensions through the visualization management page.
Real-time database status monitoring
DMC can collect database instance data in real time such as CPU utilization, number of connections, and storage capacity usage, so as to intuitively display database instance running conditions.
Real-time database performance monitoring
DMC supports database performance monitoring at the second level, covering multiple monitoring items related to

MySQL status, InnoDB row operations, threads, and network to help you monitor fluctuations in each performance item in real time. In addition, it supports the categorized calculation of sessions and allows you to kill instance sessions, helping you quickly locate causes of exceptions and implement performance optimization.

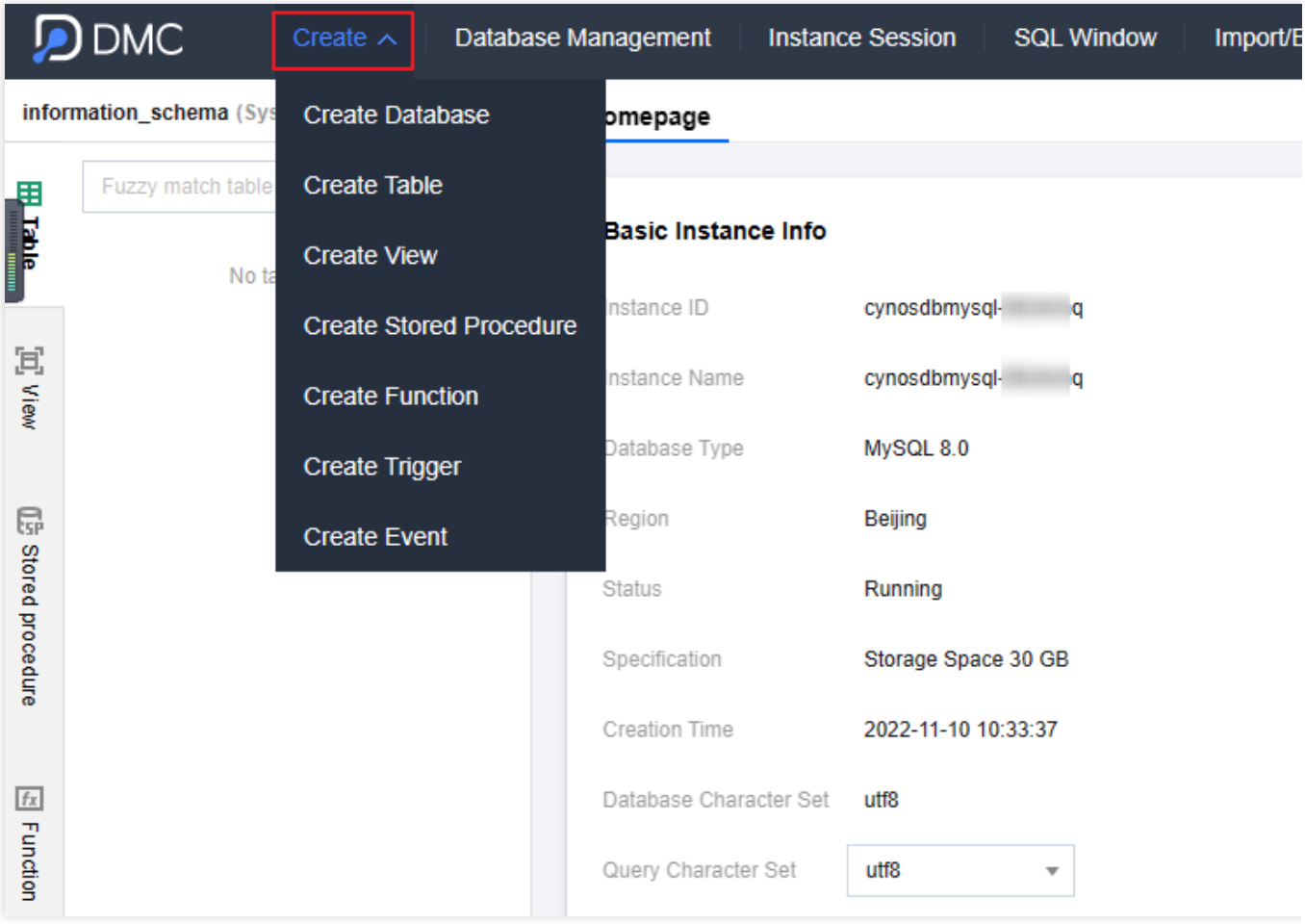
DMC Management

Last updated：2024-07-01 15:11:41

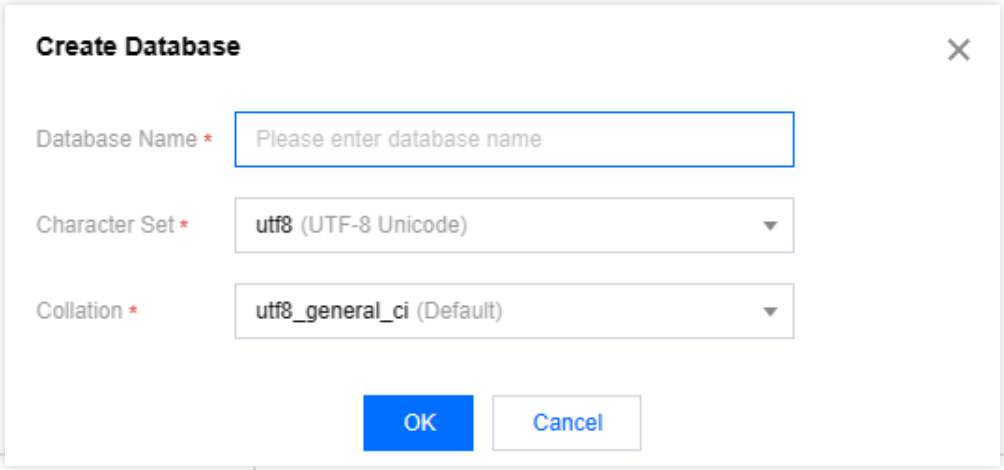
This document describes the features of DMC, such as database/table creation, database management, instance session management, data import/export, and visualized table data editing.

Creating Database

1. By employing one of the following three login methods, log in to the DMC console.
Directly log in to the DMC console.
Or, log in to the TDSQL-C for MySQL console and click **Log In** in the **Operation** column in the cluster list (list view).
Or, log in to the TDSQL-C for MySQL console, click the target cluster in the cluster list to enter the cluster management page, and click **Log In** in the top-right corner (tab view).
2. On the navigation bar, select **Create** > **Create Database** > **Create Database**.



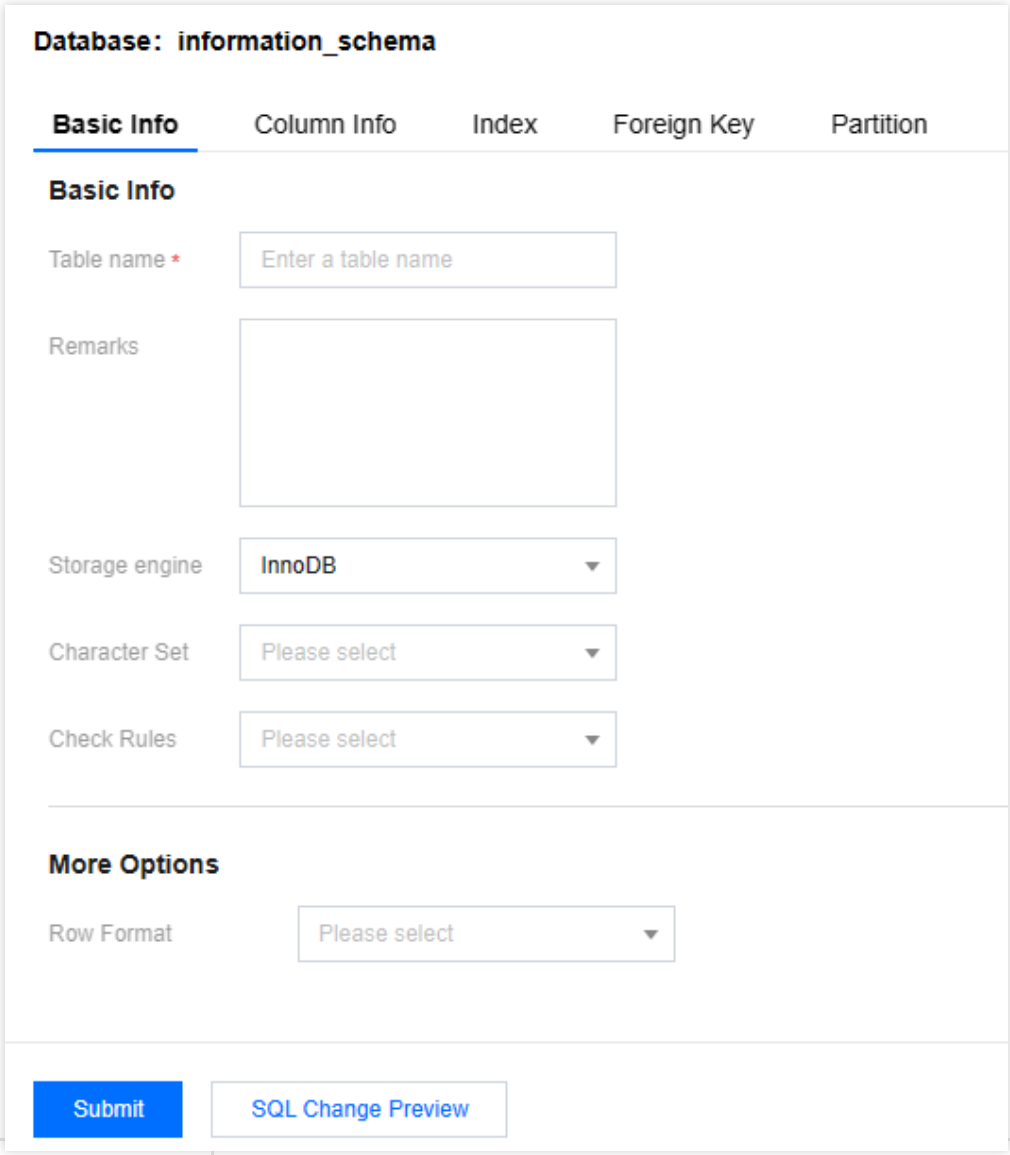
3. In the pop-up window, configure the new database.Click **OK** to complete the creation process.



Parameter	Description
Database Name	Enter the database name, which can contain up to 64 characters, consisting of uppercase and lowercase letters, numbers, hyphens (-), and underscores (_), and must start with a letter and end with a letter or digit.
Character Set	Select the character set. Refer to the console for supported character sets.
Collation	After selecting the character set, the collation will be selected by default. Refer to the console for specific collations.

Creating Table

1. By employing one of the following three login methods, log in to the DMC console.
Directly log in to the DMC console.
Or, log in to the TDSQL-C for MySQL console and click **Log In** in the **Operation** column in the cluster list (list view).
Or, log in to the TDSQL-C for MySQL console, click the target cluster in the cluster list to enter the cluster management page, and click **Log In** in the top-right corner (tab view).
2. First, select the database in which you wish to create a table from the options on the left. Then, navigate to and select **Create** > **Create Table** from the navigation bar.
3. On the page for creating a new table, after completing the necessary configurations as required, click **Submit** .



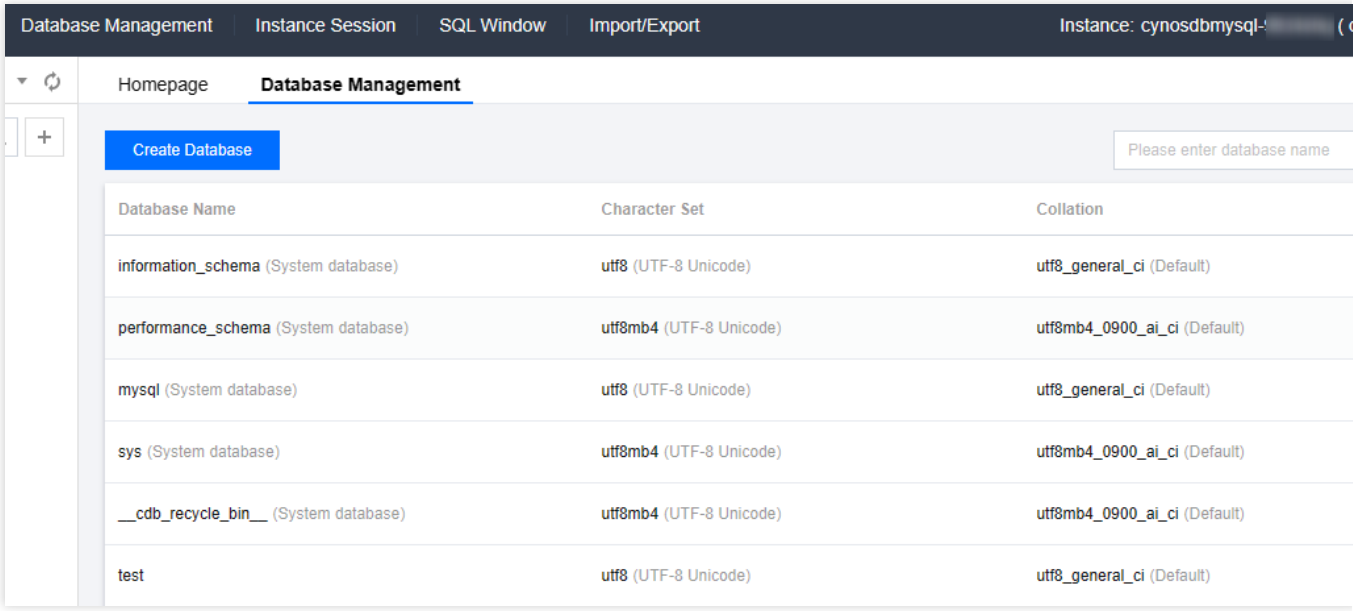
Parameter	Description
Basic Information	On the Basic Information Page, you need to set the table name, add remarks (optional), choose the storage engine (default is InnoDB; for other engines, refer to the actual console), set the character set, set validation rules, set row format, set average row length, set minimum row length, and set maximum row length.
Column Information	On the Column Information Page, you can use add, delete, insert, move up, and move down operations to configure the table's columns.
Index	On the Index Page, you can add new indexes. It helps quickly locate data rows with specific data values. Refer to the console for the specific supported index types and indexing methods.
Foreign Key	On the Foreign Key Page, you can add new foreign keys. Foreign keys are used to establish constraints on relationships between tables. A foreign key constraint specifies a column in one table whose values must match the values in a column of another table. This matched column is usually the primary key of the other table.

Partition	On the Partition Page, you can improve query performance and simplify data maintenance and management by dividing table data into multiple partitions.

Note:
For more information on the character set and sorting rule, see MySQL documentation.

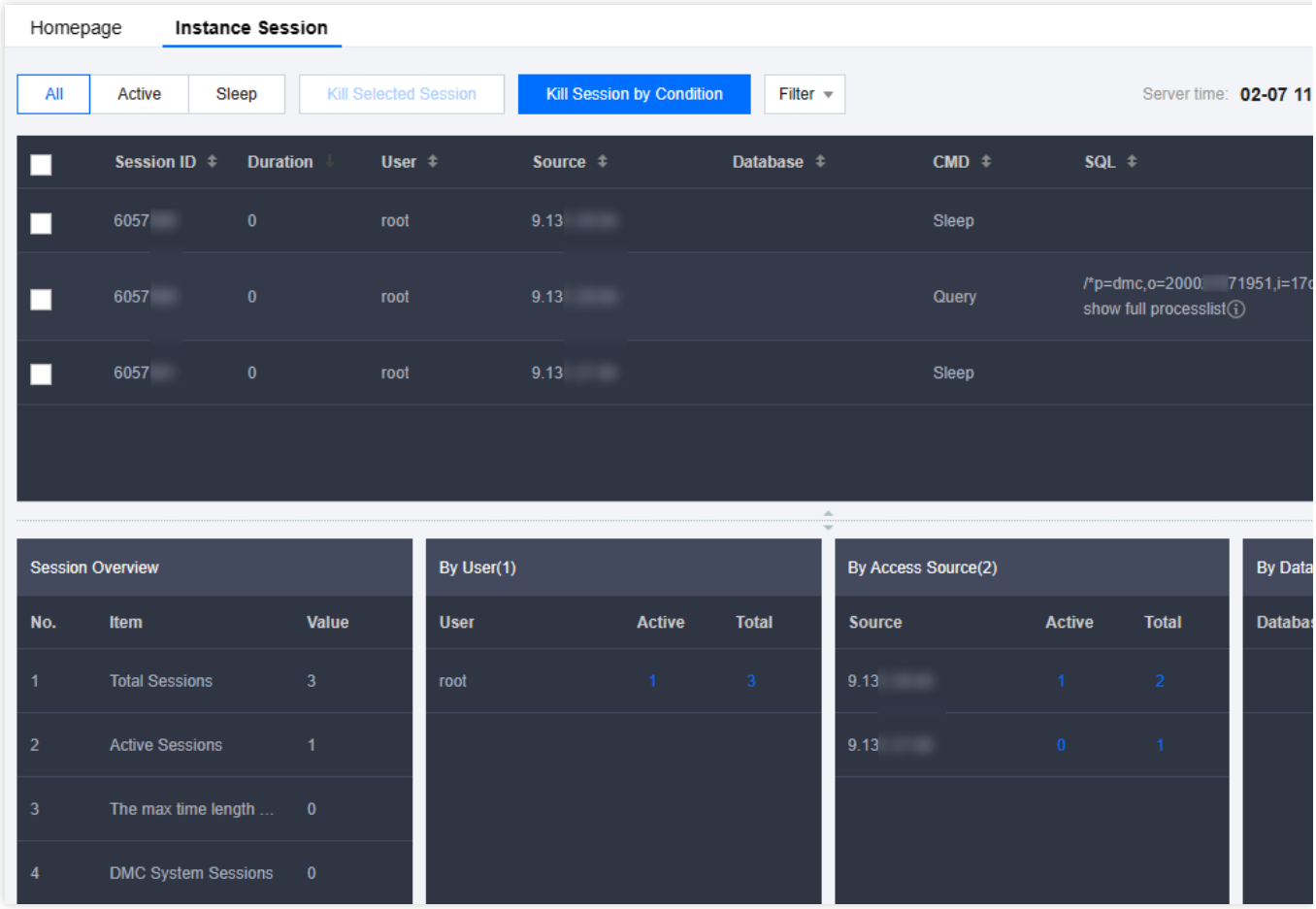
Database management

Log in to the DMC console, select **Database Management** on the navigation bar at the top, and create, edit, or delete databases on the displayed page.



Instance session management

Log in to the DMC console, select **Instance Session** on the navigation bar at the top, and enter the instance session management page. You can view the details of all database sessions in the instance from four dimensions: session overview, users, access sources, and databases.
DMC allows you to kill sessions, facilitating your session management.



SQL window

Log in to the DMC console, select **SQL Window** on the navigation bar at the top, or click **SQL Operation** on the **Operation** menu on the left sidebar to access the SQL window, which supports the following features:

Run SQL commands and view results

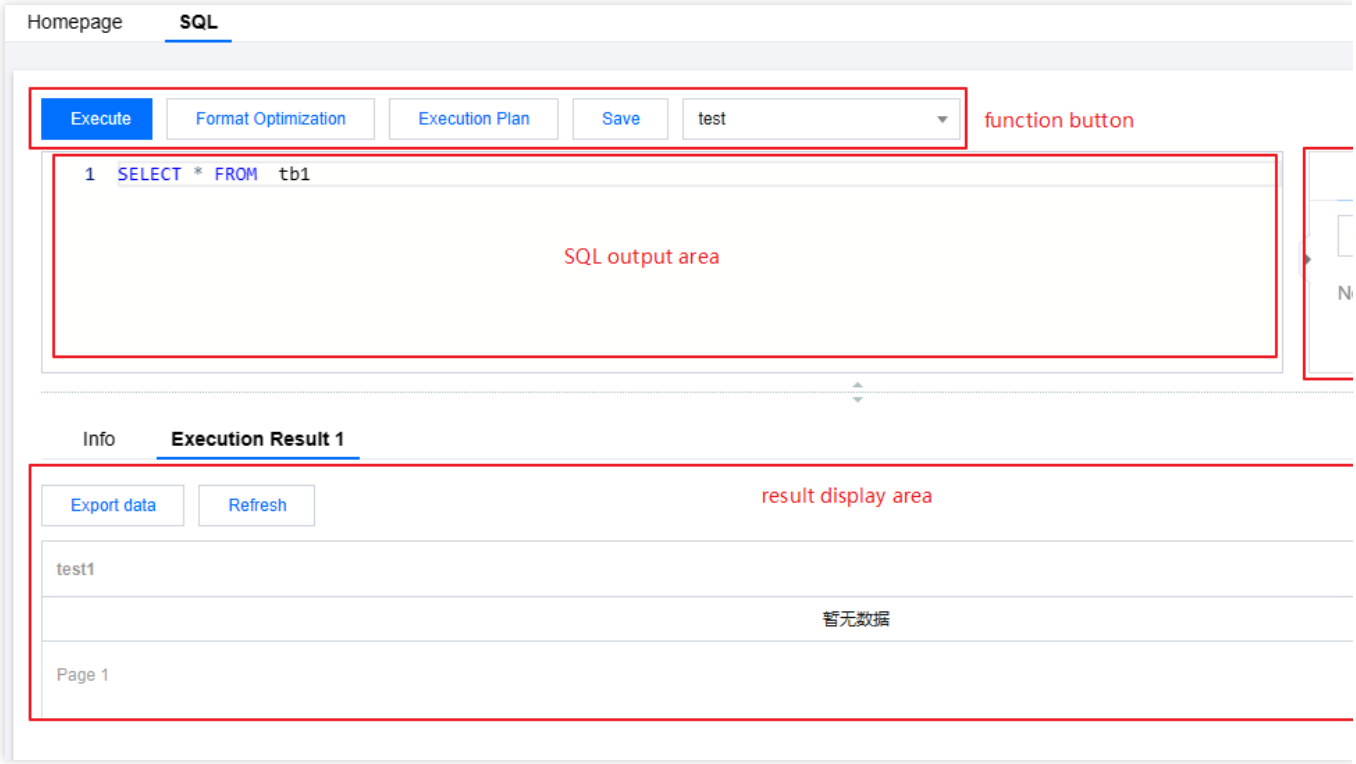
Optimize SQL statement formats

View SQL command execution plans

Save commonly used SQL statements

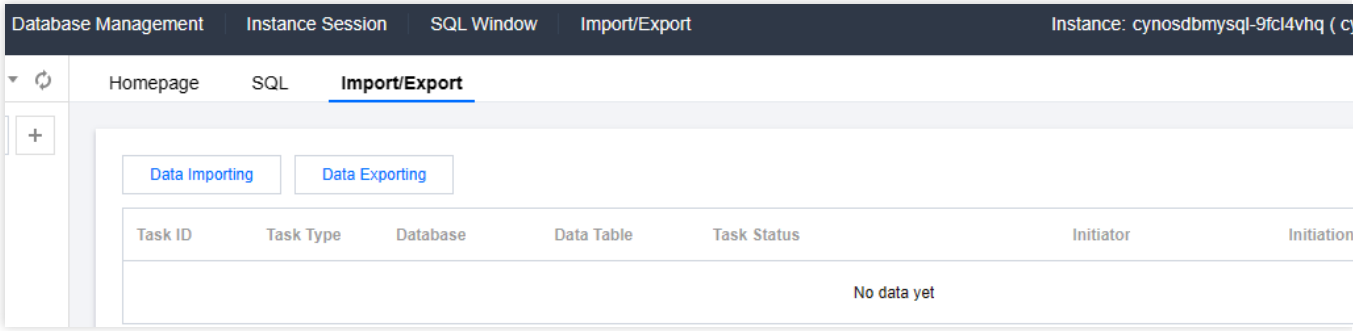
Use SQL templates

Export SQL statement execution results



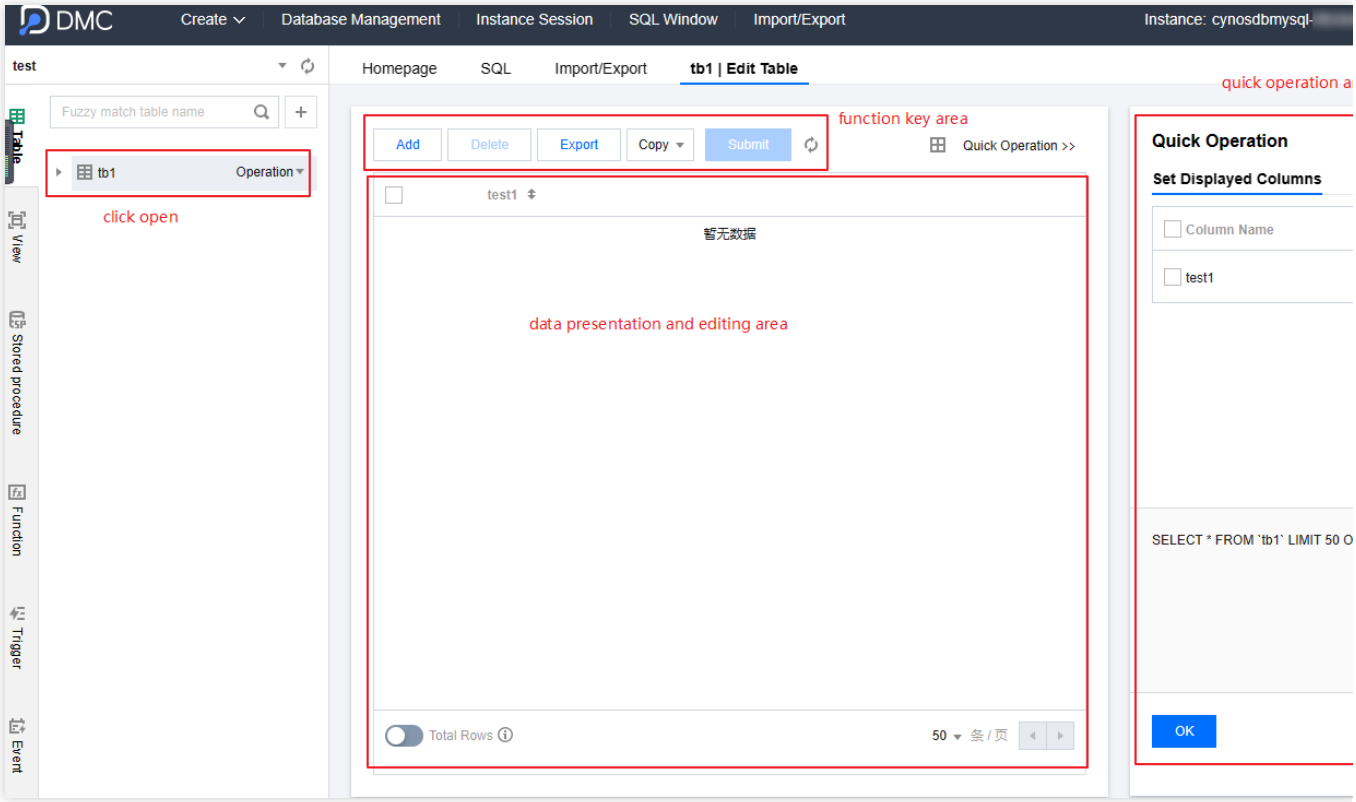
Import/Export

Log in to the DMC console, select **Import/Export** > **Data Importing** or **Data Exporting** on the navigation bar at the top, and you can import data into or export data from a database.



Visualized table data editing

DMC for TDSQL-C for MySQL supports inserting, deleting, and updating data. You can click a table in the table list on the left sidebar to insert, delete, and update its data in batches in the right pane, and then click **OK** in the **Quick Operation** pane to preview the SQL statements and implement the modification.



Columnar Storage Index (CSI)
Introduction
Feature Overview

Last updated：2024-06-07 16:50:50

Background

As more and more users migrate their core systems to the cloud, the amount of user data is rapidly expanding, and business logic is becoming increasingly complex. Although the cloud-native architecture of TDSQL-C for MySQL can handle transaction requests competently, queries based on row store may not always meet response time requirements of diverse analytical queries from users. A standard solution is to use data synchronization tools to replicate the data of the transaction system into a separate analysis system and route the user's query and analysis requests to this system. However, this solution harbors issues such as additional costs for data synchronization and independent analysis clusters, as well as problems related to real-time performance and data consistency of synchronization.
To address these challenges, TDSQL-C for MySQL provides the **Column Store Index (CSI)** feature that stores, retrieves, and manages data in a column-wise format, achieving higher query performance and data compression ratio.

Advantages

Compared to conventional row-based storage, CSI can achieve great query performance improvement using column-based data storage and query processing and significantly improve the data compression over the uncompressed data size.

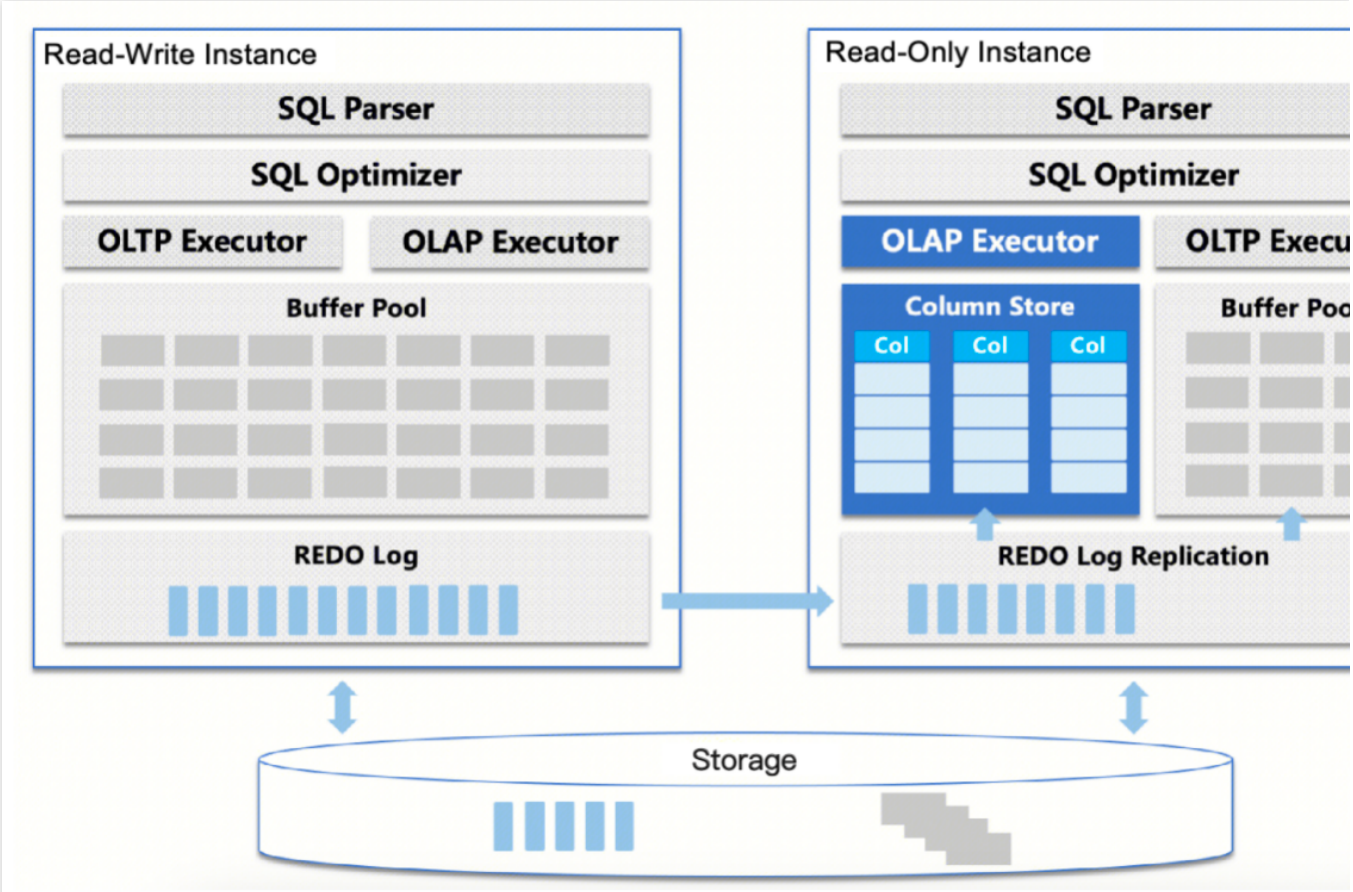
Supported Versions

The kernel version is TDSQL-C for MySQL version 8.0 3.1.14 or later.
Note:
For read-only instances that meet the version requirements, the CSI feature can be enabled only on those with four or more CPU cores.

Application Scenarios

Scenarios requiring real-time analysis of online data, such as online report and data dashboard.
Scenarios involving analytical queries for large volumes of data.

Technical Principle



The CSI feature of TDSQL-C for MySQL is based on the following three key technologies:

1. Row-column mixed storage of data

TDSQL-C for MySQL stores data in a row-wise format by default. However, column-based storage can better support data analysis and query. Based on a unified architecture, column store indexes can be created for row store tables, thereby achieving mixed data storage.

2. Generation and scheduling of hybrid row/column scheduling plans

TDSQL-C for MySQL uses the optimization statistics and cost-based optimization (CBO) model to assess the cost of plans. It adapts column store indexes to the current row store optimization model and includes the indexes in the CBO search space. In this way, it pushes down execution plan segments for mixed scheduling.

3. Efficient computation

Data is stored in data blocks in a column-wise format. During computing, only the data of required columns are retrieved, thus significantly reducing I/O layer overheads, particularly in scenarios involving large wide tables. Additionally, columnar data is stored in a more compact manner in the memory. Operators can process the column data of multiple rows in a batch during computing to fully utilize the cache affinity, thereby improving computational efficiency.

Supported Statements and Restrictions

Last updated：2024-06-07 16:53:57

This document describes the supported statements of and restrictions on the Column Store Index (CSI) feature.

Prerequisites

The kernel version is TDSQL-C for MySQL version 8.0 3.1.14 or later.

Note:

For read-only instances that meet the version requirements, the CSI feature can be enabled only on those with four or more CPU cores.

Supported Statements

TDSQL-C for MySQL supports queries using SQL statements involving the following scenarios. More scenarios will be gradually supported in the future.

Scenario	Description
Single-table scan	Forward and reverse scans are supported for full-table scan, index scan, index range scan, and other types of single-table scans.
Multi-table join	A variety of join algorithms such as nested loop join, hash join, and sort-merge join are supported for multi-table joins.
Subquery	Non-correlated subqueries are supported.
Data type	Queries of various types of data are supported, including integer, character, floating-point, and datetime data.
Operation formula	Arithmetic expressions (+, -, *, %, /, |, &), conditional expressions (<, <=, >, >=, <>, between/and, In), logical operations (or, and, not), general functions (character, integer, and datetime functions), and aggregate functions (count, sum, avg, min, max) are supported.
Viewing of execution plan pushed down	The EXPLAIN statement can be used to view the execution plan that is pushed down. However, the COLUMNSTORE flag is only displayed when using the EXPLAIN FORMAT=TREE statement.

Restrictions

TDSQL-C for MySQL will continue to enhance the CSI-based querying capability. Currently, restrictions on the CSI feature are as follows:

Index Creation

Only one column store index can be created for each table.
Column store indexes cannot be created when Generated Column, BLOB, TEXT, JSON, BIT, and GEOMETRY are used.
The COPY algorithm cannot be used to create or delete a column store index or to delete an index first and then add again.
Column store indexes cannot be created for non-InnoDB tables.

DDL-Based Operation on Tables with Column Store Indexes

Partitioning operations on tables with column store indexes are not supported.

Use of Column Store Indexes in SELECT Statements

DML SELECT statements, including INSERT ... SELECT and REPLACE ... SELECT, are not supported.
Common table expressions (CTEs) are not supported.
Window functions are not supported.
WITH ROLLUP is not supported.
UNION is not supported.
Transactions with isolation levels other than ReadCommit are not supported.
Statements for addling locks during data read, such as SELECT FOR UPDATE and SELECT ... FOR SHARE, are not supported.
Stored Procedures are not supported.
SQL_MODE statements, such as MODE_PAD_CHAR_TO_FULL_LENGTH and MODE_INVALID_DATES, are not supported.
GIS-related functions such as SP_WITHIN_FUNC and ST_Distance are not supported.
Custom functions created by users are not supported.
XML functions such as XML_STR are not supported.
Lock-related functions, such as IS_FREE_LOCK, IS_USED_LOCK, RELEASE_LOCK, RELEASE_ALL_LOCKS, and GET_LOCK, are not supported.

Precautions

CSI-based queries may have the following issues:
Unordered Result Set

When no ORDER is specified, the result set order of CSI-based queries may be different from that of B-tree queries due to the data storage difference. In fact, this is in accordance with ANSI standards. However, it is better to specify a sorting rule to obtain results consistent with those of serial queries if the user is sensitive to the result set order.

Precision and Truncation Issues

Use of CSI can lead to subtle differences in the final results, as the storage and computation precision of certain data types may not completely be the same as MySQL, potentially causing precision or truncation issues of the floating-point data.

Increase in Errors or Alerts

For queries that report errors or alerts during serial execution, the overall number of errors or alerts may increase when CSI is used for queries because each working thread may report errors or alerts during parallel execution.

Impact on Resources

The buffer pool will occupy system memory space when CSI is used.

The parallel execution of CSI-based queries will consume more threads, memory, and I/O resources, compared with serial queries that use only one thread.

Utilization of Columnar Storage Indexes
Basic Settings

Last updated：2024-07-18 14:10:07

This document describes operations related to Column Store Index (CSI).

Prerequisites

The kernel version of TDSQL-C for MySQL 8.0 is 3.1.14 or later.

Note:

For read-only instances that meet the version requirements, the CSI feature can be enabled only on those with four or more CPU cores.

Enabling or Disabling CSI

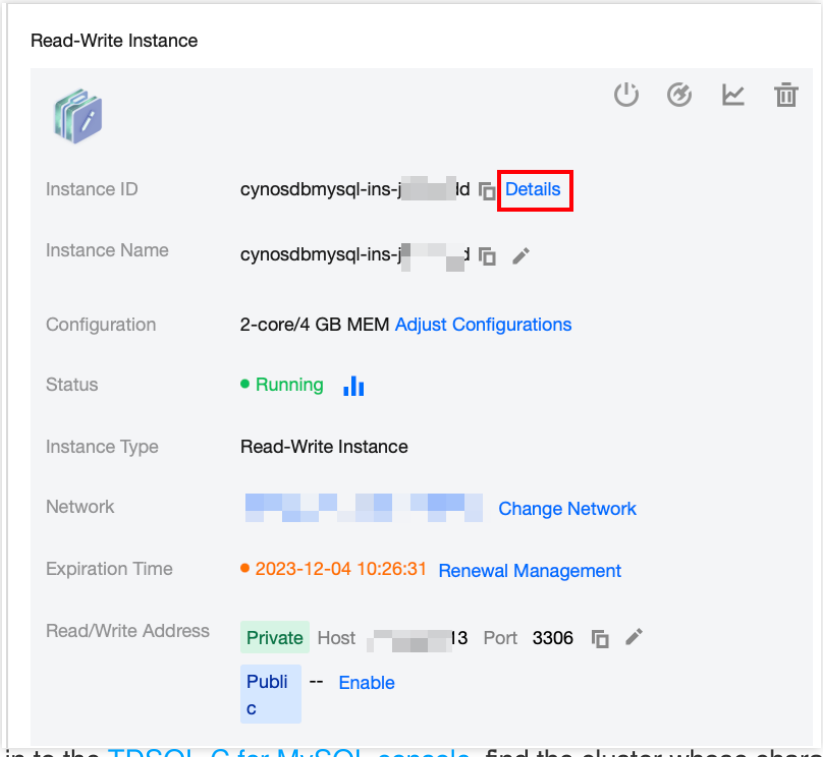
1. On the cluster list page, go to the instance details page according to the view mode.

Tab View

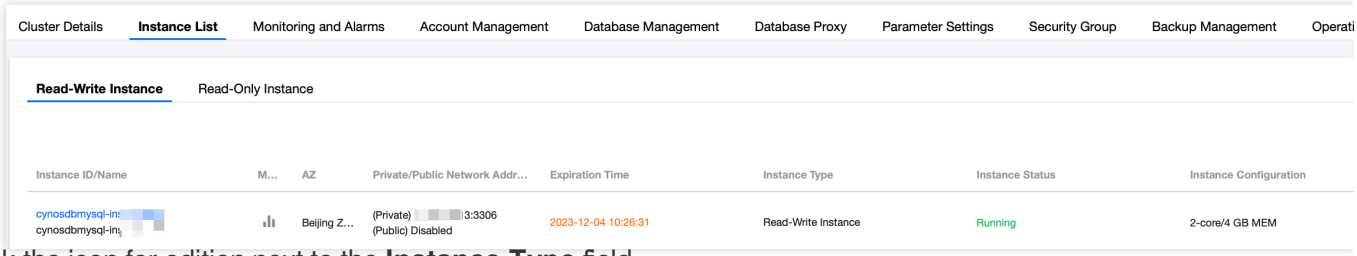
List View

1. Log in to the TDSQL-C for MySQL console, and click the **target cluster** in the cluster list on the left to go to the cluster management page.

2. On the **Cluster Details** tab, click **Details** of the target instance next to the instance ID to go the instance details page.



1. Log in to the TDSQL-C for MySQL console, find the cluster whose character set needs modification in the cluster list, and click the **Cluster ID** to go to the cluster management page.
2. On the cluster management page, select the **Instance List** tab. Locate the read-write instance or read-only instance for which you intend to enable or disable CSI, and click the **instance ID** to access the instance details page.



2. Click the icon for edition next to the **Instance Type** field.
3. In the pop-up window, change the instance type and then click **Check**.

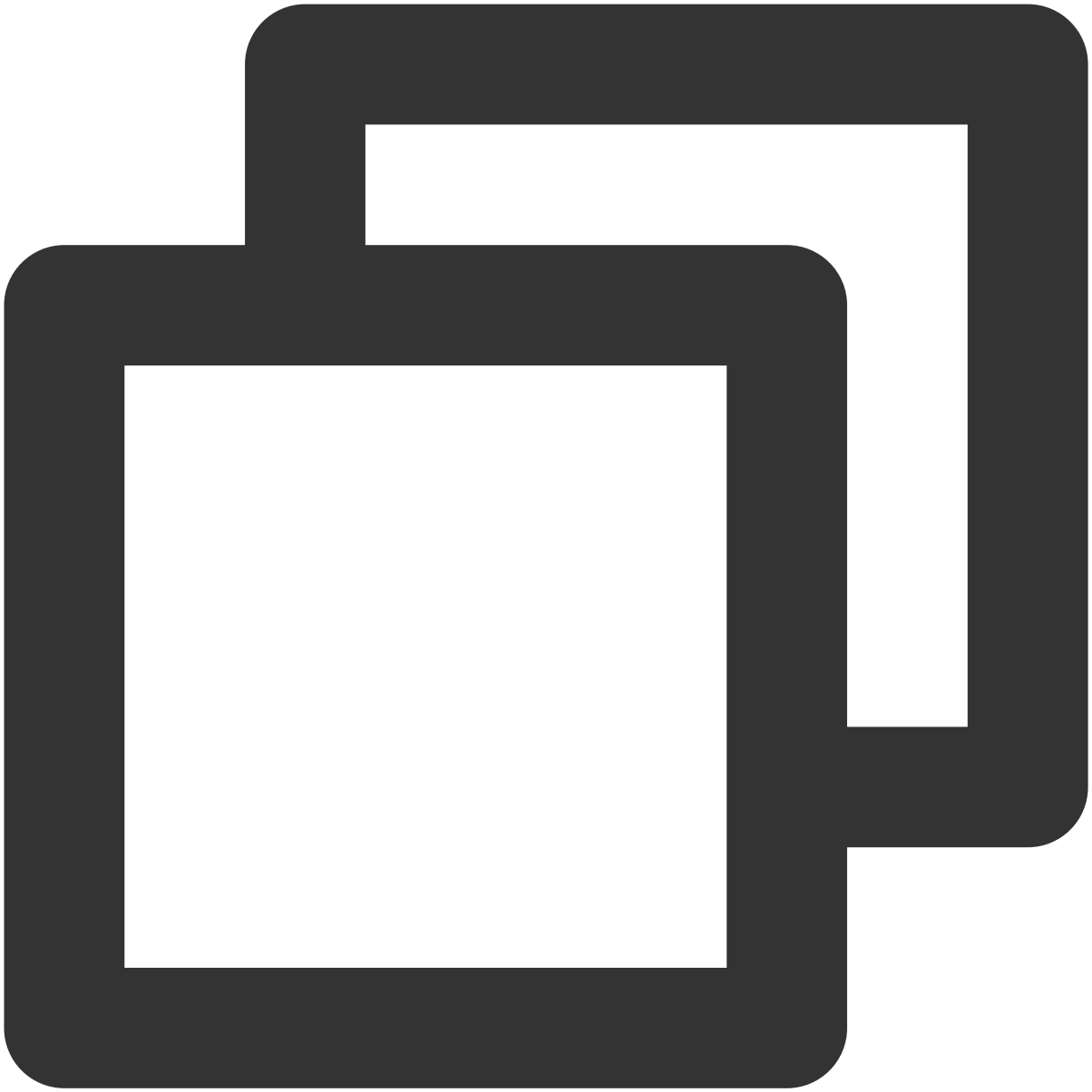
If the instance type changes from **Line Store** to **Mix Store**, the CSI feature is **enabled**.

If the instance type changes from **Mix Store** to **Line Store**, the CSI feature is **disabled**.

Creating Column Store Indexes

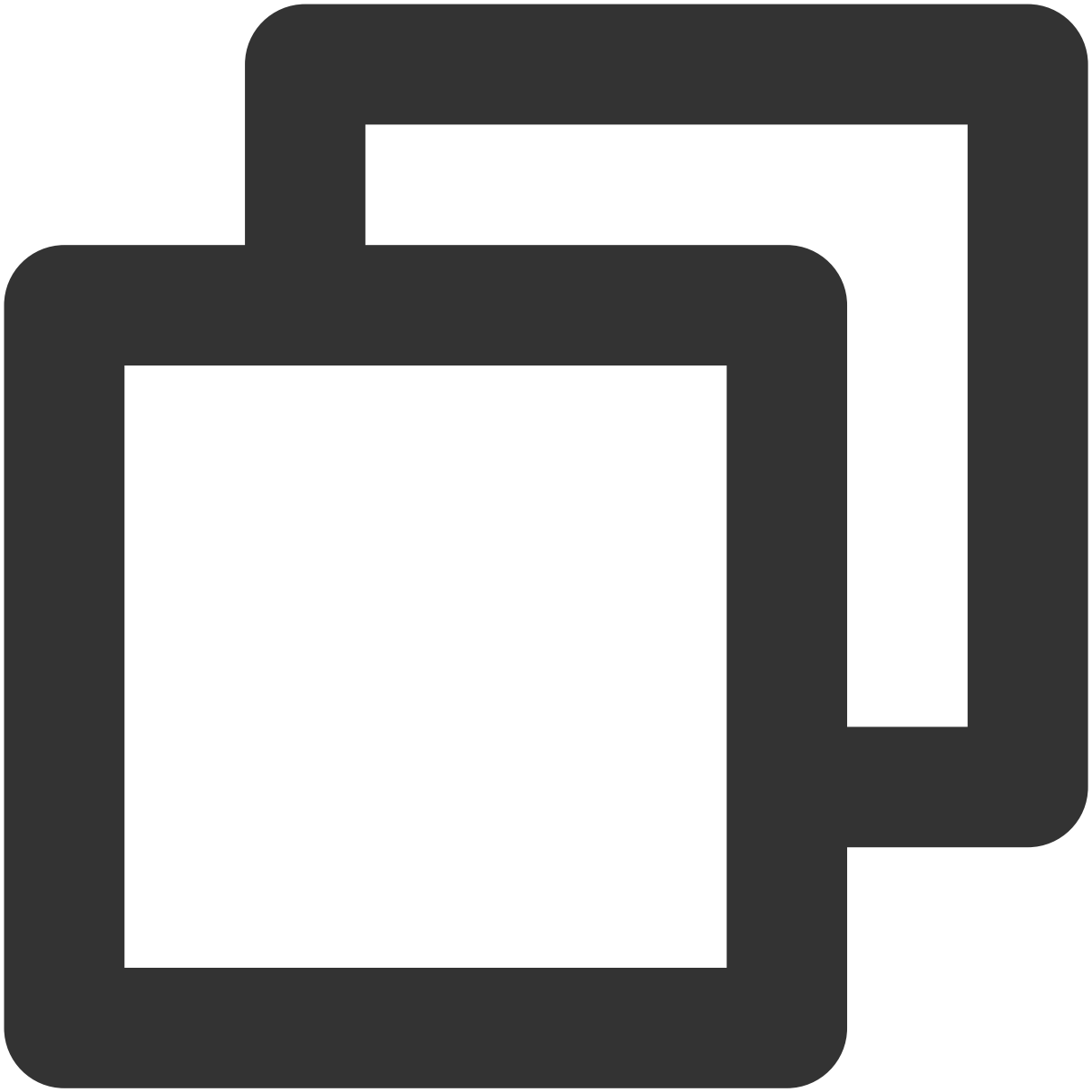
After the CSI feature is enabled, the following commands can be used to create column store indexes:

1. Create indexes when creating the table.



```
CREATE TABLE table_name (col1, col2,... COLUMNSTORE INDEX [index_name] [(col1, col2
```

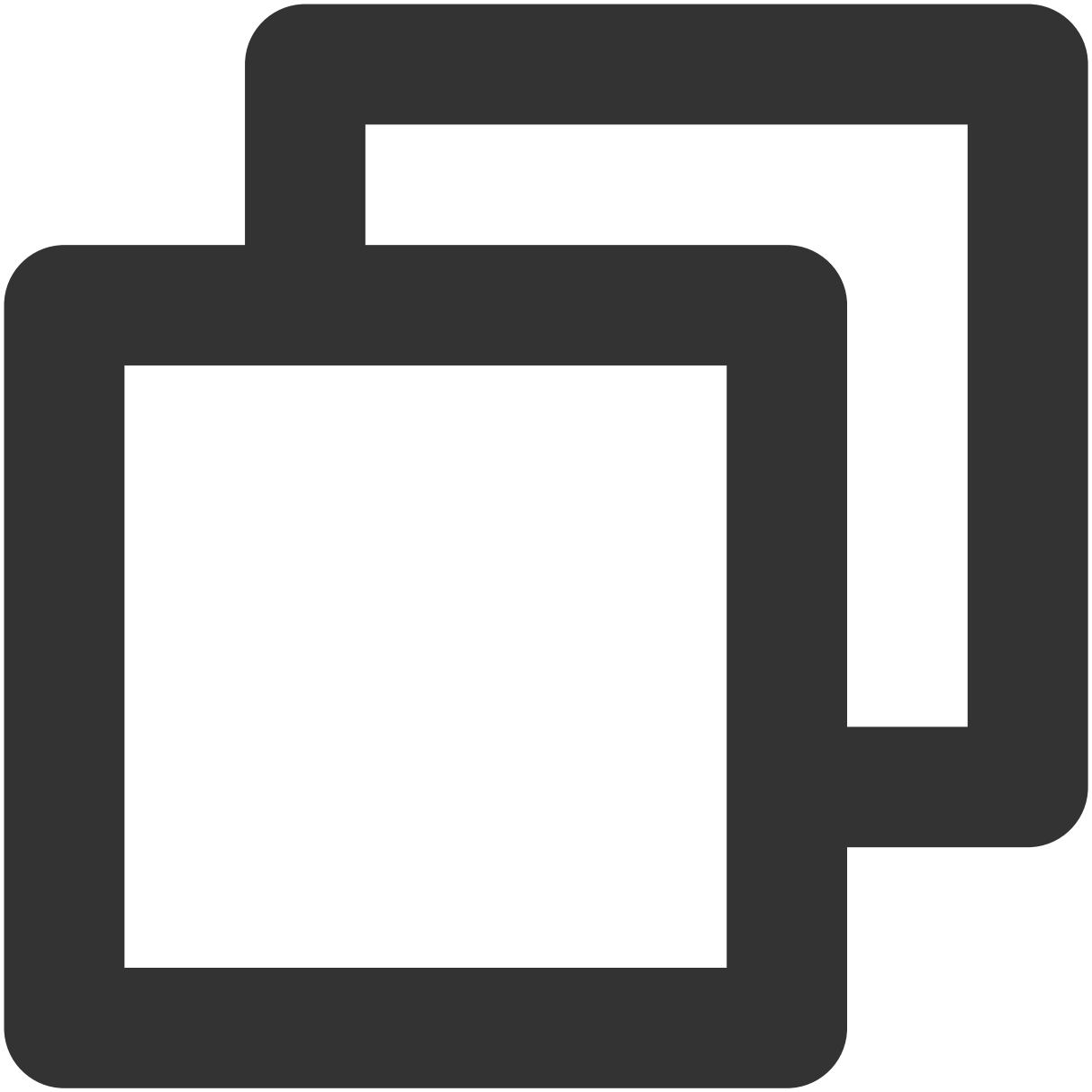
2. Create indexes after the table is created.



```
CREATE COLUMNSTORE INDEX [index_name] ON table_name[(col1, col2,...)];
ALTER TABLE table_name ADD COLUMNSTORE INDEX [index_name] [(col1, col2,...)];
```

Deleting Column Store Indexes

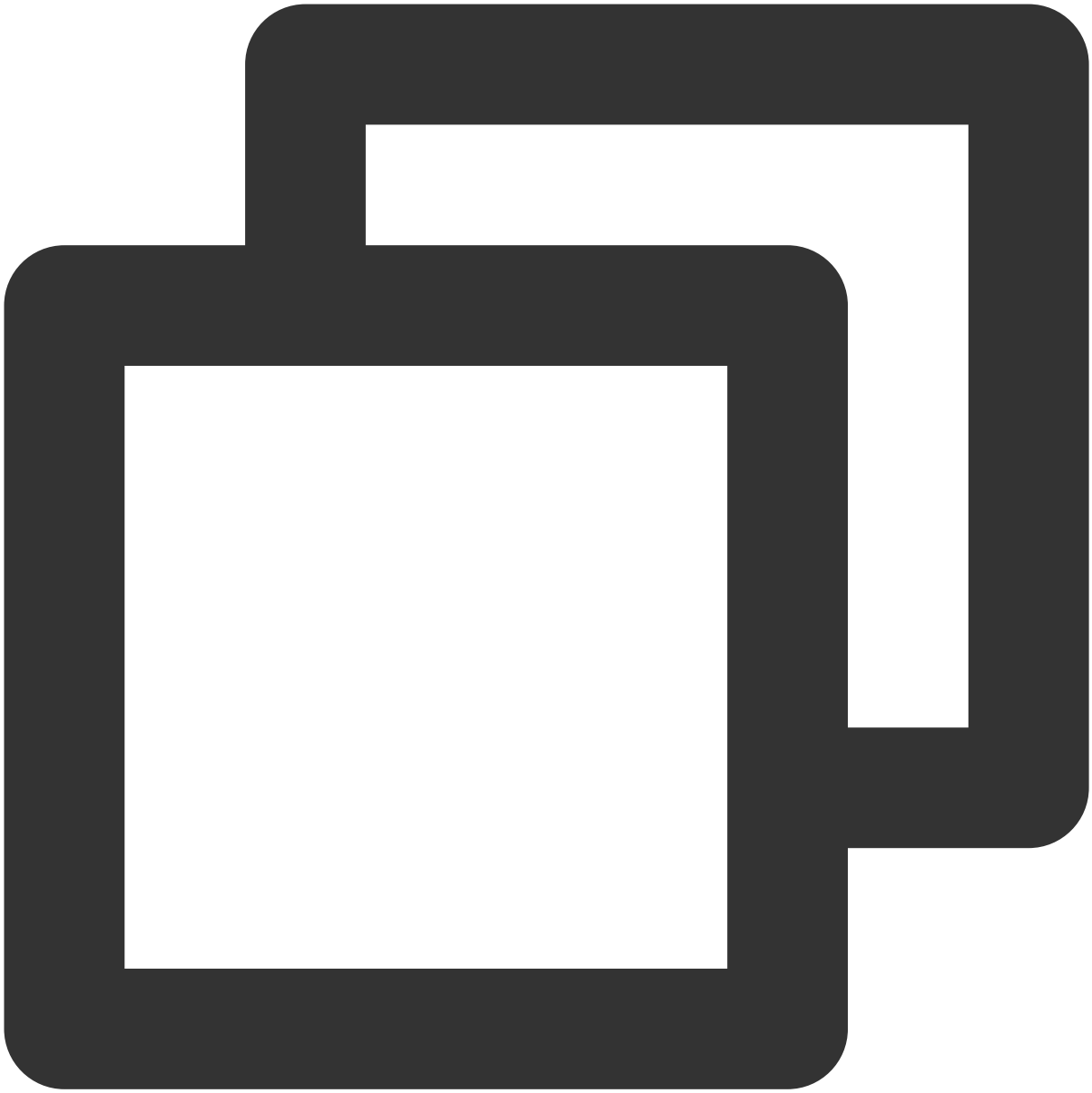
After the CSI feature is enabled, the following command can be used to delete column store indexes:



```
ALTER TABLE table_name DROP INDEX index_name;
```

Renaming Column Store Indexes

After the CSI feature is enabled, the following command can be used to rename column store indexes:

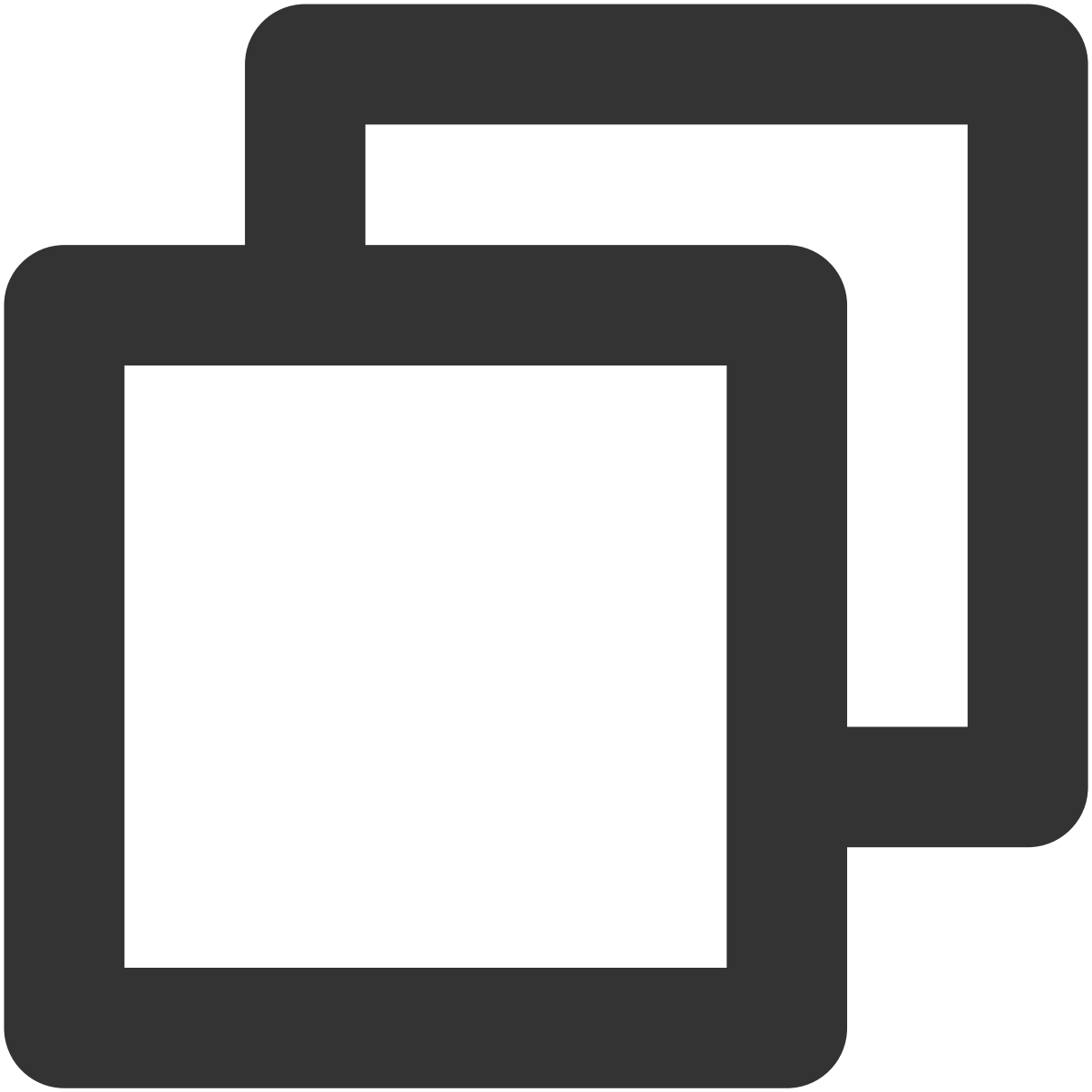


```
ALTER TABLE table_name RENAME index old_index_name to new_index_name;
```

Using HINT Statements for Column Store Indexes

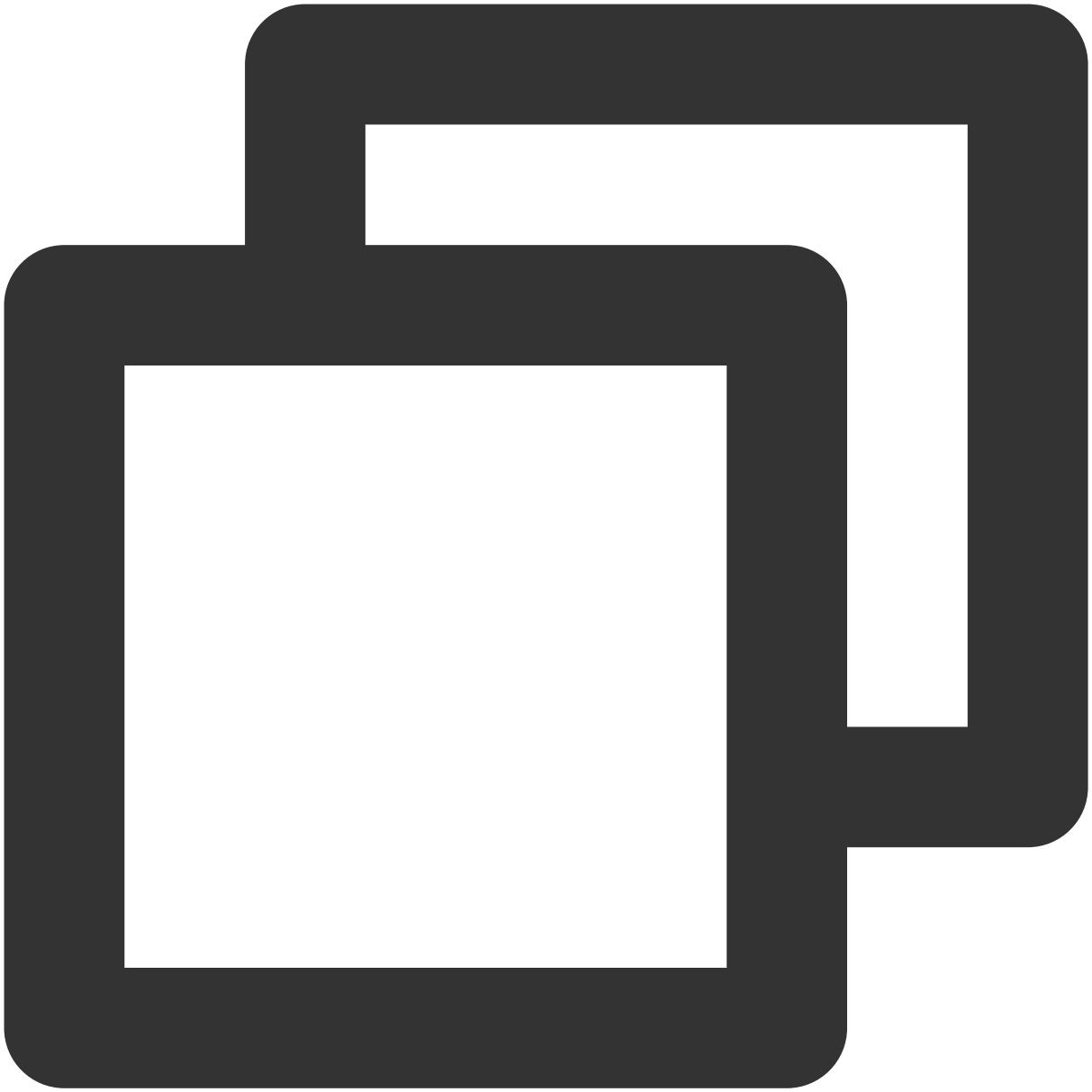
1. Enforce statements to use row store indexes or column store indexes.

Enforce statements to use row store indexes.



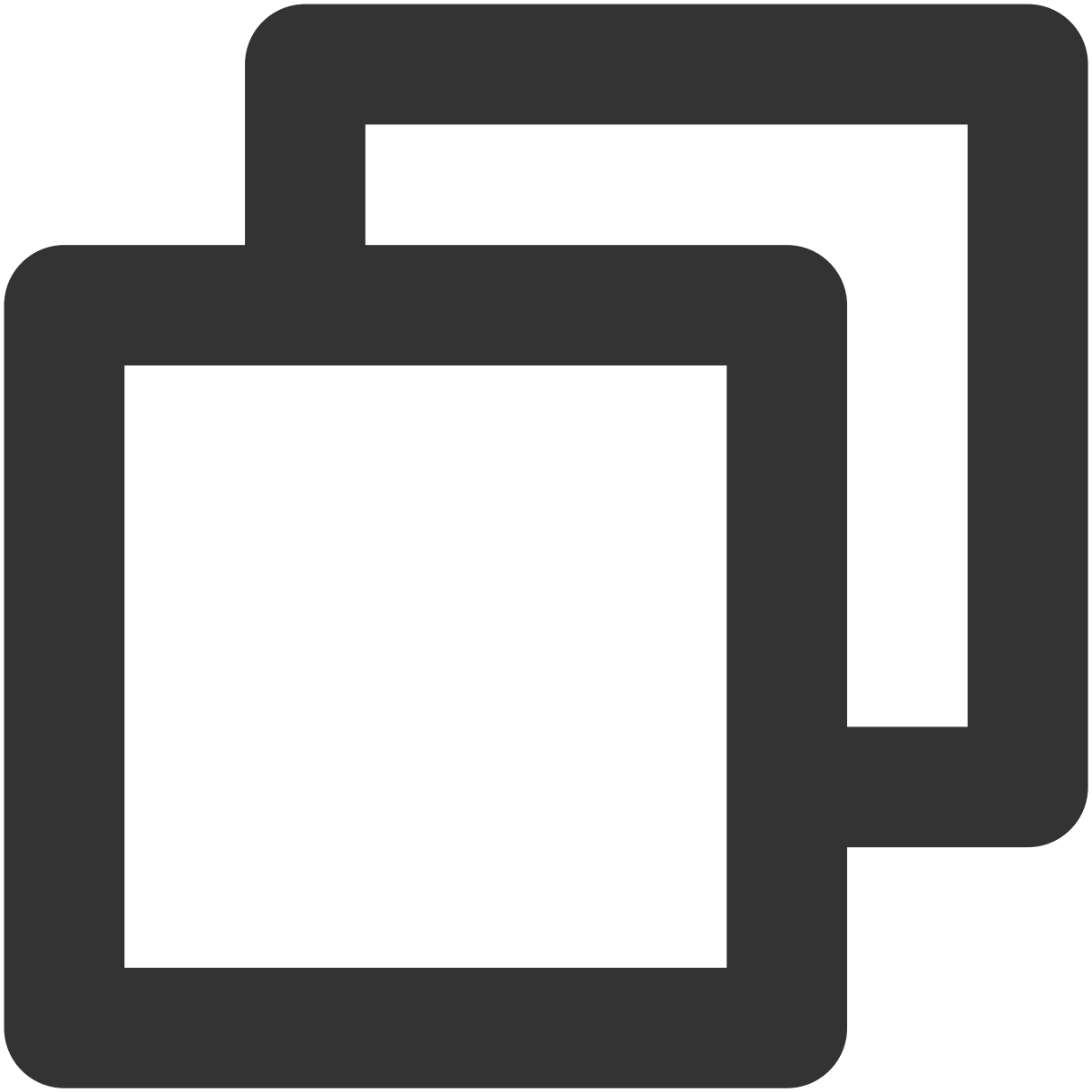
```
SELECT a FROM t IGNORE INDEX (csi);
```

Enforce statements to use column store indexes.



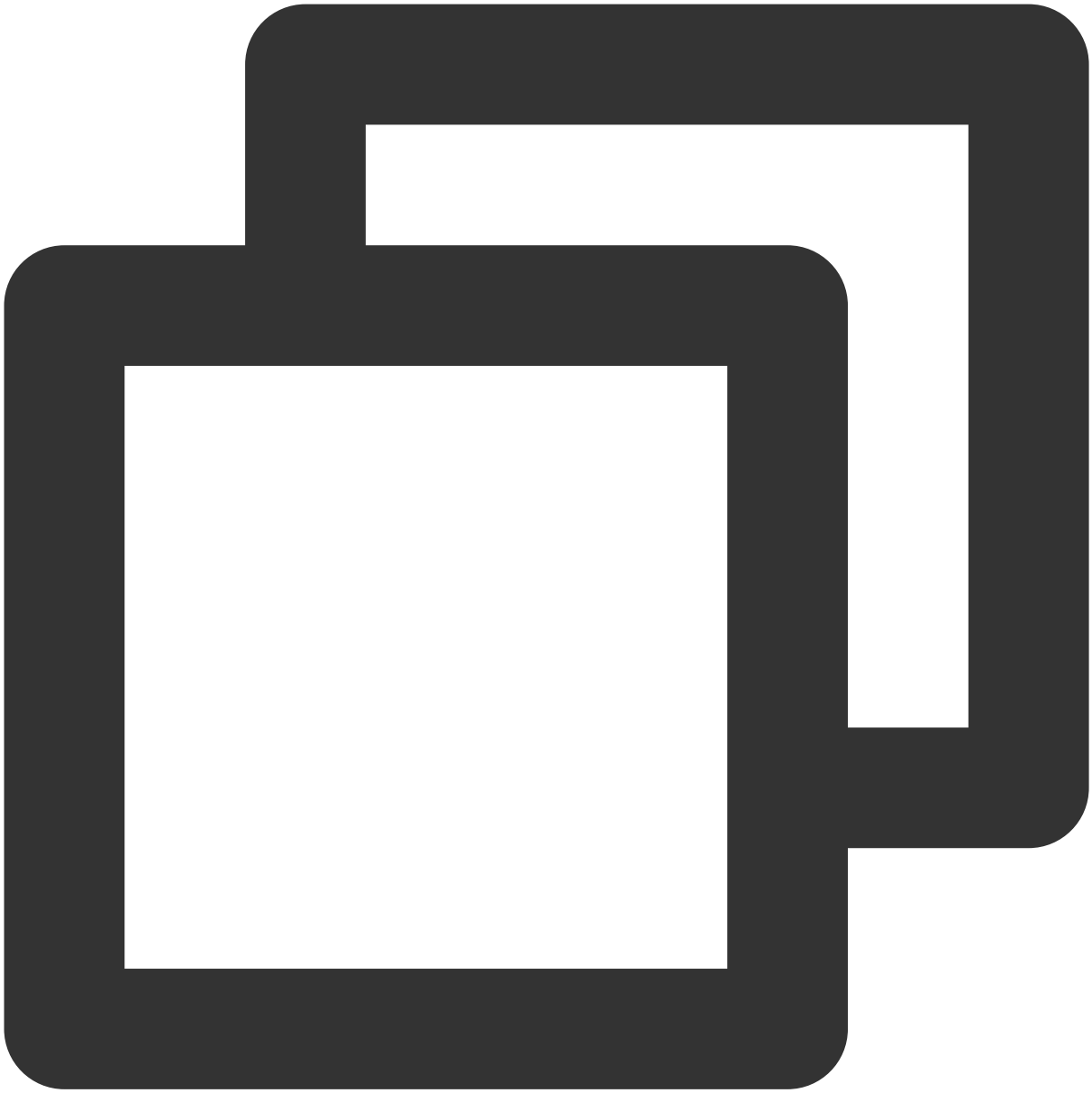
```
SELECT a FROM t FORCE INDEX (csi);
```

2. Use HINT statements for parallel CSI-based queries.



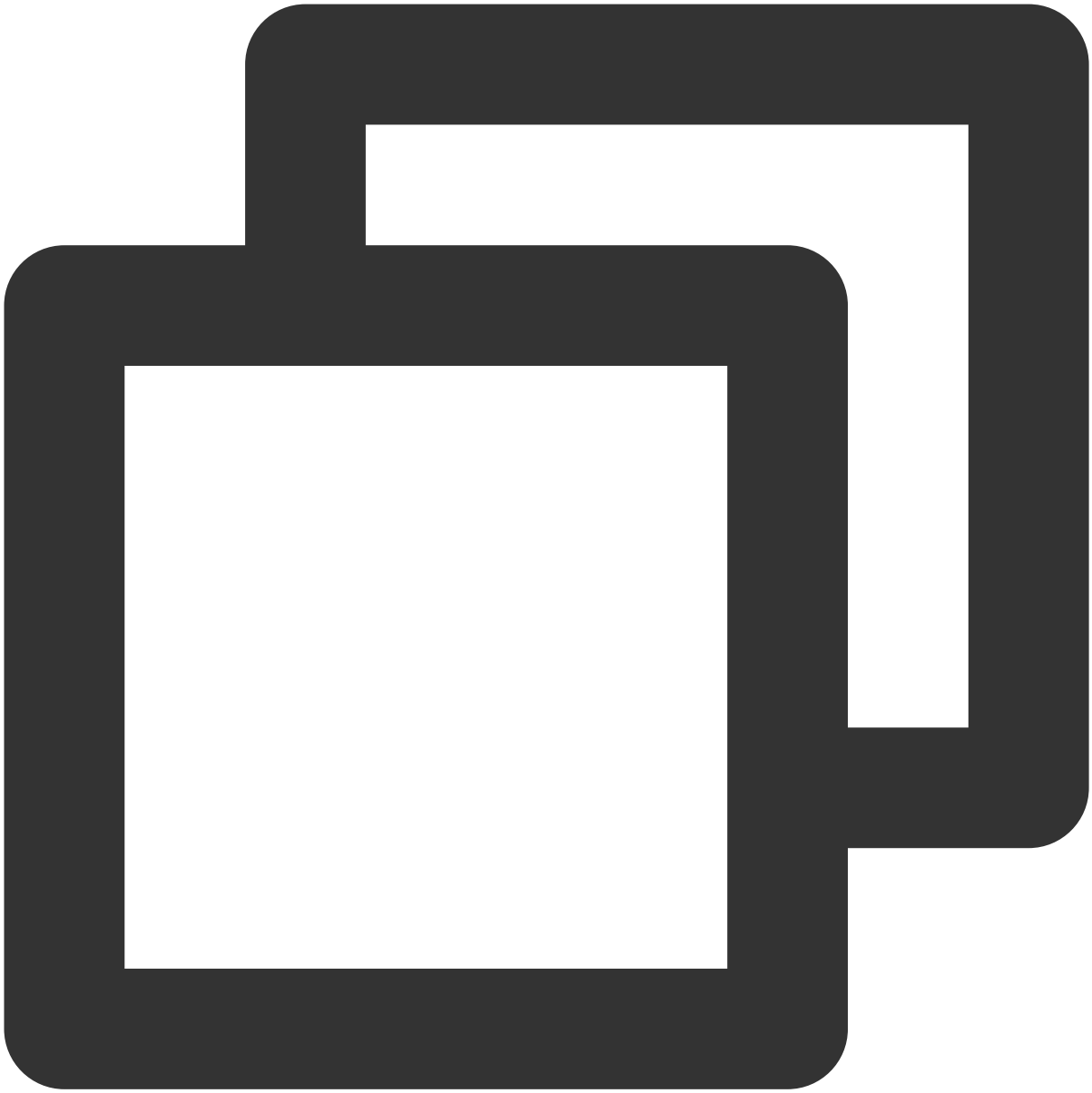
```
SELECT /*+PARALLEL(2)*/ a FROM t FORCE INDEX (csi);
```

Example of Creating a Table and Column Store Index



```
CREATE TABLE t (a int, columnstore index csi (a));
INSERT INTO t VALUES (0), (1), (2);
SHOW CREATE TABLE t;
SHOW INDEX FROM t;
```

Execution result:

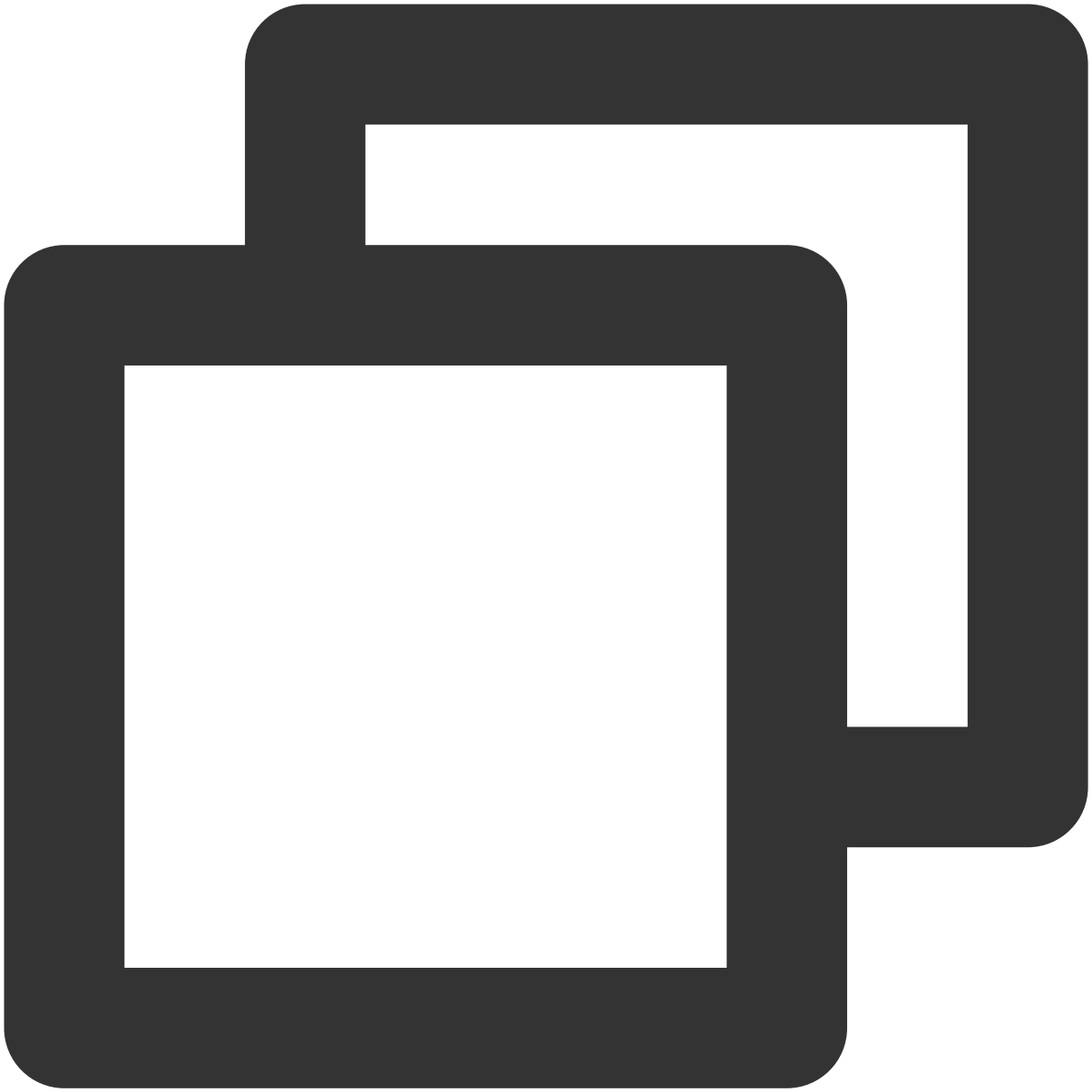


```
MySQL [test]> CREATE TABLE t (a int, columnstore index csi (a));
Query OK, 0 rows affected (0.01 sec)
MySQL [test]> INSERT INTO t VALUES (0), (1), (2);
Query OK, 3 rows affected (0.01 sec) Records: 3  Duplicates: 0  Warnings: 0
MySQL [test]> SHOW CREATE TABLE t;
+-------+-----------------------------------------------------------------------
| Table | Create Table
+-------+-----------------------------------------------------------------------
| t     | CREATE TABLE `t` (   `a` int DEFAULT NULL,   COLUMNSTORE KEY `csi` (`a`)
+-------+-----------------------------------------------------------------------
MySQL [test]> SHOW INDEX FROM t;
```

```
+-------+------------+----------+--------------+-------------+-----------+---------
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinal
+-------+------------+----------+--------------+-------------+-----------+---------
| t     |          1 | csi      |            1 | a           | NULL      |
+-------+------------+----------+--------------+-------------+-----------+---------
1 row in set (0.00 sec)
```

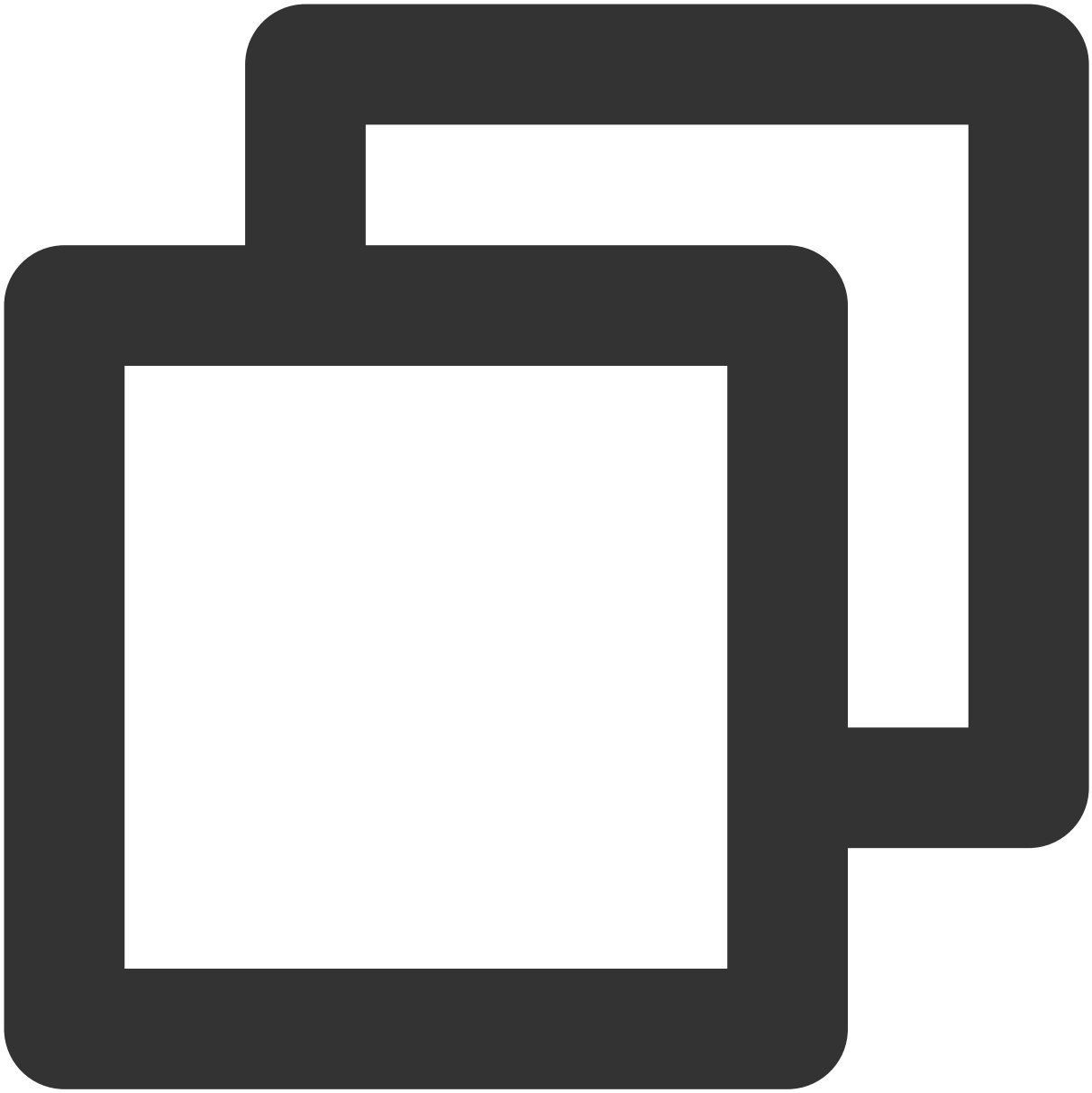
Usage of HINT

1. Enforce statements to use column store indexes.



```
SELECT a FROM t FORCE INDEX (csi);
EXPLAIN FORMAT=TREE SELECT a FROM t FORCE INDEX (csi);
```

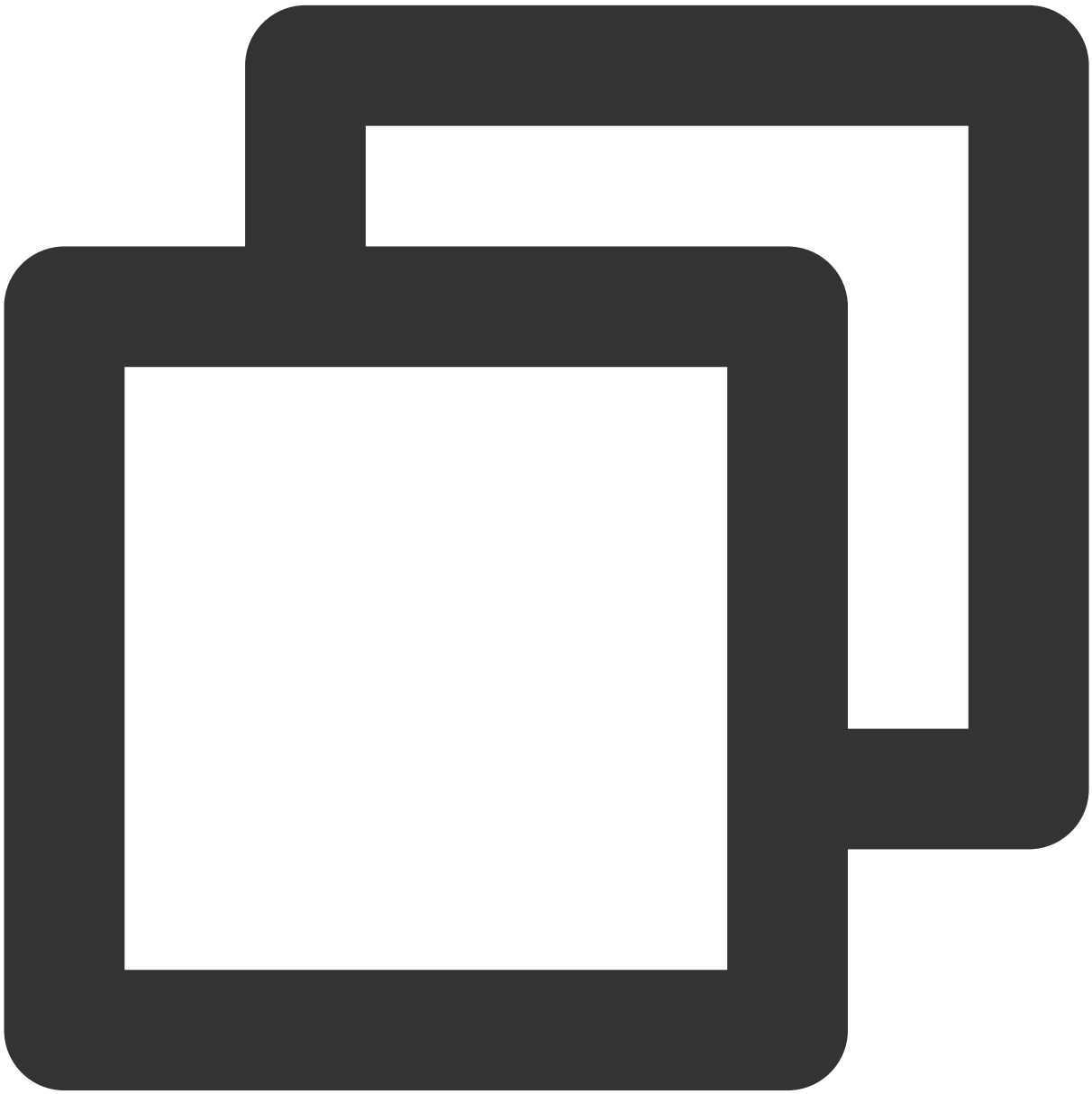
Execution result:



```
MySQL [test]> SELECT a FROM t FORCE INDEX (csi);
+------+
| a    |
+------+
|    0 |
|    1 |
|    2 |
+------+
3 rows in set (0.00 sec)
MySQL [test]> EXPLAIN FORMAT=TREE SELECT a FROM t FORCE INDEX (csi);
+-------------------------------------------------------------+
```

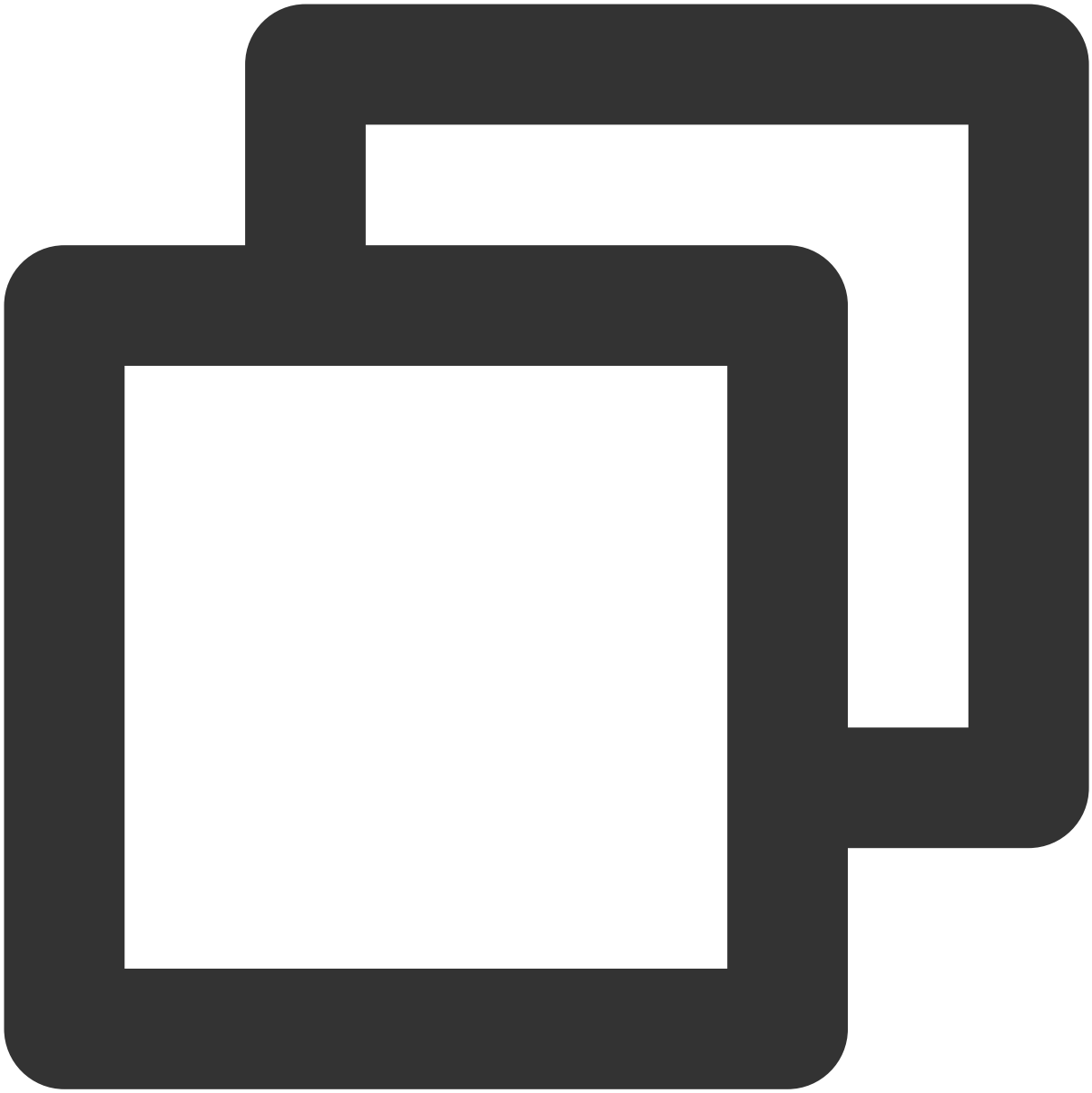
```
| EXPLAIN                                                                 |
+-------------------------------------------------------------------------+
| -> COLUMNSTORE Index scan on t using csi  (cost=1.30 rows=3)  |
+-------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

2. Enforce statements to avoid using column store indexes (using row store indexes).



```
SELECT a FROM t IGNORE INDEX (csi);
EXPLAIN FORMAT=TREE SELECT a FROM t IGNORE INDEX (csi);
```

Execution result:



```
MySQL [test]> SELECT a FROM t IGNORE INDEX (csi);
+------+
| a    |
+------+
|    0 |
|    1 |
|    2 |
+------+
3 rows in set (0.00 sec)
MySQL [test]> EXPLAIN FORMAT=TREE SELECT a FROM t IGNORE INDEX (csi);
+----------------------------------------+
```

```
| EXPLAIN                                    |
+--------------------------------------------+
| -> Table scan on t  (cost=0.55 rows=3)  |
+--------------------------------------------+
1 row in set (0.00 sec)
```

Viewing Creation Status of Column Store Indexes

show create table TABLE

Note:

By default, the COLUMNSTORE prefix is not displayed. It will be displayed only when columnstore_display_in_show_create is set to 1.

show index from TABLE

explain format=tree

Note:

Once CSI is enabled, explain format=tree can be used to check the status of column store index creation. The statement checks if the execution plan operator has the COLUMNSTORE prefix to determine whether the operator uses column store indexes for query execution. The COLUMNSTORE prefix is not displayed by default. It will be displayed only when format is set to tree.

Parameter Configuration and Monitoring Metrics

Last updated：2024-06-07 14:38:10

This document describes the parameter configuration and monitoring metrics of Column Store Index (CSI).

Prerequisites

The kernel version of TDSQL-C for MySQL 8.0 is 3.1.14 or later.

Note:

For read-only instances that meet the version requirements, the CSI feature can be enabled only on those with four or more CPU cores.

Parameters

Name	Type	Default Value	Value Range	Description
columnstore_buffer_pool_size	int	1024	0, INT_MAX	Total memory of C-Store (size of the CSI buffer pool), in MB.
columnstore_scan_pushdown_rows_threshold	int	100000	0, INT_MAX	If the number of data rows in an index table is less than this threshold, the data of the table will not be accessed through column storage. This parameter is not a global parameter and is effective only for instances with the CSI feature enabled.

CSI Monitoring Metrics

Metric Name	Unit	Aggregation Method	Description
Innodb_csi_select_pushdown		SUM	Number of executed statements for CSI-based queries.
Innodb_csi_select_fallback		SUM	Number of CSI rollback statements.
innodb_csi_lag_seconds	Second	MAX	Synchronization delay between read-write instances and instances with CSI enabled.
Innodb_csi_disk_usage	GB	MAX	Storage space used by column store indexes.
txsql_csi_fail		SUM	Number of column store indexes failed to be created.

Performance Test Report

Last updated：2023-12-12 14:40:17

This document is the Column Store Index (CSI) performance test report of TDSQL-C for MySQL.

Standard Performance Test Report

Data set: TPC-H.

Tested data volume: 100 GB.

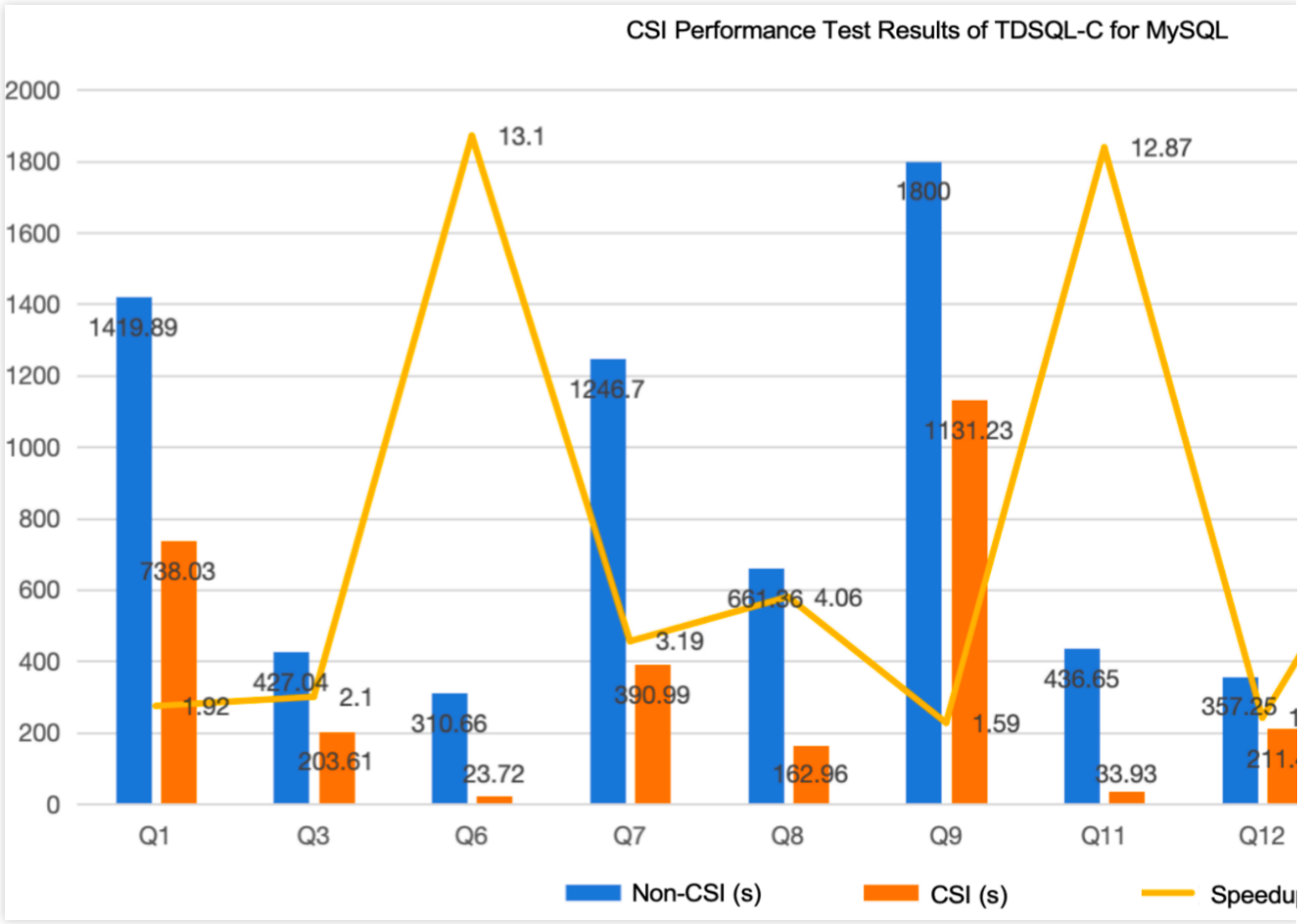
Instance specification: 32 cores and 256 GB of memory.

Query: execution time speedup of Q1, Q3, Q6, Q7, Q8, Q9, Q11, Q12, Q14, Q15, and Q19.

Execution Time

Query	Non-CSI (s)	CSI (s)	Speedup
Q1	1419.89	738.03	1.92
Q3	427.04	203.61	2.10
Q6	310.66	23.72	13.10
Q7	1246.7	390.99	3.19
Q8	661.36	162.96	4.06
Q9	1800	1131.23	1.59
Q11	436.65	33.93	12.87
Q12	357.25	211.42	1.69
Q14	329.48	48.19	6.84
Q15	709.06	99.90	7.10
Q19	561.56	269.04	2.09

Execution Time Speedup



Conclusion

CSI can improve the query performance by significantly reducing the time consumed by queries.

Parameter Configuration
Parameter Overview

Last updated：2023-03-01 14:33:46

This document describes system variables (also known as parameters), which are configurations items used to manage and control a database.

Use cases

TDSQL-C for MySQL provides a rich set of parameters for you to optimize the database performance. In the console, you can directly modify the default values of database parameters for formats, permissions, feature enablement, character set, execution conditions, cache size, and time/quantity/size limits to better suit your needs.

Parameter categories

TDSQL-C for MySQL parameters are divided into global parameters and session parameters. Once configured, values of global parameters will take effect for all instances in the cluster, while values of session parameters only apply to the target instance and can be synced to other instances.

Supported parameter operations

Select one of many preset templates with default parameter values when creating a TDSQL-C for MySQL cluster to improve the performance while guaranteeing the stability.
Adjust parameter values individually or in batches based on your business needs to flexibly adapt to different use cases.
Create custom templates by modifying a default parameter template.
Generate templates by importing parameters from configuration file `my.conf` .
Save parameter configurations as templates and apply them to other clusters.
Use formulas to set the values of certain parameters. When the instance specification is changed, parameter values set with formulas will also be automatically changed accordingly to keep the database in the optimal or most stable status.
Use global and session parameters.

References

For more information on how to modify parameters individually or in batches, modify parameters by importing a parameter template or configuration file, export the parameter configuration file, or query parameter modifications, see Setting Instance Parameters.

For more information on how to create, copy, import, export, or delete parameter templates and modify parameter values in parameter templates, see Applying Parameter Template.

For suggestions on how to set parameters, see Suggestions on Parameter Configuration.

Setting Instance Parameters

Last updated：2023-12-06 16:57:27

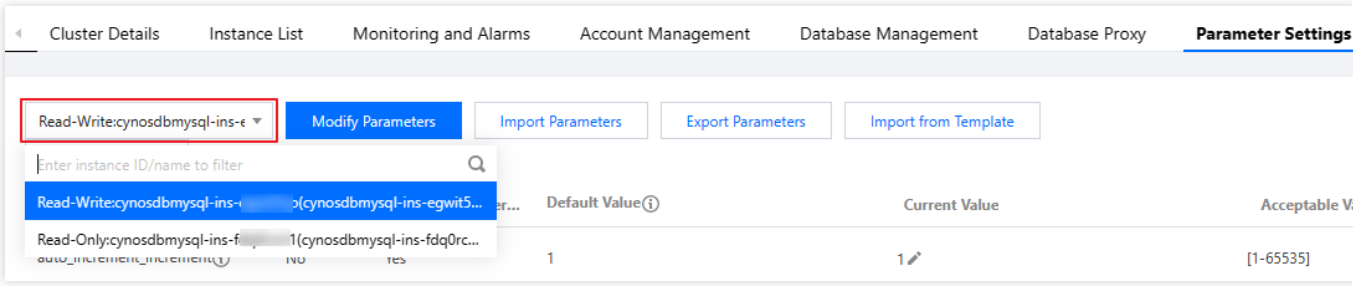
This document describes how to modify parameters and view parameter modifications in the console.

Notes

To ensure instance stability, only some parameters can be modified in the console. These parameters are displayed on the **Parameter Settings** tab.
If the modified parameter requires instance restart to take effect, the system will ask you if you want to restart. We recommend that you do so during off-peak hours and ensure that your application has a reconnection mechanism.
Parameter task: You can query the details of parameter modification tasks in the task list.
Parameter modification tasks that haven't been executed yet can be canceled.
If a cluster already has an ongoing parameter modification task, modifying parameters again will fail.
Global parameters can be modified only on a read-write instance. Once modified, they will be applied to all instances in the cluster. Session parameters can be set in individual instances in the cluster.

Modifying one parameter

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
2. On the cluster management page, click the **Parameter Settings** tab, select the target instance ID (global parameters can be modified only on a read-write instance and will take effect for the entire cluster once configured, while session parameters can be set in individual instances).



Note:

TDSQL-C for MySQL provides the **Global Parameters** field to help you quickly distinguish between parameters that take effect for the entire cluster and that can be set separately for different instances. Once configured, values of

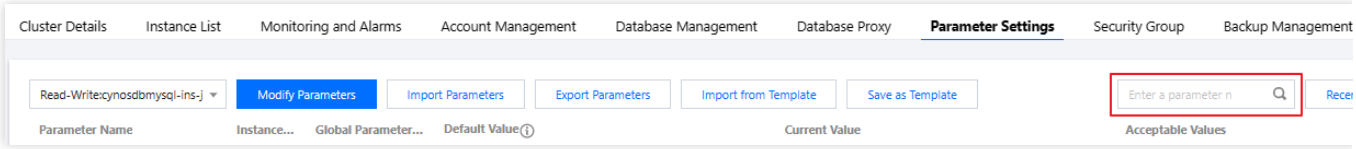
global parameters will take effect for all instances in the cluster, while values of session parameters will only apply to the target instance and can be synced to other instances.

3. Find the target parameter in the parameter list and click

in the **Current Value** column.

Note:

You can quickly find the target parameter in the search box on the right of the **Parameter Settings** tab.



4. Enter the target parameter value as prompted in the **Acceptable Values** column and click

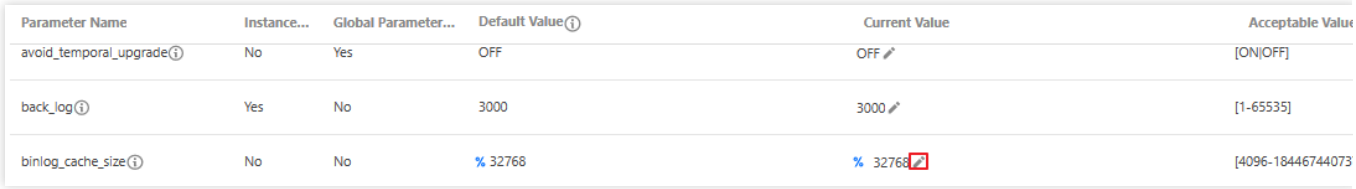
to save the change. You can also click

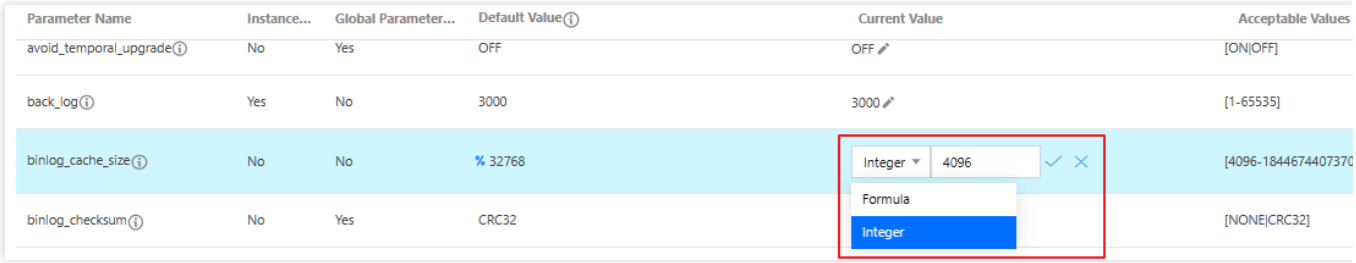
to discard the change.

Note:

If the

icon is displayed before a parameter value, the parameter supports two value types: integer and formula. After the parameter is set based on a formula, the value will change automatically along with specifications such as CPU and memory (you can only modify numeric variables in a formula; for more information, see Parameter Formula).





5. Click

. In the pop-up window, confirm that everything is correct, set **Execution Time**, and click **OK**.

Execution Time:

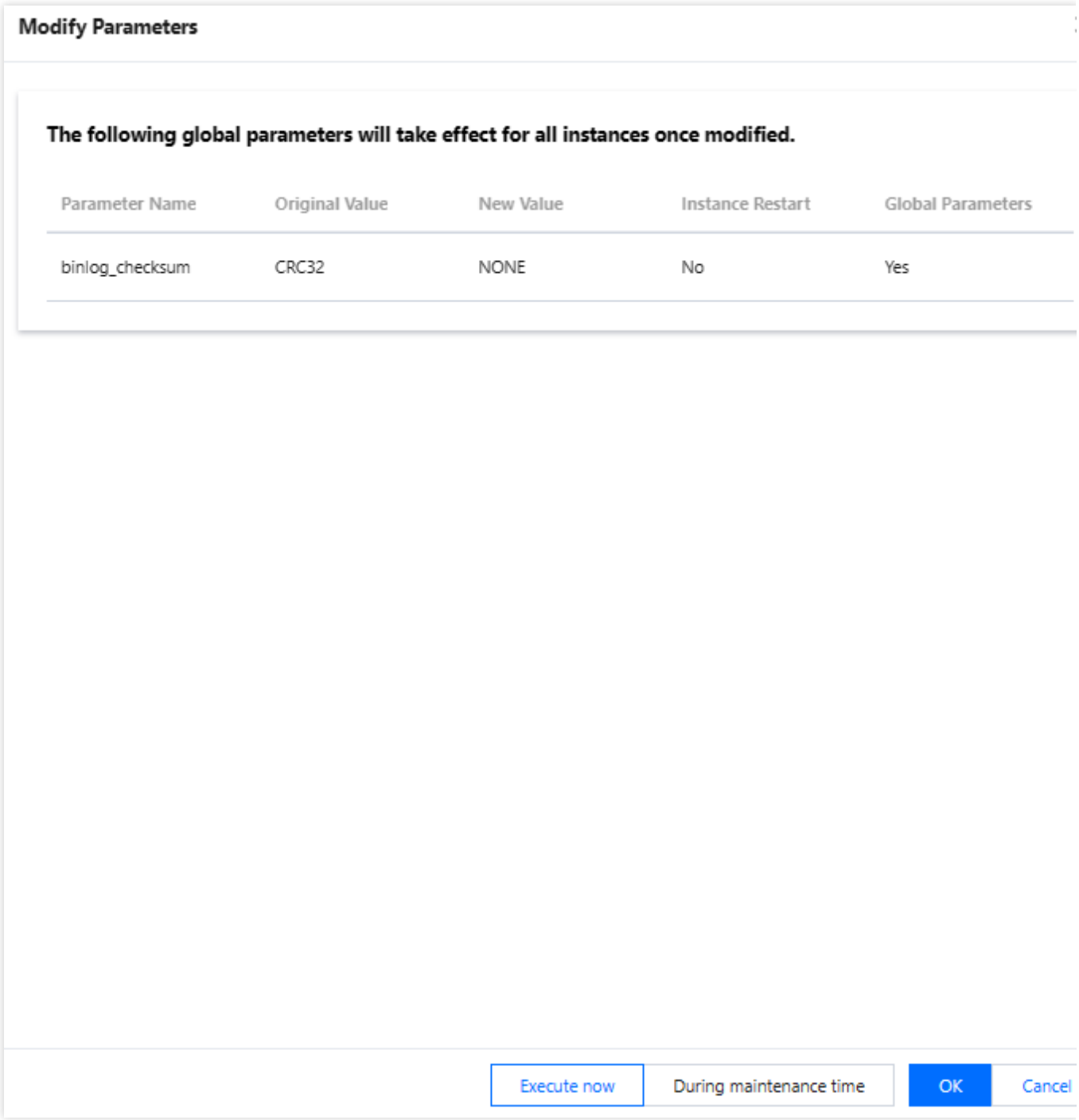
If **Execute now** is selected, modification will be triggered immediately upon confirmation.

If **During maintenance time** is selected, the parameter will be modified within the maintenance time of the instance. For more information, see Modifying Instance Maintenance Time.

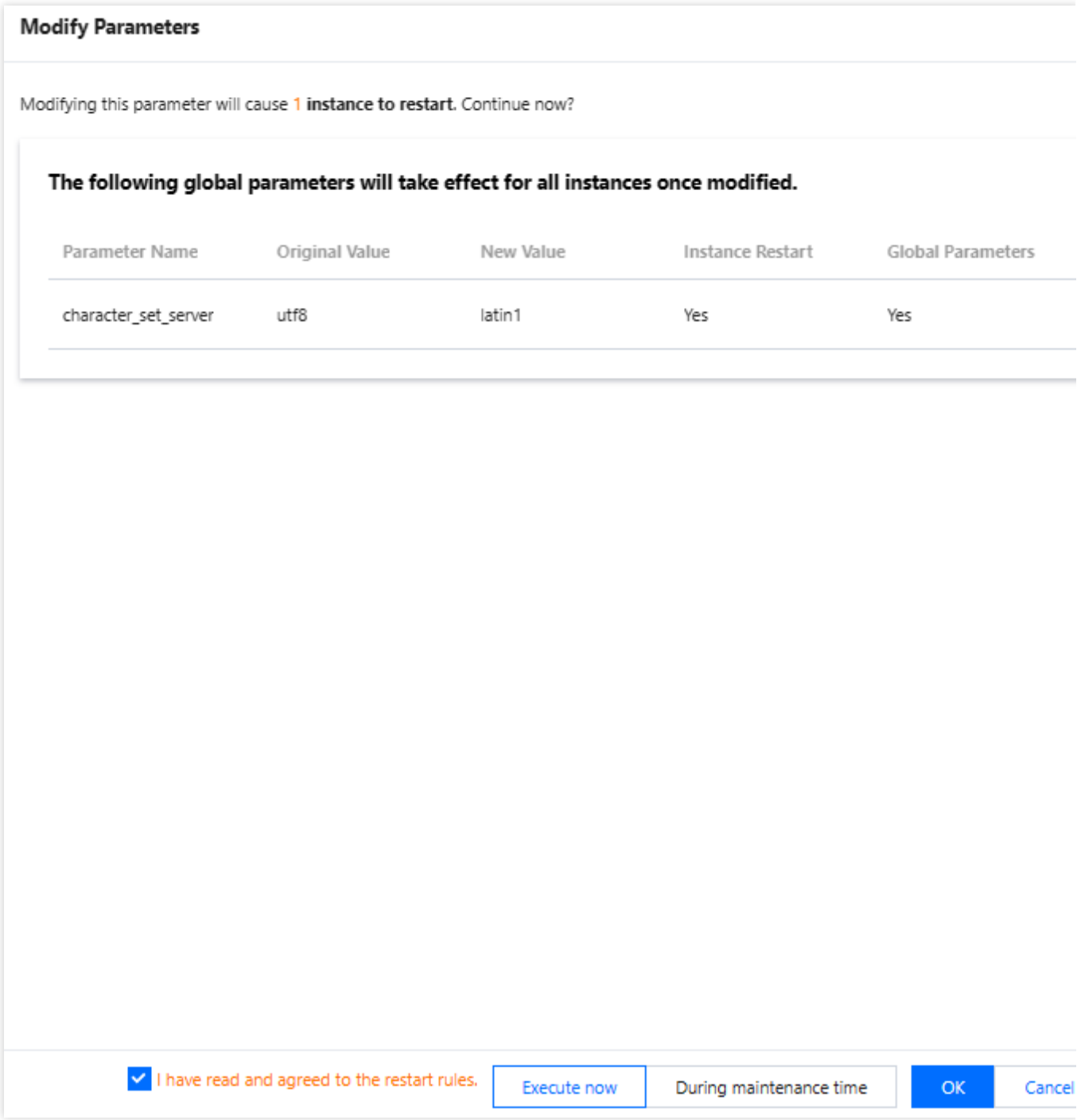
Note:

When you confirm the parameter modification, the system will ask you whether to restart the database instance. You can select **Execute now** or **During maintenance time**.

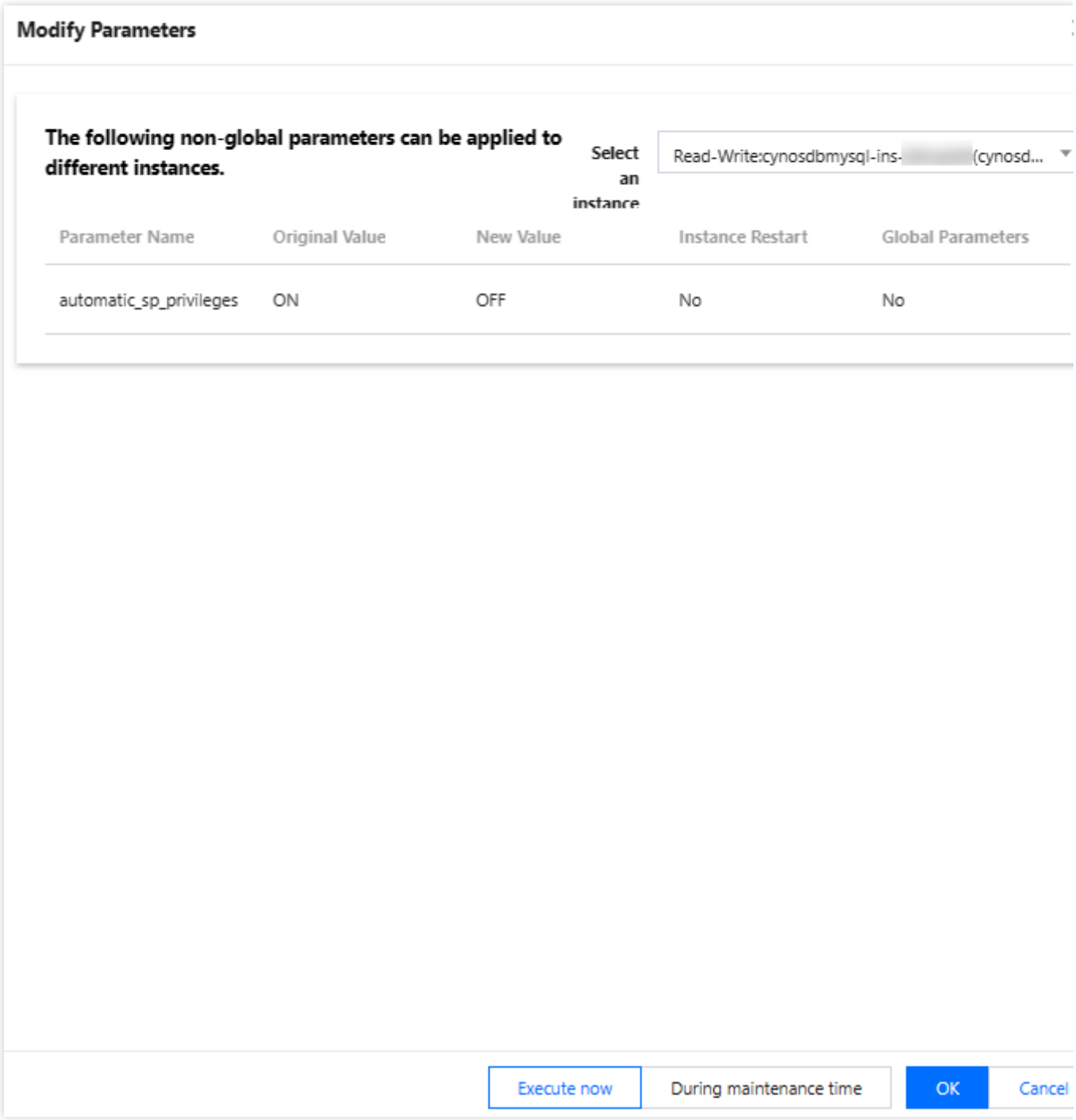
The page displayed when you modify a parameter that doesn't involve restart is as follows:



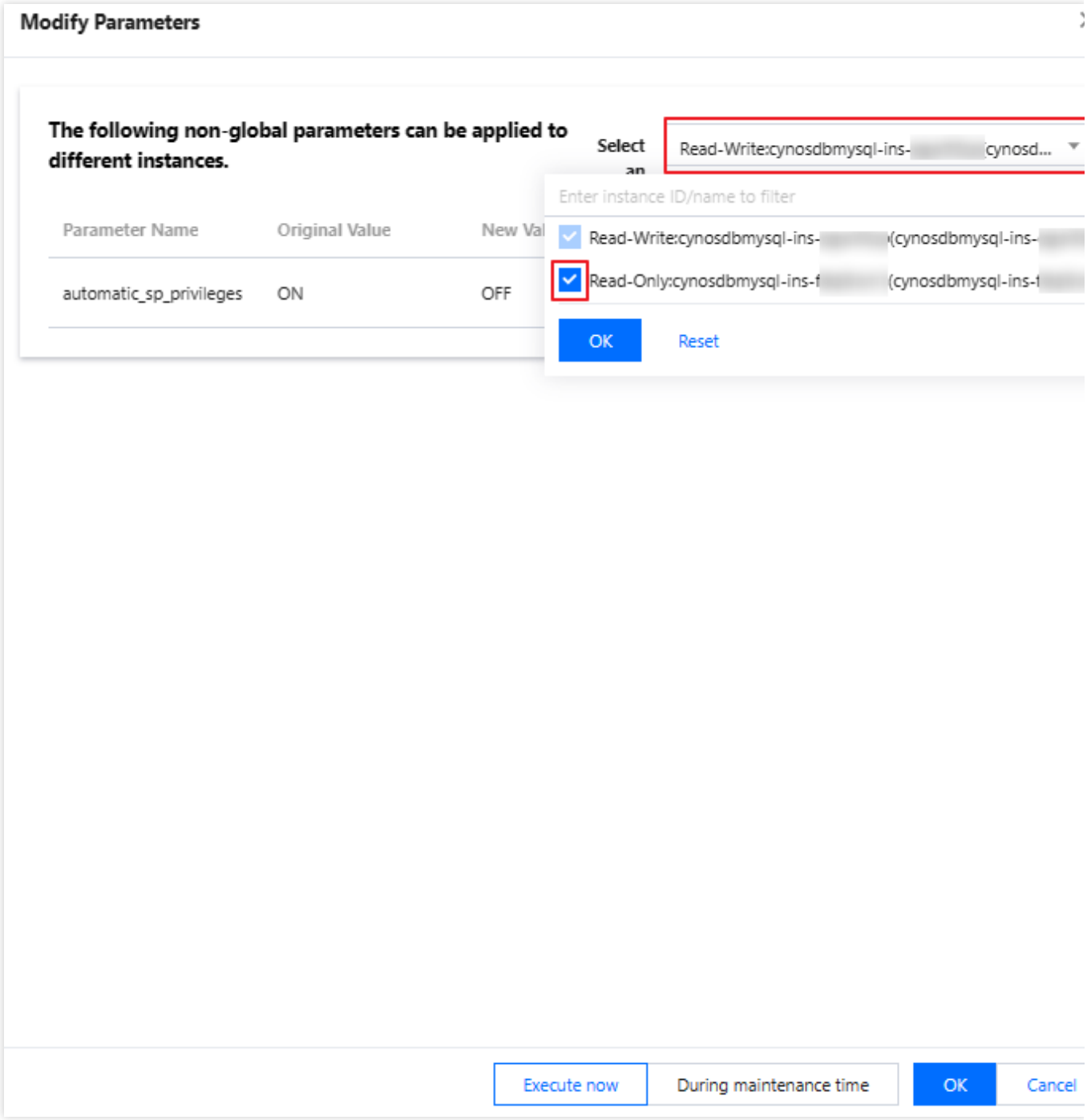
The page displayed when you modify a parameter that involves restart is as follows:



If the modified parameter is local, the system will ask you whether to sync its value to other instances.

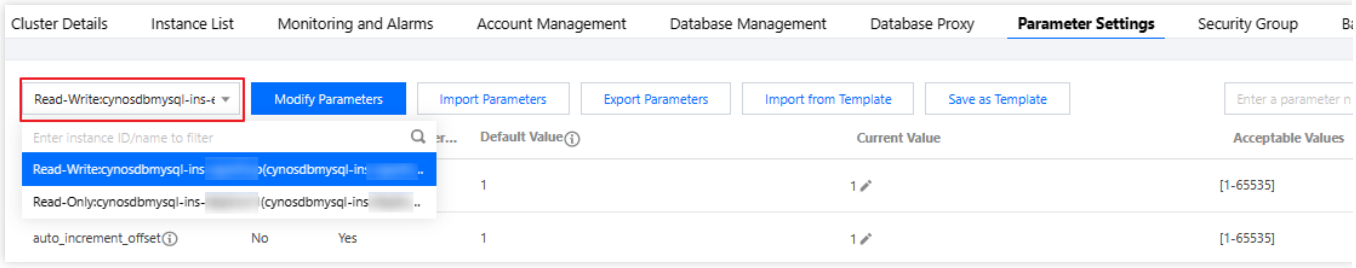


You can also select the target instance ID in the drop-down list to sync the value to it.



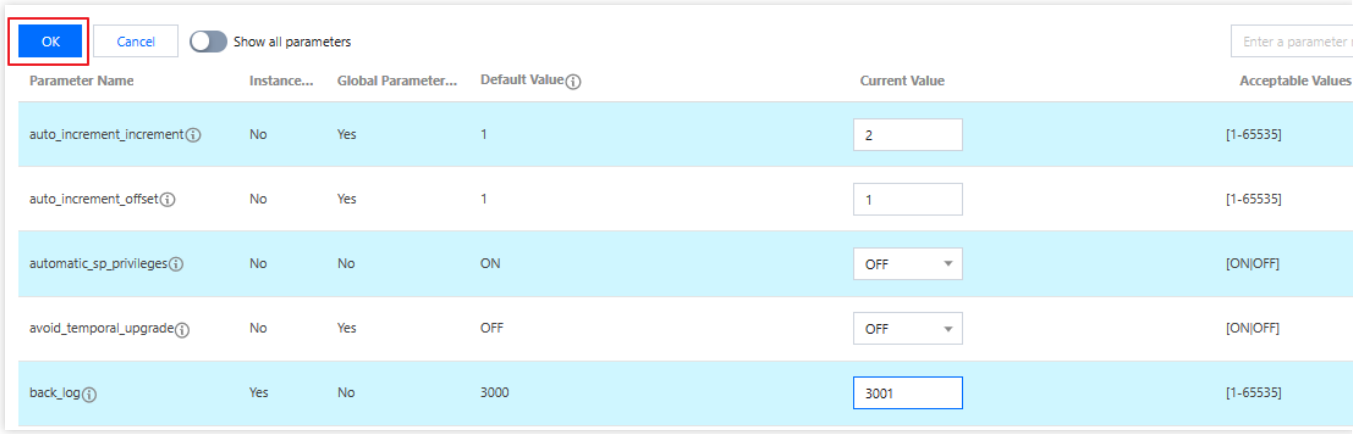
Batch modifying parameters

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
2. On the cluster management page, click the **Parameter Settings** tab, select the target instance ID (global parameters can be modified only on a read-write instance and will take effect for the entire cluster once configured, while session parameters can be set in individual instances).

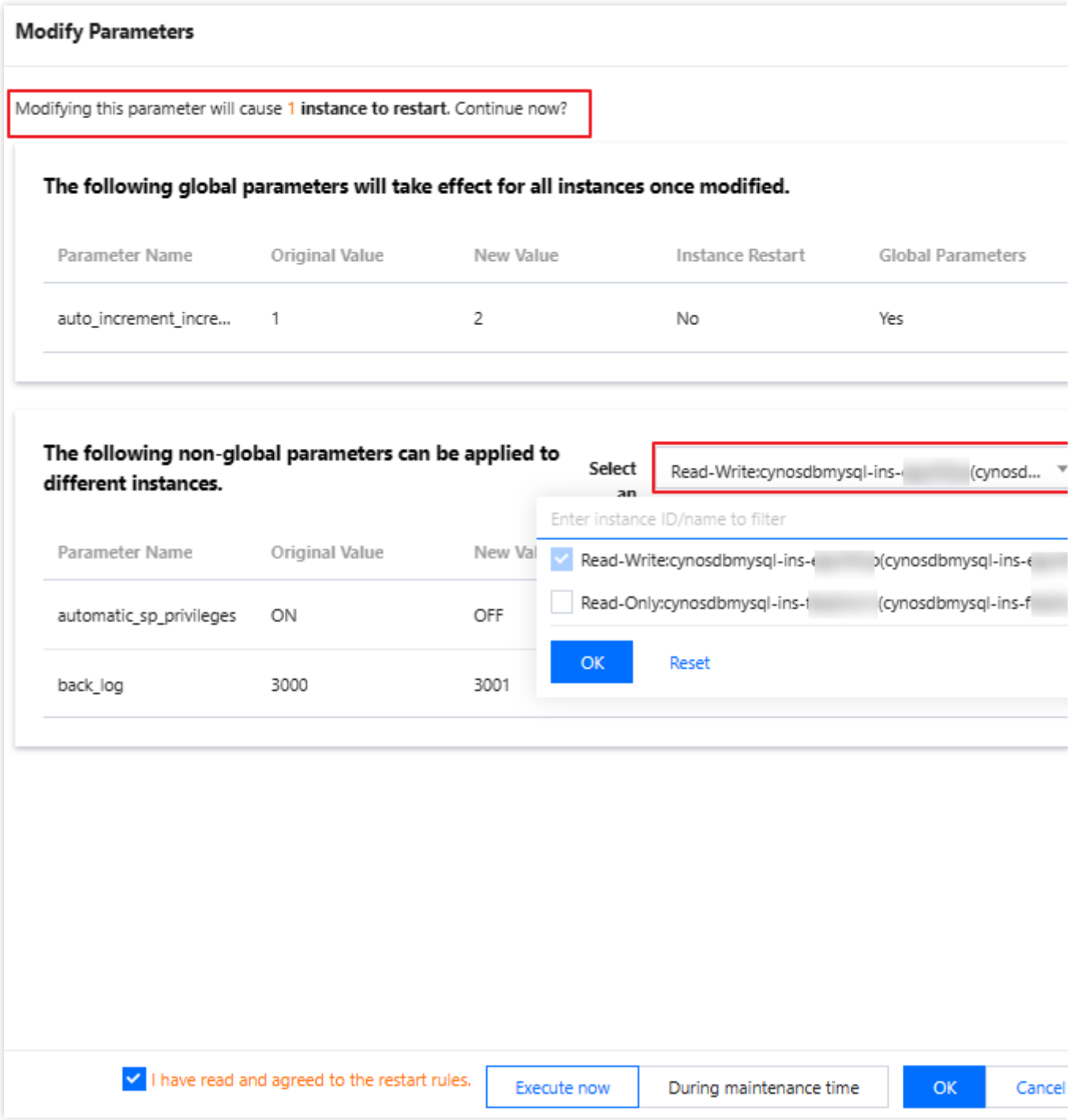


3. Click **Modify Parameters**.

4. Find the target parameters and modify their values in the **Current Value** column. After confirming that everything is correct, click **OK**.

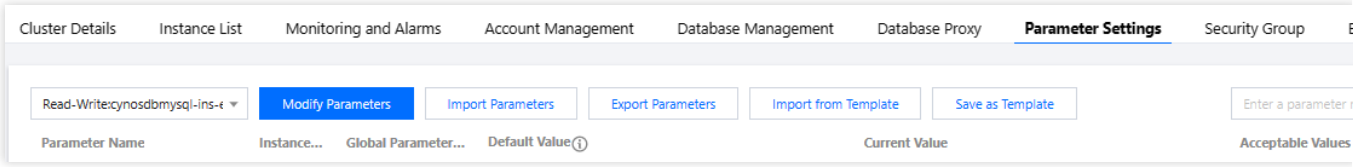


5. In the pop-up window, the system will ask you whether to restart the database instance. If there are session parameters involved, you can choose whether to sync them to other instances. After confirming that everything is correct, select **Execute now** or **During maintenance time** and click **OK**.



Modifying parameters by importing a parameter template

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
2. On the cluster management page, click the **Parameter Settings** tab, select the target instance ID, and click **Import from Template**.



3. In the pop-up window, select a parameter template and click **OK**.

4. After confirming that everything is correct, click **OK** in the top-left corner.

5. In the pop-up window, the system will ask you whether to restart the database instance. If there are session parameters involved, you can choose whether to sync them to other instances. After confirming that everything is correct, select **Execute now** or **During maintenance time** and click **OK**.

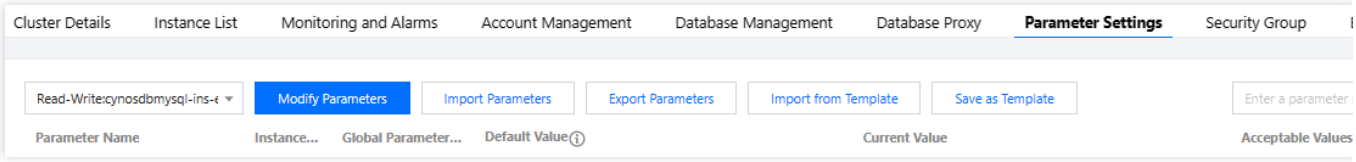
Modifying parameters by importing a parameter configuration file

Note:

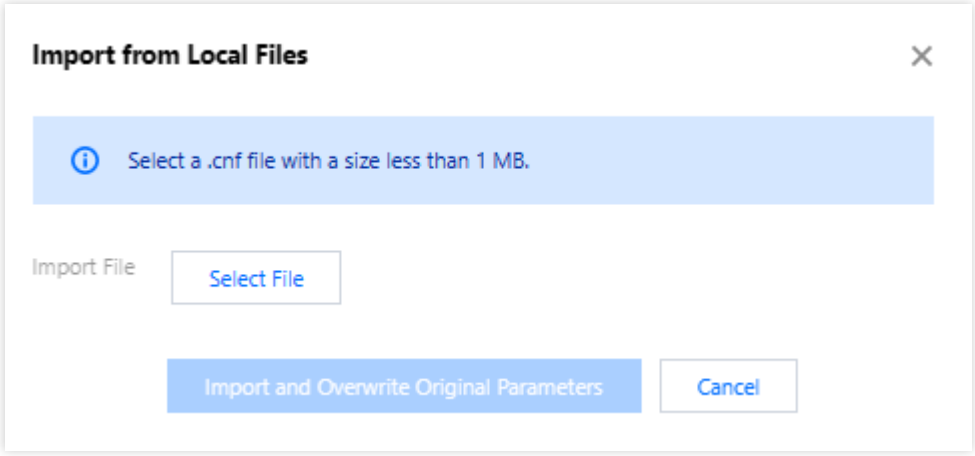
You cannot import formulas from a parameter configuration file.

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.

2. On the cluster management page, click the **Parameter Settings** tab, select the target instance ID, and click **Import Parameters**.



3. In the pop-up window, select the file to be uploaded and click **Import and Overwrite Original Parameters**.



4. After confirming that everything is correct, click **OK** in the top-left corner.

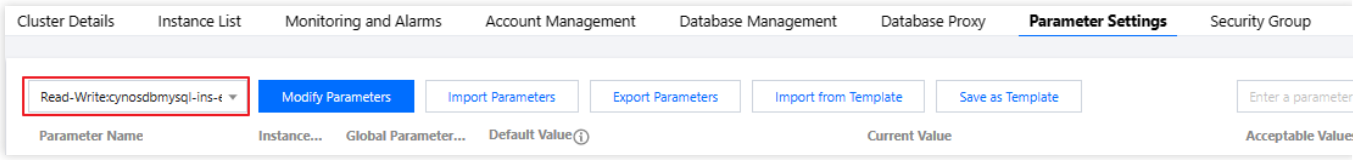
5. In the pop-up window, the system will ask you whether to restart the database instance. If there are session parameters involved, you can choose whether to sync them to other instances. After confirming that everything is correct, select **Execute now** or **During maintenance time** and click **OK**.

Exporting the parameter configuration as a file

Note:

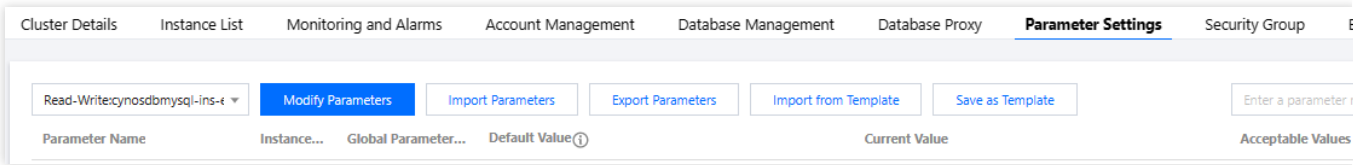
Currently, when you export a parameter configuration file, formula-based parameter values will be automatically converted into integer values for export.

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
2. On the cluster management page, click the **Parameter Settings** tab, select the target instance ID, click **Export Parameters**, and the exported file will be directly saved to the local file system.

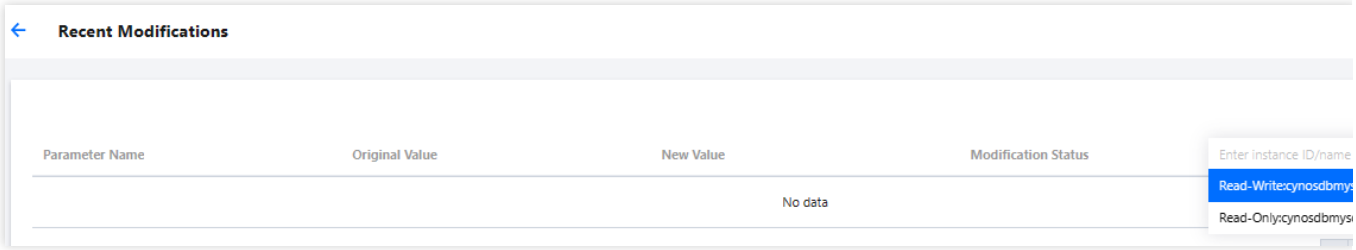


Querying parameter modifications

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
2. On the cluster management page, select the **Parameter Settings** tab and click **Recent Modifications** on the right.



3. On the page redirected to, you can select the target instance ID on the right to view the following fields of parameter modifications: **Parameter Name**, **Original Value**, **New Value**, **Modification Status**, and **Modification Time**.



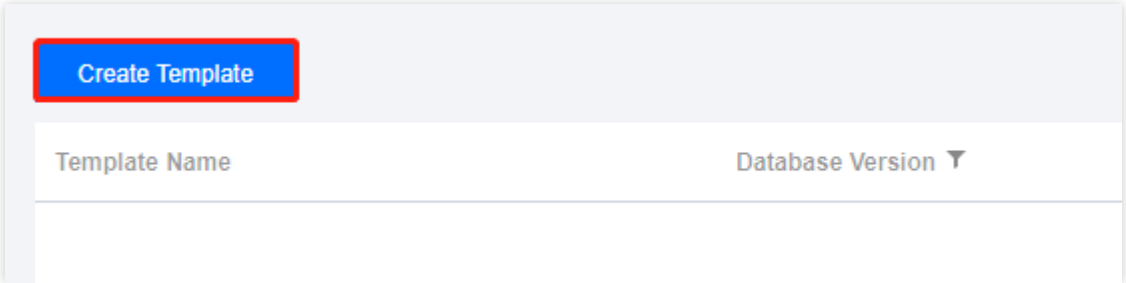
Applying Parameter Template

Last updated：2023-11-09 15:09:47

In addition to the various system parameter templates provided by TDSQL-C for MySQL, you can also create custom parameter templates to configure parameters in batches as needed.
You can apply a parameter template to configure and manage the parameters of a database engine. A template is like a container of the values of database engine parameters, which can be applied to one or more database instances.
You can log in to the console and click **Parameter Template** on the left sidebar to view parameters. The following parameter template features are supported:
Use a default parameter template. The default parameter values improve the TDSQL-C for MySQL performance while guaranteeing the stability.
Create custom templates by modifying a default parameter template.
Generate templates by importing parameters from configuration file `my.conf` .
Save parameter configurations as templates.
Set the values of certain parameters in a parameter template based on formula.
Note
If the parameters in the template are updated, the instance parameters are not updated unless they are manually re-applied to the instances.
You can apply the parameter changes to single or multiple instances by importing a template.

Creating a Parameter Template

To use your own database parameter template, you can create a parameter template, modify the parameter values, and apply the template to instances.
1. Log in to the TDSQL-C for MySQL console, select **Parameter Template** on the left sidebar, and click **Create Template**.



2. In the pop-up window, configure the following parameters and click **OK**.

Parameter	Description
Template	Enter a unique template name.

Name	
Template Description	Enter a brief description of the parameter template.
Template Type	Select the template type: **General** for a monthly subscribed or pay-as-you-go cluster or **Serverless** for a Serverless cluster.
Database Version	Select the required database version.

3. After the creation is completed, you can modify, import, and export parameters on the template details page.

Copying a Parameter Template

To include most of the custom parameters and values of an existing parameter template in a new template, you can copy the existing template.

Option 1. Copying an existing parameter template

1. Log in to the TDSQL-C for MySQL console, select **Parameter Template** on the left sidebar, find the target parameter template, and click **View/Modify** in the **Operation** column to enter the template details page.
2. Click **Save as Template** at the top of the template details page.
3. In the pop-up window, configure the following parameters:
Template Name: Enter a unique template name.
Template Description: Enter a brief description of the parameter template.
4. After confirming that everything is correct, click **OK**.

Option 2. Saving parameters of an instance as a parameter template

1. Log in to the TDSQL-C for MySQL console, find the target cluster in the cluster list, and click the cluster ID to enter the cluster management page.
2. On the cluster management page, select the **Parameter Settings** tab and click **Save as Template**.
3. In the pop-up window, configure the following parameters:
Template Name: Enter a unique template name.
Template Description: Enter a brief description of the parameter template.
4. After confirming that everything is correct, click **OK**.

Modifying Parameter Values in a Parameter Template

1. Log in to the TDSQL-C for MySQL console, select **Parameter Template** on the left sidebar, find the target parameter template, and click **View/Modify** in the **Operation** column to enter the template details page.
2. Click **Modify Parameters** or in the **Current Value** column to modify parameter values.

Note

If you click **Import Parameters**, you need to upload a local parameter configuration file. To avoid importing failures, the configuration file should be in the same format as the configuration file of the MySQL database server, or you can use the file template of the exported parameters.

Importing a Parameter Template

1. Log in to the TDSQL-C for MySQL console, select **Parameter Template** on the left sidebar, find the target parameter template, and click **View/Modify** in the **Operation** column to enter the template details page.
2. On the template details page, click **Import Parameters**.

Note

If you click **Import Parameters**, you need to upload a local parameter configuration file. To avoid importing failures, the configuration file should be in the same format as the configuration file of the MySQL database server, or you can use the file template of the exported parameters.

3. In the pop-up window, select a file and click **Import and Overwrite Original Parameters**.

Exporting a Parameter Template

Note

Currently, when you export a parameter template, formula-based parameter values will be automatically converted into integer values for export.

Option 1

1. Log in to the TDSQL-C for MySQL console and select **Parameter Template** on the left sidebar.
2. In the parameter template list, find the target template and click **Export** in the **Operation** column.

Option 2

1. Log in to the TDSQL-C for MySQL console, select **Parameter Template** on the left sidebar, and click **View/Modify** in the **Operation** column of a template to enter the template details page.
2. Click **Export Parameters** at the top of the template details page.

Deleting a Parameter Template

If a parameter template is created redundantly or no longer needed, it can be easily deleted.

1. Log in to the TDSQL-C for MySQL console and select **Parameter Template** on the left sidebar.
2. In the parameter template list, find the target template and click **Delete** in the **Operation** column.
3. In the pop-up window, click **OK**.

Suggestions on Parameter Configuration

Last updated：2023-03-01 14:33:46

Parameters in TDSQL-C for MySQL have been optimized on the basis of official default values in MySQL. We recommend you configure the following parameters for your TDSQL-C for MySQL instance after purchase based on your business scenarios.

character_set_server

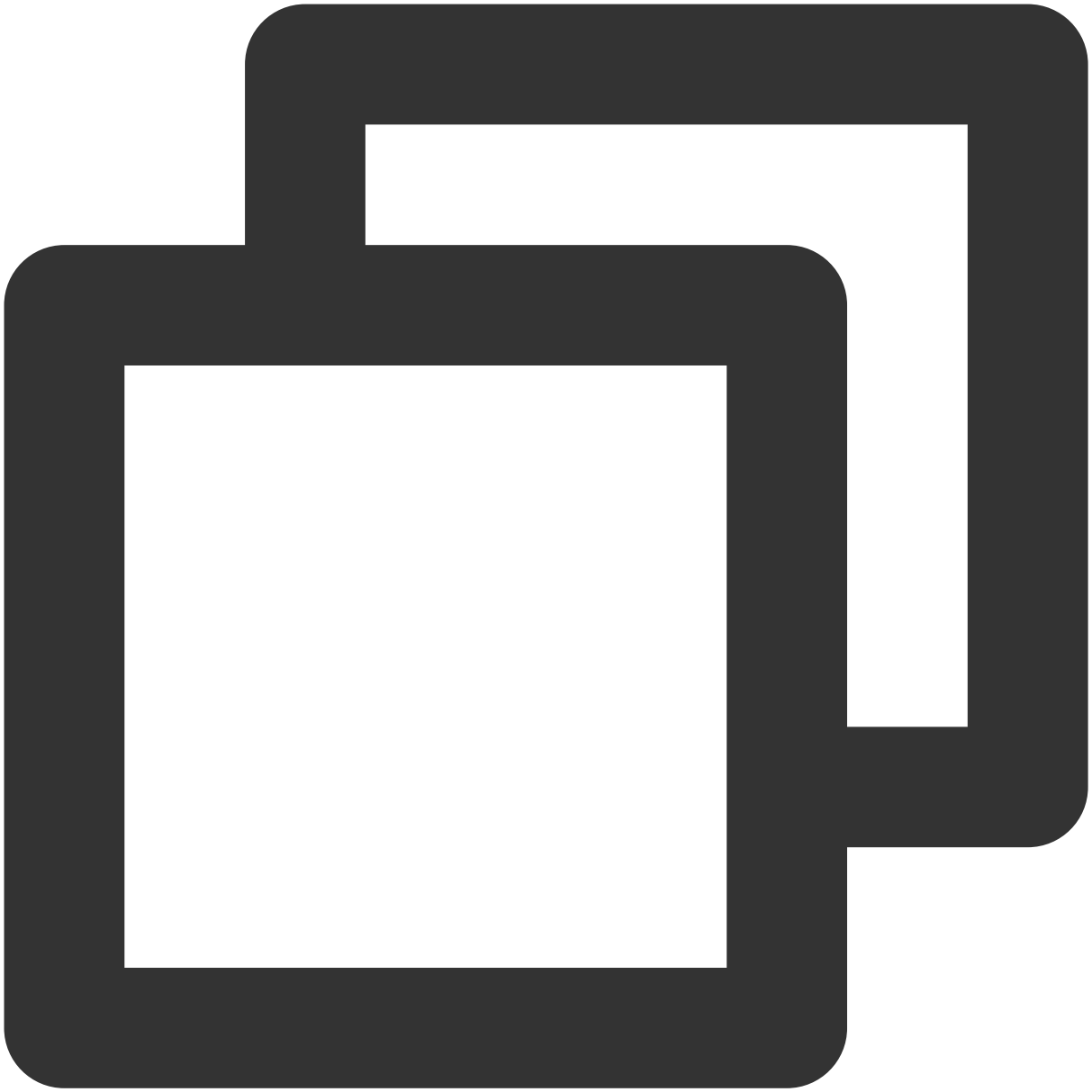
Default value: UTF8
Restart required: Yes
Role: It is used to configure the default character set of the TDSQL-C for MySQL server. TDSQL-C for MySQL provides four character sets: LATIN1, UTF8, GBK, and UTF8MB4.
LATIN1 supports English letters. Each character in it occupies 1 byte.
UTF8 is a highly universal international encoding that contains the characters used by all countries/regions in the world. Each character in it occupies 3 bytes.
GBK uses 2 bytes to encode any character, that is, no matter whether it is a Chinese or English character, it is represented by 2 bytes.
As a superset of UTF8, UTF8MB4 is completely compatible with UTF8. Each character in it occupies 4 bytes, and emojis are supported.
Recommendation: After purchasing an instance, you need to select the appropriate character set based on the data format required in your business to ensure that the client and the server use the same character set, preventing garbled text and unnecessary restarts.

lower_case_table_names

Default value: 0
Restart required: Yes
Description: When creating a database or table, you can set whether storage and query operations are case-sensitive. This parameter can be set to 0 (case-sensitive) or 1 (case-insensitive). The default value is 0.
Recommendation: TencentDB for MySQL is case-sensitive by default. We recommend that you configure this parameter based on your business needs and usage habits.

sql_mode

Default values:



```
NO_ENGINE_SUBSTITUTION (v5.6); ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, NO_ZERO_IN_
```

Restart required: No

Description: TencentDB for MySQL can operate in different SQL modes, which define the SQL syntax and data check that it should support.

The default value of this parameter on v5.6 is `NO_ENGINE_SUBSTITUTION` , indicating that if the used storage engine is disabled or not compiled, an error will be reported.

On v5.7 and v8.0, the default values are `ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES` , `NO_ZERO_IN_DATE, NO_ZERO_DATE` , `ERROR_FOR_DIVISION_BY_ZERO` , `NO_AUTO_CREATE_USER` ,

and `NO_ENGINE_SUBSTITUTION` .

Here:

If `ONLY_FULL_GROUP_BY` is enabled, MySQL rejects queries for which the select list, `HAVING` condition, or `ORDER BY` list refer to nonaggregated columns that are neither named in the `GROUP BY` clause nor are functionally dependent on `GROUP BY` columns.

`STRICT_TRANS_TABLES` enables strict SQL mode. `NO_ZERO_IN_DATE` controls whether the server permits dates in which the year part is nonzero but the month or day part is zero. The effect of `NO_ZERO_IN_DATE` depends on whether strict SQL mode is enabled.

`NO_ZERO_DATE` controls whether the server permits a zero date as valid. Its effect depends on whether strict SQL mode is enabled.

`ERROR_FOR_DIVISION_BY_ZERO` means that in strict SQL mode, if data is divided by zero during the INSERT or UPDATE process, an error rather than a warning will be generated, while in non-strict SQL mode, NULL will be returned.

`NO_AUTO_CREATE_USER` prohibits the GRANT statement from creating a user with an empty password.

`NO_ENGINE_SUBSTITUTION` means that if the storage engine is disabled or not compiled, an error will be reported.

Recommendation: As different SQL modes support different SQL syntax, we recommend that you configure them based on your business needs and development habits.

long_query_time

Default value: 10

Restart required: No

Description: Used to define the time threshold for slow queries, with the default value as 10s. If a query execution takes 10s or longer, the execution details will be recorded in the slow log for future analysis.

Recommendation: As business scenarios and performance sensitivity may vary, we recommend that you set the value in consideration of future performance analysis.

Parameter Formula

Last updated：2023-12-13 10:18:41

TDSQL-C for MySQL now allows certain parameters to use formulaic values for smarter database adaptation. When the instance specifications are adjusted, such parameter values will also be automatically adjusted accordingly, ensuring optimal status or stability of the database.

Notice

The parameter formula is only applicable to parameters with numerical values. Formulaic values are not supported for parameters of other data types.
Upon the setting of formulaic parameter values, these values will change in accordance with the instance specification adjustment. If the calculated parameter value resulting from your formula exceeds the parameter value range (minimum value to maximum value), the parameter value will be set to the nearest boundary value within this range. That is, if it is smaller than the minimum value, the parameter will adopt the minimum value. If it exceeds the maximum value, the parameter will adopt the maximum value.
Example:
A parameter, as calculated by the configured formula, results in a value of 7. However, the specified range for the parameter is 1 to 6. In this case, the final value assigned to the parameter will be 6.
A parameter, as calculated by the configured formula, results in a value of 5. However, the specified range for the parameter is 6 to 10. In this case, the final value assigned to the parameter will be 6.
When export from or import from configuration files is configured, formula-based parameter values are not supported. During export, they will automatically be adjusted to integer values.
To ensure the availability of the database, only a select set of parameters currently accommodates formulaic configurations. For more parameters, see upcoming iterations.

Parameter Formula Description

Parameter Formula Composition	Name	Description
Variable	DBinitMemory	The memory size of the instance specification represented as an integer, in the unit of MB.
	DBInitCpu	The number of CPU cores in the instance specification, represented as an integer, in the unit of cores.

Operator	Division operator (/)	Divides the dividend by the divisor to return an integer quotient. If the result is a decimal, the integral part is used.
	Multiplication operator (*)	The product of two factors is returned as an integer. If the result is a decimal, the integral part is used.
Function	MIN()	Returns the smallest value in the form of an integer or parameter formula list.
	MAX()	Returns the greatest value in the form of an integer or parameter formula list.

Example:
The MAX(DBInitCpu/2,4) function indicates that the number of CPU cores of the instance divided by 2 is compared with 4 and the larger value is returned.

Parameters Supporting Parameter Formula

The following parameters support parameter formula in the current version. You can modify default formulas according to your business requirements.

Parameter	Description	Default Formula
binlog_cache_size	The size of the memory buffer used to store binary logs of adjustments during the transaction.	MIN(DBInitMemory/4000 * 32768,2097152)
max_heap_table_size	This variable sets the maximum size of user-created tables in the memory.	MIN(DBInitMemory/1000 * 4194304,134217728)
innodb_buffer_pool_size	The size (in bytes) of the buffer pool, which is the memory area of InnoDB	min((DBInitMemory - 500), DBInitMemory*3/4)*1000000

	cache table and index data.	
innodb_buffer_pool_instances	The number of regions divided according to the InnoDB cache pools.	MIN(DBInitMemory/2000,16)
innodb_read_io_threads	The number of I/O threads in InnoDB used for read operations.	MAX(DBInitCpu/2,4)
innodb_write_io_threads	The number of I/O threads in InnoDB used for write operations.	MAX(DBInitCpu/2,4)
join_buffer_size	The minimum size of the cache for table joins used in common index scans, range index scans, and full table scans.	MIN(DBInitMemory*128,262144)
max_connections	The maximum number of connections.	MIN(DBInitMemory/4+500,100000)
table_definition_cache	The number of opened table cache instances.	MAX(DBInitMemory*512/1000,2048)
table_open_cache	The size of table descriptor cache, which can reduce the number of times the files are opened and closed.	MIN(MAX(DBInitMemory*512/1000,2048), 65536)
table_open_cache_instances	The number of partitions for the MySQL table handle cache.	MIN(DBInitMemory/1000,16)
thread_pool_size	This parameter indicates the number of thread groups in the thread pool. By default, the number of thread groups equals the number of CPUs.	MIN(DBInitCpu,64)
thread_cache_size	The number of threads that should be retained in the cache for reuse.	MIN(DBInitMemory/125+8,512)

tmp_table_size	The maximum size of the internal memory temporary table.	MIN(DBInitMemory/1000*4194304,134217728)

Multi-AZ Deployment Overview

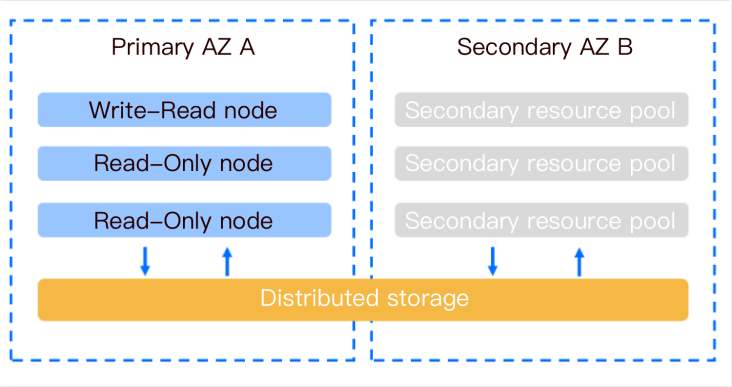
Last updated：2024-06-18 09:29:30

TDSQL-C for MySQL clusters come in either pre-configured resources or Serverless forms, both supporting multi-AZ deployments. Compared to single-AZ deployment, multi-AZ deployment offers higher disaster recovery capabilities to protect the database against instance failures or AZ interruptions, capable of withstanding data center-level failures. Multi-AZ deployment provides database instances with High Availability and failover support. A multi-AZ structure combines several single-AZs within the same region into a physical zone at a level above single-AZ. This article introduces multi-AZ deployment for clusters with pre-configured resources. For details on multi-AZ deployment for Serverless instance types, see Multi-AZ deployment.

Prerequisites

The cluster region has at least two AZs.
The target AZ has sufficient computing resources.
Database version requirements:
Database version 5.7 with kernel minor version 2.0.15 or above.
Database version 8.0 with kernel minor version 3.0.1 or above.

Multi-AZ Deployment Architecture



Supported Regions and AZs

Currently, this feature is in beta test and only supports the following regions and AZs.

This feature will gradually support more regions and AZs.

If required by your business, you can submit a ticket to apply for deployment in other regions and AZs.

Supported Regions	Supports primary availability zone	Supports secondary availability zone
Beijing	Beijing Zone 3	Beijing Zone 5
	Beijing Zone 5	Beijing Zone 7
	Beijing Zone 6	Beijing Zone 7
	Beijing Zone 7	Beijing Zone 5
Guangzhou	Guangzhou Zone 3	Guangzhou Zone 4
	Guangzhou Zone 4	Guangzhou Zone 6
	Guangzhou Zone 6	Guangzhou Zone 4
Shanghai	Shanghai Zone 2	Shanghai Zone 4
	Shanghai Zone 4	Shanghai Zone 2
	Shanghai Zone 5	Shanghai Zone 4
Hong Kong (China)	Hong Kong (China) Zone 2	Hong Kong Zone 3
Singapore	Singapore Zone 2	Singapore Zone 4
	Singapore Zone 3	Singapore Zone 4
Silicon Valley	Silicon Valley Zone 2	Silicon Valley Zone 1
Frankfurt	Frankfurt Zone 1	Frankfurt Zone 2
	Frankfurt Zone 2	Frankfurt Zone 1
Virginia	Virginia Zone 1	Virginia Zone 2
	Virginia Zone 2	Virginia Zone 1
Tokyo	Tokyo Zone 2	Tokyo Zone 1

How to Implement Multi-AZ Architecture

Regarding the multi-AZ architecture, it can be achieved by creating a new cluster via the console. Existing single-AZ clusters will also be upgraded to multi-AZ clusters. This upgrade is automatically completed through online data

migration, with no impact on your business. For more details, see Setting Multi-AZ Deployment.

Multi-AZ Billing Description

There are no additional fees for the multi-AZ feature for the time being.

Note:

Currently, single-AZ clusters can also be upgraded to multi-AZ clusters for free.

Multi-AZ Information Display

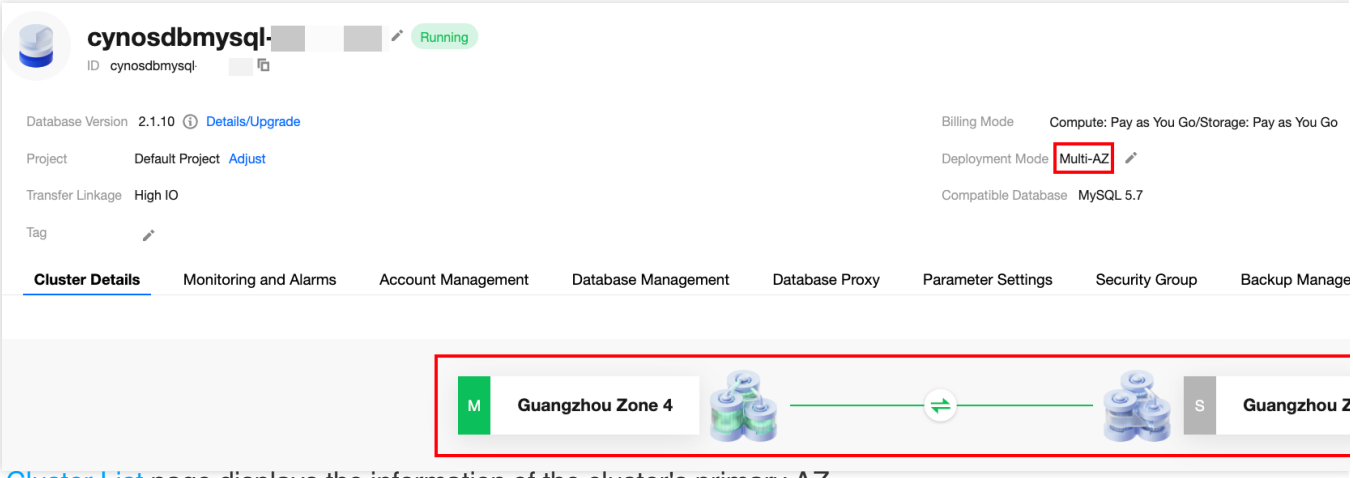
On the cluster list page, display according to the view mode actually used:

Tab View

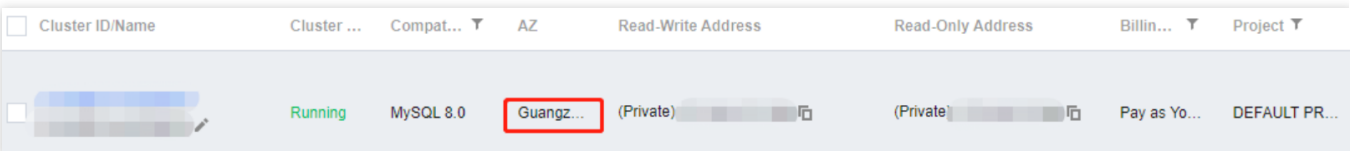
List view

1. Click the cluster in the left side of the Cluster List page to enter the cluster management page.

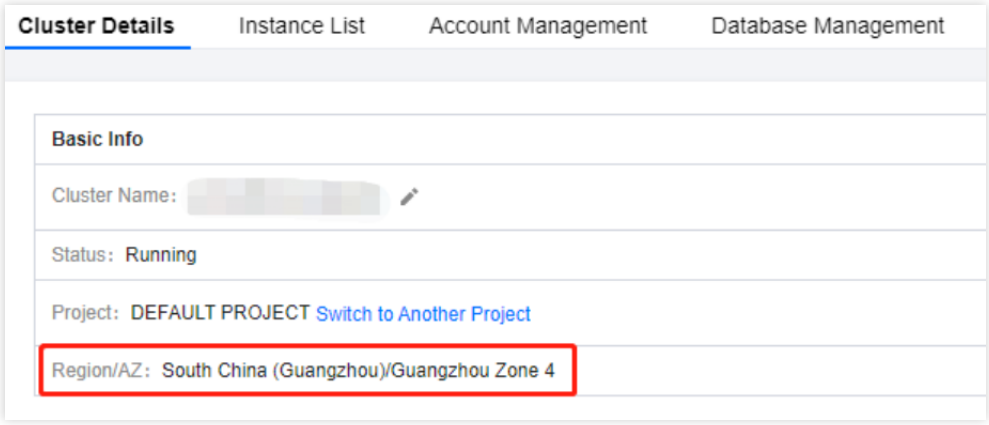
2. On the cluster management page, under **Deployment Mode**, and in the **Architecture** section of the cluster details, you can see the deployment mode and primary and secondary availability zones of the cluster.

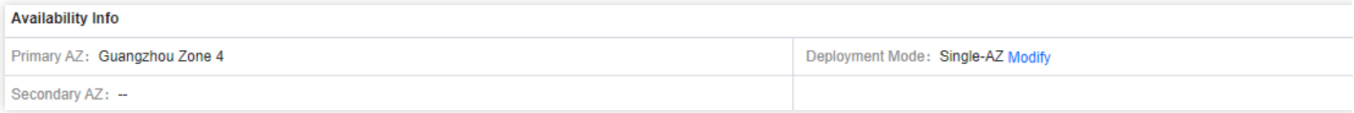


1. The Cluster List page displays the information of the cluster's primary AZ.



2. On the cluster details page, you can view the availability zones where the cluster deployment is distributed in **Basic Info** and **Availability Info** sections.





Setting Multi-AZ Deployment

Last updated：2024-06-17 15:05:06

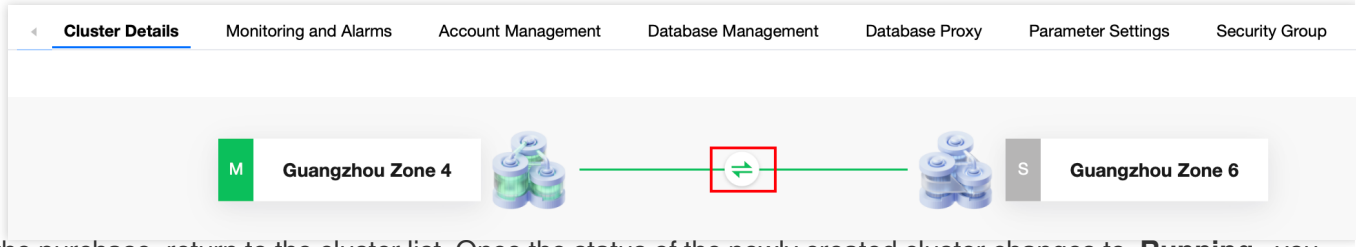
TDSQL-C for MySQL supports primary and secondary availability zones when you create a cluster. If a single availability zone deployment is selected during cluster creation, it also supports setting it to multi-availability-zone deployment after creation. This topic introduces how to set up multi-availability-zone deployment through the purchase page and console.

Overview

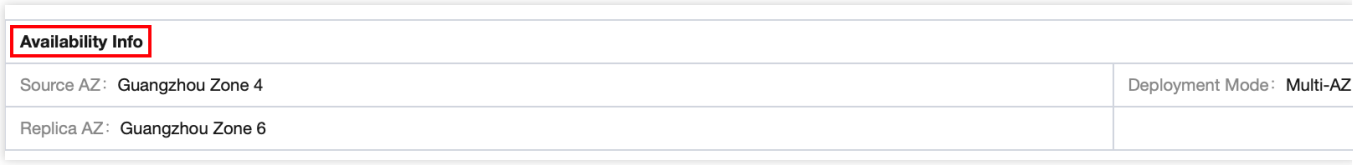
TDSQL-C for MySQL supports multi-AZ deployment in the same region, which has higher availability and better disaster recovery capability than single-AZ deployment.
TDSQL-C for MySQL clusters benefit from increased availability and durability when deployed in multi-AZ mode. When you provision a multi-AZ database cluster, TDSQL-C for MySQL will automatically create a primary database instance and synchronously replicate data to the secondary instance in another AZ. Each AZ runs on its own independent and physically distinct infrastructure designed for high reliability. In the event of an infrastructure failure, automatic failover to the secondary instance will be performed, so that you can resume database operations as soon as the failover is completed. As the endpoints of the database instances remain unchanged after failover, applications can resume database operations without manual intervention required.

Setting multi-availability-zone deployment on the purchase page when creating a cluster

1. Log in to the TDSQL-C for MySQL console and click **Create** in the cluster list.
2. In the **Database Configuration** item on the purchase page, select the desired region, and available primary and secondary AZs will be displayed below, which you can select as needed.
3. After selecting the database configuration, click **Next** to enter the **Basic Information** and **Advanced Configuration** items.
4. After completing the configuration and confirming that everything is correct, click **Buy Now**.
5. After the purchase is completed, you can check the availability zone information under the corresponding view.
Tab View
List view
After the purchase is completed, you can return to the cluster list. Once the status of the created cluster changes to **Running** , you can check the availability zone information through the architecture under cluster details on the cluster management page.



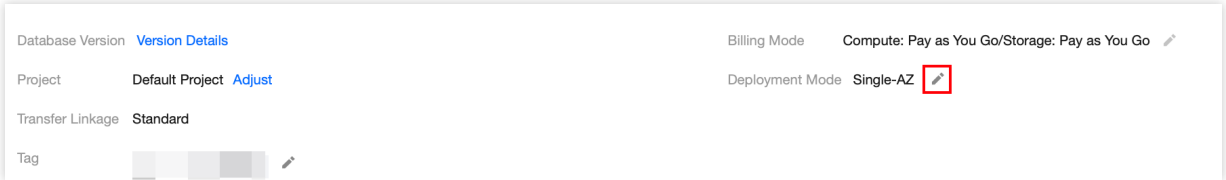
After the purchase, return to the cluster list. Once the status of the newly created cluster changes to **Running** , you can check the availability zones through **Availability Information** page in **Cluster Details** section by either browsing the cluster list page or clicking on the cluster ID.



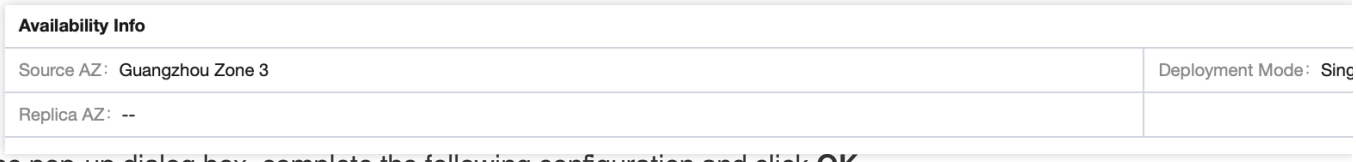
Setting multi-availability-zone deployment in the console after creating the cluster (condition: initially set as single-availability-zone deployment)

1. On the cluster list page, proceed according to the view mode actually in use to enter the deployment method modification interface.
Tab View
List view
1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.
2. On the cluster management page, click

after clicking on the deployment method.



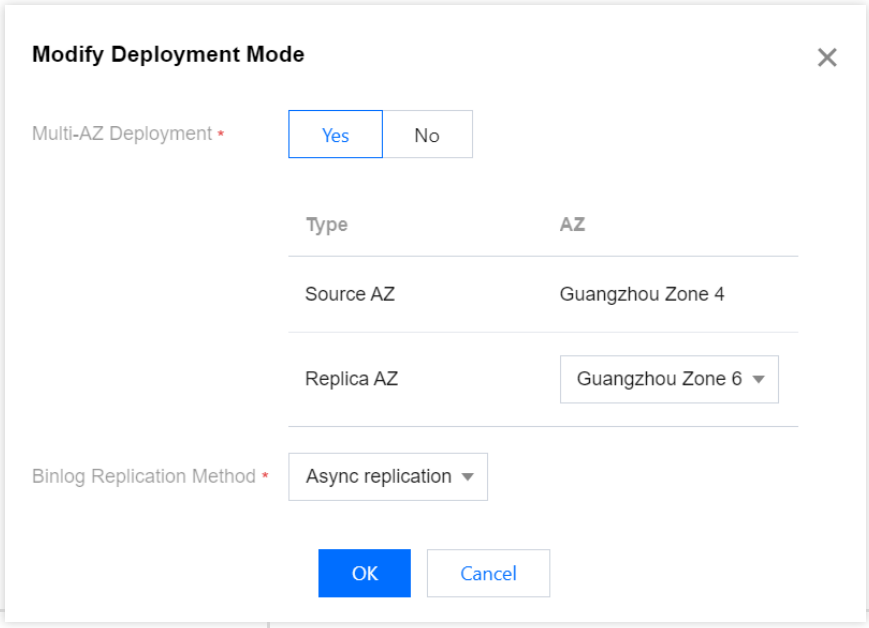
1. Log in to the TDSQL-C for MySQL console, in the cluster list, click the cluster ID or **Operation** in the **Management** column to enter the cluster details page.
2. On the cluster details page, in the **availability information** module, click **Modify** behind **Deployment Method** to change the availability zone.



2. In the pop-up dialog box, complete the following configuration and click **OK**.

Note:

Currently, only modifications to the secondary availability zone are supported. Changes to the primary availability zone are not supported.



Parameter	Description
Multi-AZ Deployment	Selecting **Yes** indicates changing the cluster from single-availability-zone to multi-availability-zone deployment.
Replica AZ	Select the secondary database's availability zone. The actual availability zones supported for selection depend on the **Modify Deployment Mode** window.
Binlog Replication Method	Select the binlog replication method. The default is **Async replication**. **Async**, **Semi-sync**, and **Strong sync** replication modes are supported.

Note:

If you select **Strong sync** as the binlog replication method, be aware that strong sync replication will affect the instance performance.

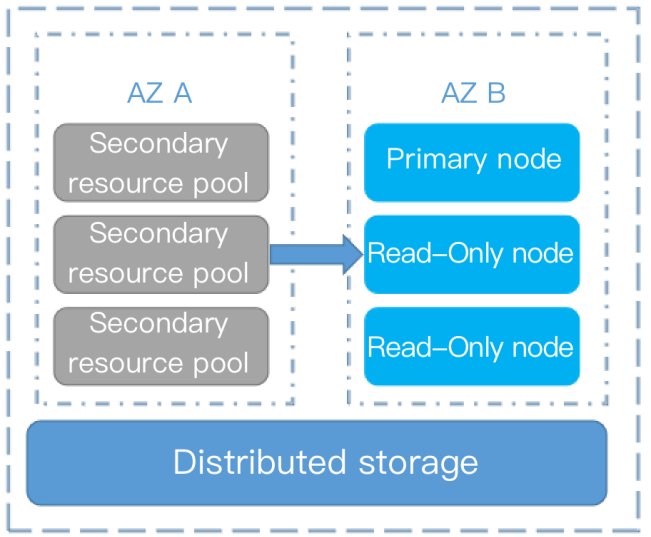
Switching AZs

Last updated：2024-04-25 14:50:14

The TDSQL-C MySQL version supports the switching of the primary and secondary availability zones within a cluster. This means that in the event of a cluster failure, the secondary database can be switched to the primary database to ensure system availability and data integrity. This document will guide you through the process of switching availability zones via the console.

Use Cases

The availability zone switching feature is applicable in scenarios such as disaster recovery and equipment failure.



Notes

The connection will be disconnected for about 2 to 5 seconds during the primary/secondary AZ switch. We recommend that you switch during off-peak hours and that your application has a reconnection mechanism.
The data does not need to be migrated if the target AZ is a secondary AZ. The system simply needs to switch the database compute nodes, allowing for a rapid cross-data center switch. This is often used in disaster recovery drills.
The cluster undergoes a comprehensive primary-secondary switch, meaning that all nodes within the cluster, including both read-write and read-only nodes, are switched (partial node switching is not supported).
The read-write and read-only addresses will not be changed after the switch.

Directions

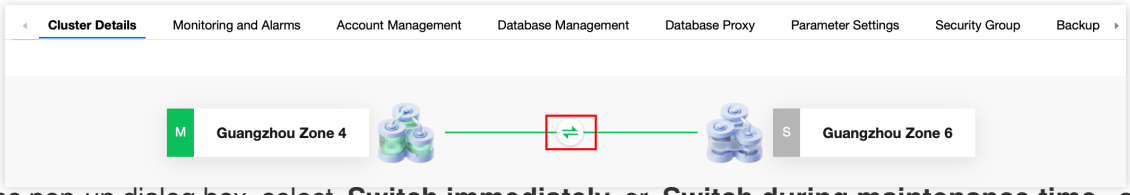
Automatic switch

This feature supports automatic switching between the primary and secondary AZs in case of failure, ensuring your business can continue to operate and your database is protected. In addition, you can follow the steps below to manually switch AZs.

Manual switch

If your business requires an immediate switch of the primary and secondary AZs for the cluster or a switch at a designated time, you can perform a manual AZ switching operation.
On the cluster list page, perform the operation according to the actual view mode:
Tab View
List view
1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click Target Cluster to enter the cluster management page.
2. On the Cluster Management page, between the primary and secondary AZs in the architecture diagram under the cluster details, click

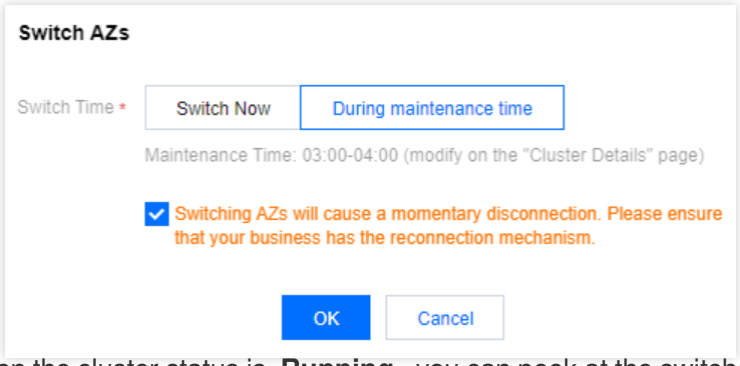
to switch the primary and secondary AZs.



3. In the pop-up dialog box, select **Switch immediately** or **Switch during maintenance time** , and check Switching AZs will cause a momentary disconnection. Ensure that your business has the reconnection mechanism, and click **Confirm** .

Note:

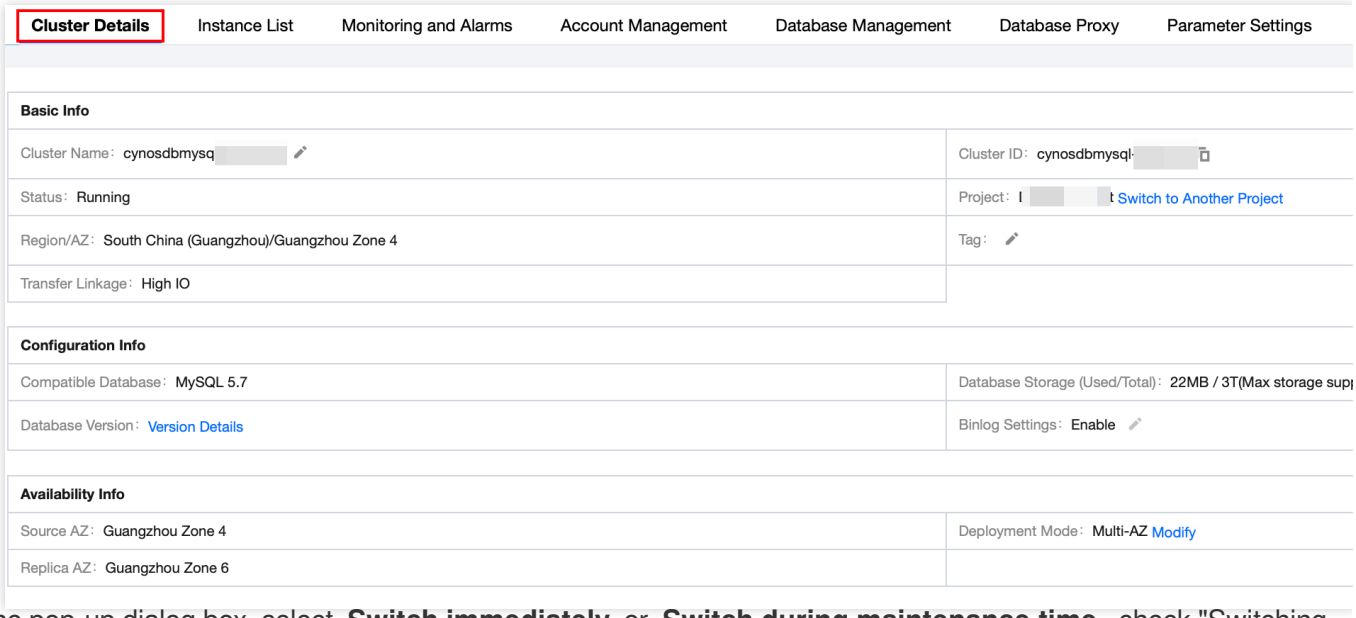
Switching AZs will cause a momentary disconnection. Ensure that your business has the reconnection mechanism. Select the **Instance List** page, click the corresponding Instance ID to enter the instance details page, and click edit behind **Maintenance Information** to set or modify the maintenance time.



4. When the cluster status is **Running** , you can peek at the switched AZ in the architecture diagram under **Cluster Details** .

1. Log in to the TDSQL-C for MySQL console, locate the cluster list, and click a cluster ID or **Operation** column's **Management** to enter the cluster details page.

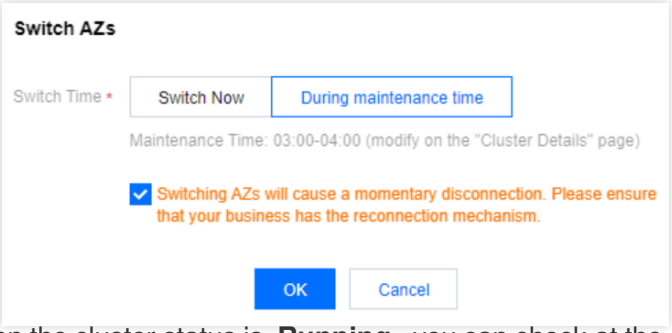
2. In the **Availability Information** module on the cluster details page, click **Switch AZs** .



3. In the pop-up dialog box, select **Switch immediately** or **Switch during maintenance time** , check "Switching AZs will cause a momentary disconnection. Ensure that your business has the reconnection mechanism", and click **Confirm** .

Note:

Switching AZs will cause a momentary disconnection. Ensure that your business has the reconnection mechanism. Select the **Instance List** page, click the corresponding Instance ID to enter the instance details page, and click edit behind **Maintenance Information** to set or modify the maintenance time.



4. When the cluster status is **Running** , you can check at the switched AZ in the Availability Information section on the **Cluster Details page** .

FAQs

How long does it take to complete an AZ switch?

If the target AZ is a secondary AZ, the data does not need to be migrated when the primary AZ changes. The system simply needs to switch the database compute nodes, allowing for a rapid cross-data center switch.

Will the read-write address be changed after the entire cluster is switched?

When the entire cluster undergoes a switch, all the read-write nodes and read-only nodes within the cluster are switched, but the read-write addresses and read-only addresses will not change.

After manual/automatic switch, if an instance is upgraded, will the upgrade affect or reset the information of the primary and secondary AZs?

Switching does not affect or reset the information of the primary and secondary AZs.

Modifying AZ Deployment

Last updated：2024-06-17 15:27:41

This document describes how to switch the instance deployment mode between single-AZ and multi-AZ in the TDSQL-C for MySQL console.

Upgrading from Single-AZ to Multi-AZ

On the cluster list page, proceed according to the actually used view mode:

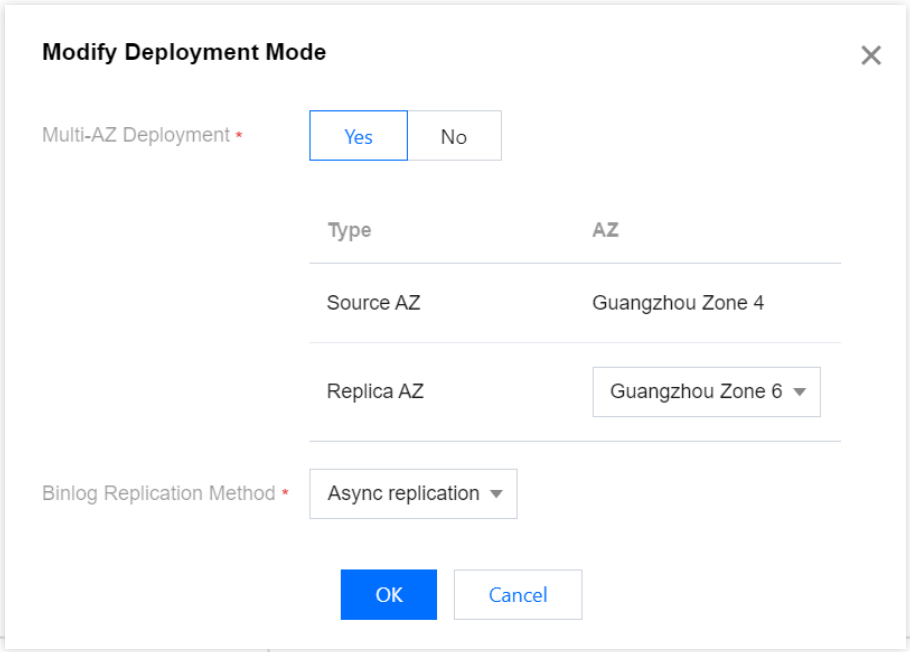
Tab View

List view

1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.
2. On the cluster management page, click

next to **Deployment Mode**.

3. In the pop-up window, select **Yes** for **Multi-AZ Deployment**, select a secondary availability zone and binlog replication method, and click **OK**.



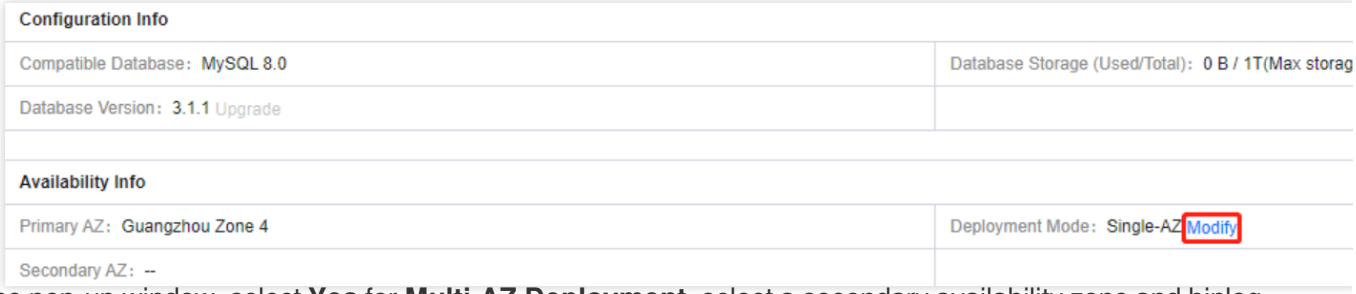
Parameter	Description
Multi-AZ Deployment	Selecting **Yes** indicates changing the cluster deployment mode from single-AZ deployment to multi-AZ deployment.

Secondary AZ	Select the availability zone of the secondary database. The actual availability zones supported for selection depend on the **Modify Deployment Mode** window.
Binlog Replication Method	Select the binlog replication method. The default is **Async replication**. **Async**, **Semi-sync**, and **Strong sync** replication modes are supported. **Note:** Note that **Strong sync replication** can impact the instance performance.

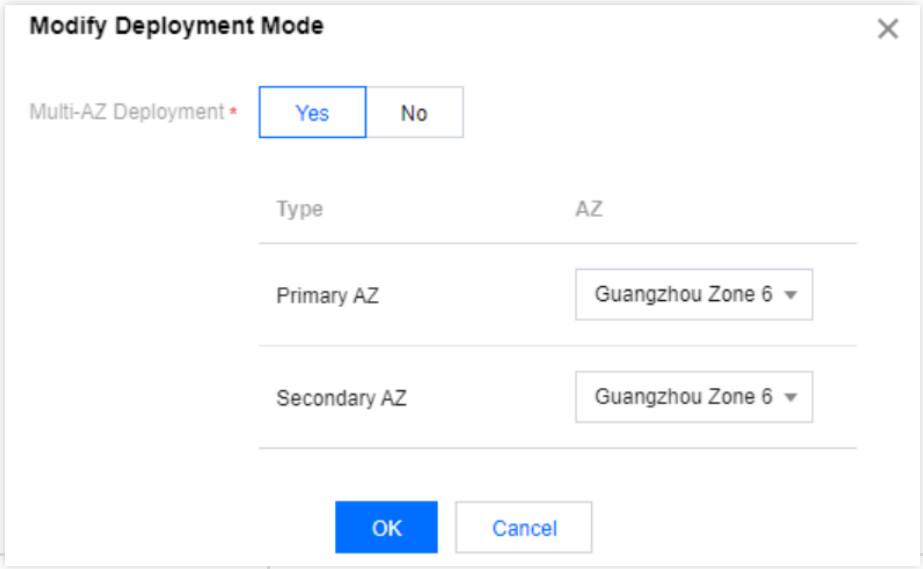
4. When the cluster status becomes **Running**, the primary and secondary availability zone information can be viewed in the architecture diagram under the cluster details.

1. Log in to the TDSQL-C for MySQL console and click a cluster name in the cluster list or **Manage** in the **Operation** column to enter the cluster details page.

2. In the **Availability Info** module on the cluster details page, click **Modify** after **Deployment Mode**.



3. In the pop-up window, select **Yes** for **Multi-AZ Deployment**, select a secondary availability zone and binlog replication method, and click **OK**.



Parameter	Description
Multi-AZ Deployment	Selecting **Yes** indicates changing the cluster deployment mode from single-AZ deployment to multi-AZ deployment.
Secondary AZ	Select the availability zone of the secondary database. The actual availability zones

	supported for selection depend on the **Modify Deployment Mode window**
Binlog Replication Method	Select the binlog replication method. The default is **Async replication**. **Async**, **Semi-sync**, and **Strong sync** replication modes are supported. **Note:** Note that **Strong sync replication** can impact the instance performance.

4. When the cluster status becomes **Running**, you can view the information of the primary and secondary AZs on the cluster details page.

Downgrading from Multi-AZ to Single-AZ

On the cluster list page, proceed according to the actually used view mode:

Tab View

List view

1. Log in to the TDSQL-C for MySQL console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.
2. On the cluster management page, click

next to **Deployment Mode**.

3. In the pop-up window, select **No** for **Multi-AZ Deployment** and click **OK**.

Note:

Only the secondary AZ can switch between single-AZ and multi-AZ deployment mode, but the primary AZ cannot.

1. Log in to the TDSQL-C for MySQL console, locate the cluster list, click a cluster ID or **Operation** column's **Manage** to enter the cluster details page.
2. In the **Availability Info** module on the cluster details page, click **Modify** after **Deployment Mode**.

3. In the pop-up window, select **No** for **Multi-AZ Deployment** and click **OK**.

Note:

Only the secondary AZ can switch between single-AZ and multi-AZ deployment mode, but the primary AZ cannot.

FAQs

Where can I query the AZ information of a cluster?

List View

In the **AZ** column in the cluster list.

In **Basic Info** > **Region**/**AZ** on the cluster details page.

In **Availability Info** on the cluster details page.

Tab View

The architecture diagram, which can be found under the **Cluster Details** section.

Will my cluster run normally during the AZ modification?

During the start and modification operations of the AZ modification task, normal access to read-only and read-write instances will be unaffected.

The database connection addresses (cluster's read-write address and read-only address) will remain unchanged after the AZ is modified, so you can continue to access your cluster at these addresses.

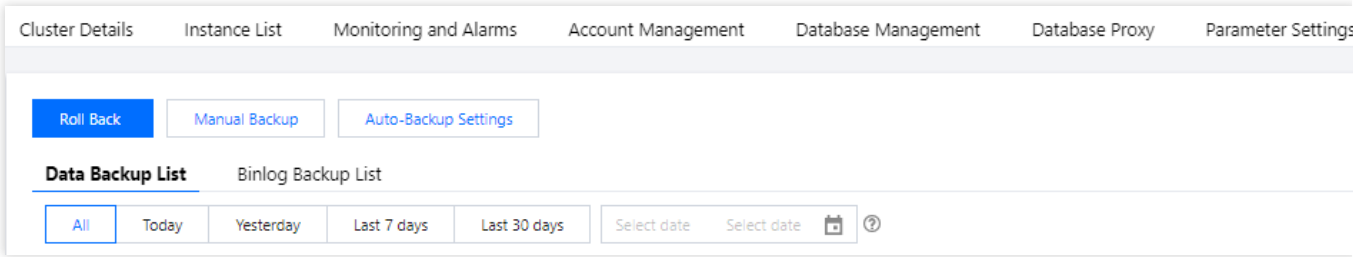
Backup and Restoration
Backup and Rollback Overview

Last updated：2023-08-22 15:15:16

Data is the core asset of enterprises. As your business develops, your data grows in a large-scale and explosive manner. Business applications require real-time, online, and fast data processing. It is more and more challenging for database Ops engineers to protect the data integrity, as data loss may occur for a variety of causes, such as accidental deletion, system vulnerabilities and viruses, hardware failures, and even natural disasters. Therefore, backup and rollback are of significant importance to databases.

Backup Overview

TDSQL-C for MySQL supports data backup and binlog backup. A complete data backup supplemented by a follow-up binlog backup allows you to restore the entire TDSQL-C for MySQL cluster or specified databases/tables to any time point.
You can visually query the field information in the data backup list, such as file name/alias, backup time, start time, end time, backup type, backup mode, backup size, status, and operation. You can also query similar fields in the binlog backup list, including file name, binlog backup start time, binlog backup end time, backup size, and operation. This helps you easily view and manage historical backups.
The backup and rollback console UI is as shown below:



Data backup

Data backup means generating a backup file containing all the data in a cluster at a specific time point. TDSQL-C for MySQL supports logical backup and snapshot backup. The former is a full backup, while the latter is a full + incremental backup.
Logical backup: It supports full backup only, where the logical structure and content of a database are stored as SQL statements. It backs up database objects, including tables, indexes, and stored procedures. This mode features a more refined backup granularity at the database or table level and a lower impact on the performance; however, it is slow and space-intensive.

Snapshot backup: It uses the redirect-on-write (ROW) technology to take snapshots of the disks at the storage layer for backup. It features fast backup in seconds, imperceptibility to the computing layer, and low space usage.
Full backup: It copies all the data at a specific time point.
Incremental backup: It backs up only new or modified files based on the last backup.

Backup types

Backup Type		Strengths/Shortcomings	Object	Mode	Download	Deletion
Logical backup	Full backup	Strengths: The backup is at the database or table level, with a smaller impact on the database performance. Shortcomings: The backup task locks the database, takes a long time, and uses a lot of space.	Objects such as tables, indexes, and stored procedures, as well as the entire cluster	Manual	✓	✓
Snapshot backup	Full backup	Strengths: The backup task can be completed in seconds and imperceptible to the business, with a small space used. Shortcomings: The backup file cannot be downloaded.	The entire cluster	Manual	×	×
				Automatic	×	×
	Incremental backup			Automatic	×	×

Binlog backup

Backup Type		Strengths/Shortcomings	Object	Mode	Download	Deletion
Binlog backup	Incremental backup	Strengths: Incremental data is recorded and can be restored to any time point. Shortcomings: The binlog backup task lowers the instance's write performance.	The entire cluster	Automatic	✓	x

A binlog backup is the incremental data generated after all the data in the cluster is backed up to a file at a specific time point. TDSQL-C for MySQL generates a large number of binlogs when executing large transactions or lots of

DML operations, which are uploaded to the cloud storage and displayed in the binlog backup list in the console. The binlog backup mode applies to operation log storage.
TDSQL-C for MySQL relies on redo logs rather than binlogs for rollback, so that even if binlog is disabled, data can still be rolled back to any time point, and the instance performance can be increased by over 30%.
Note:
Currently, TDSQL-C for MySQL doesn't support disabling binlog backup, which will be made available after the network architecture upgrade is completed.

Note

Only one manual backup task can be performed per hour. Automatic backup tasks are performed according to your configuration, which is once per day by default.
Manual backup files can be manually deleted from the backup list. They are retained as long as they are not deleted; therefore, regularly delete those no longer needed to free up the space.
Automatic backups cannot be deleted manually. You can set a retention period so that they will be deleted automatically upon expiration.
You can query the binlog backup size in the log backup list. The total binlog backup size is the sum of sizes of all binlog backups.

Rollback Overview

TDSQL-C for MySQL supports data restoration to a specific time point through the rollback feature, minimizing potential system losses.
TDSQL-C for MySQL can roll back databases/tables to the original cluster and roll back an entire cluster (clone) to a new cluster. You can choose different rollback methods based on your business needs.

Rollback Method

Rollback by backup file: This method restores the cluster to the data file state of a backup file. The selection range of the backup file is determined by the data backup retention period you set.
Rollback by time point: This method restores the cluster to any time point within the binlog backup retention period you set.

Backup and Rollback Fees

Last updated：2023-05-31 16:10:39

This document describes the backup and rollback fees.

Billing

- **Backup storage space**: Backup files take up the storage space. Backup modes include automatic and manual, and backup objects include binlog and data. Storage space used by all backup files incurs fees under this billable item.

  Note：
  The backup storage space is free of charge for now, and we will notify you of future billing changes.

- **Rollback**: TDSQL-C for MySQL supports rollback to the original cluster or a new cluster according to the backup file or time point. The former doesn't incur fees, while the latter incurs fees after the new cluster is generated. For billing details, see Product Pricing.

Performing Backup

Automatic Backup

Last updated：2023-08-22 15:53:50

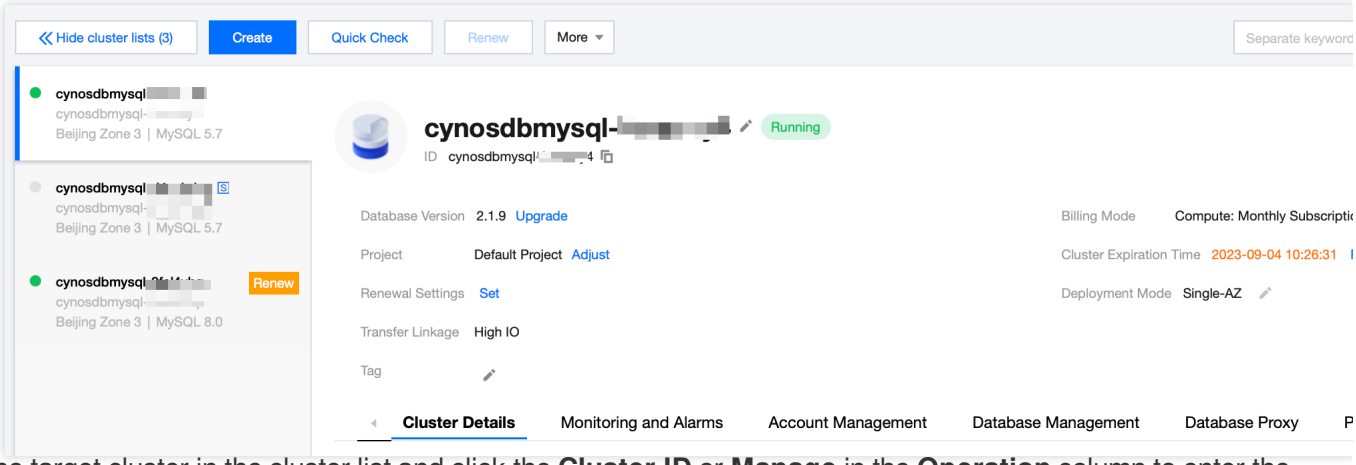
TDSQL-C for MySQL supports automatic backup and manual backup. Automatic backup is enabled by default. You can configure backup policies to automatically back up data as scheduled for data security.
After a cluster is created, TDSQL-C for MySQL generates backup files every 6-48 hours on a 24/7 uninterrupted basis, which has no impacts on the instance performance. You can also set the backup retention time in the console as needed.
This document describes how to configure automatic backup in the console.

Note

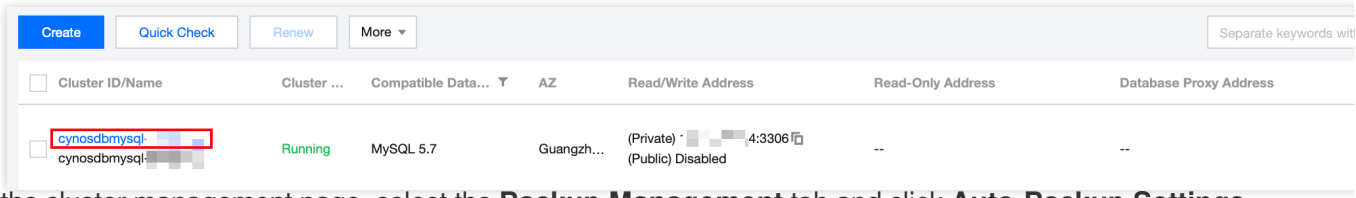
Automatic backup files cannot be deleted, but you can set the data and binlog backup retention period, so that they will be deleted upon expiration.
You can view the **Backup Time** field in the data backup list to determine the exact time point of the data image in a backup file.

Directions

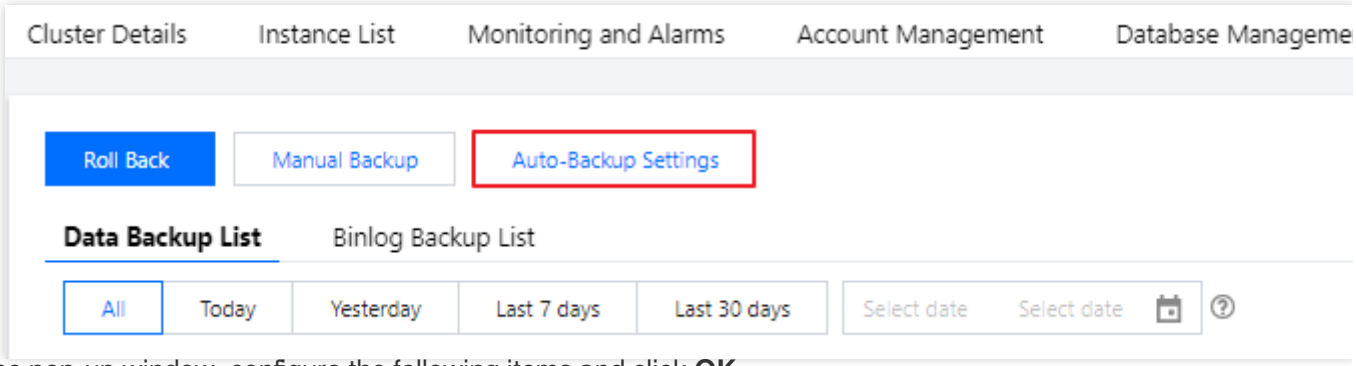
1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.
Tab view
List view
Click the target cluster in the cluster list on the left to enter the cluster management page.



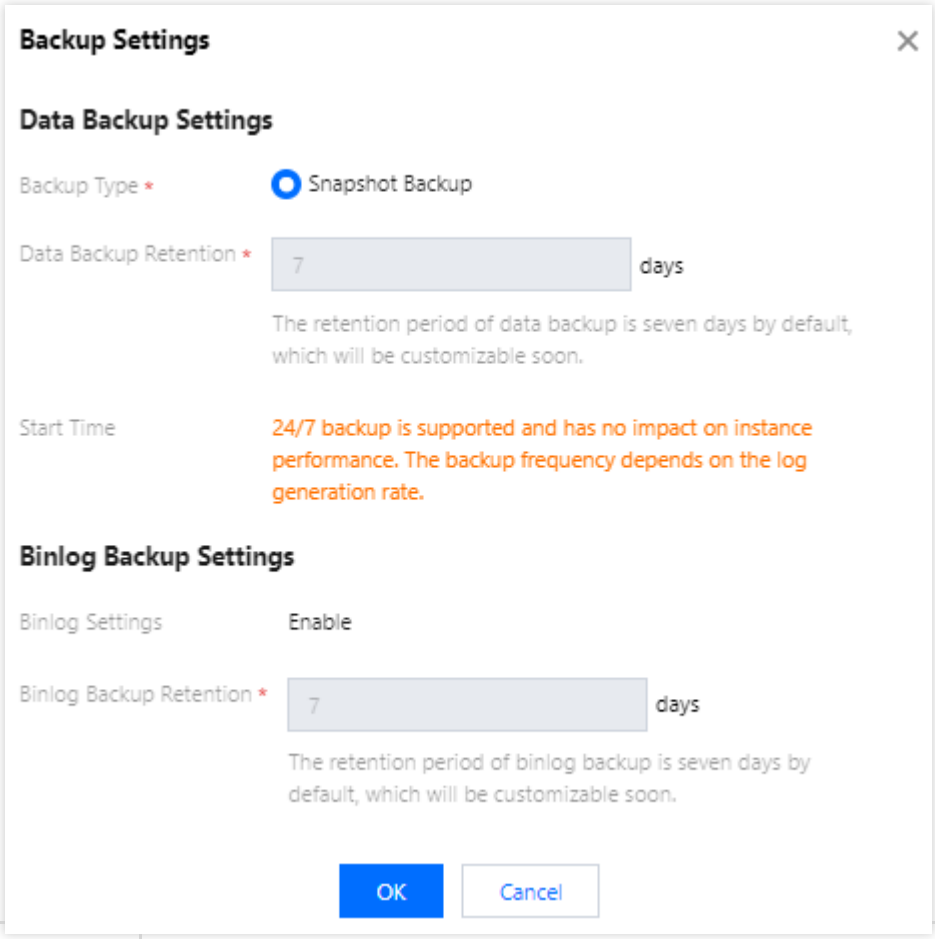
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Backup Management** tab and click **Auto-Backup Settings**.

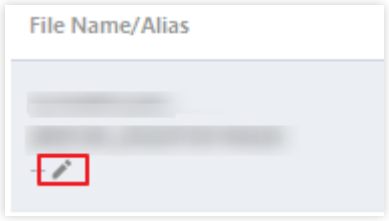


4. In the pop-up window, configure the following items and click **OK**.

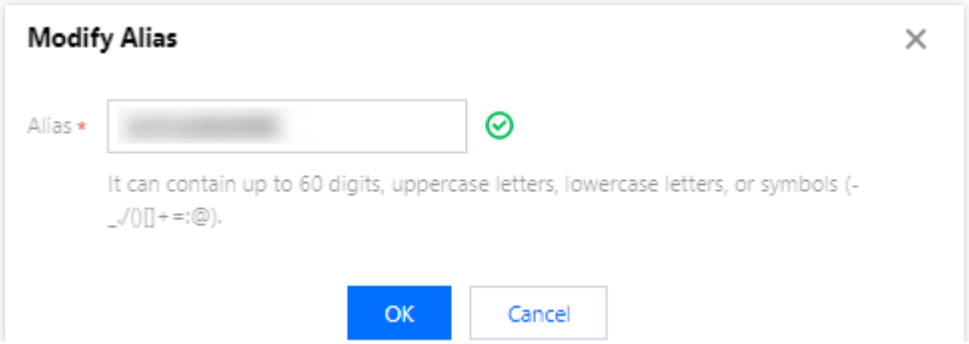


Parameter	Description
Backup Type	Snapshot backup is selected by default.
Data Backup Retention	Data backup files can be retained for 7–1,830 days.
Start Time	Backup files are generated every 6-48 hours on a 24/7 uninterrupted basis, which has no impacts on the instance performance. You can set the **Backup Time** field below the backup list to determine the exact time point of the file backup.
Binlog Backup Retention	Binlog backup files can be retained for 7–1,830 days, which cannot be shorter than the data backup retention period.

5. The name of an automatic backup file is automatically generated. You can click the edit icon after the alias in the backup list to modify it.



The settings window is as follows:



Manual Backup

Last updated：2023-05-31 16:10:39

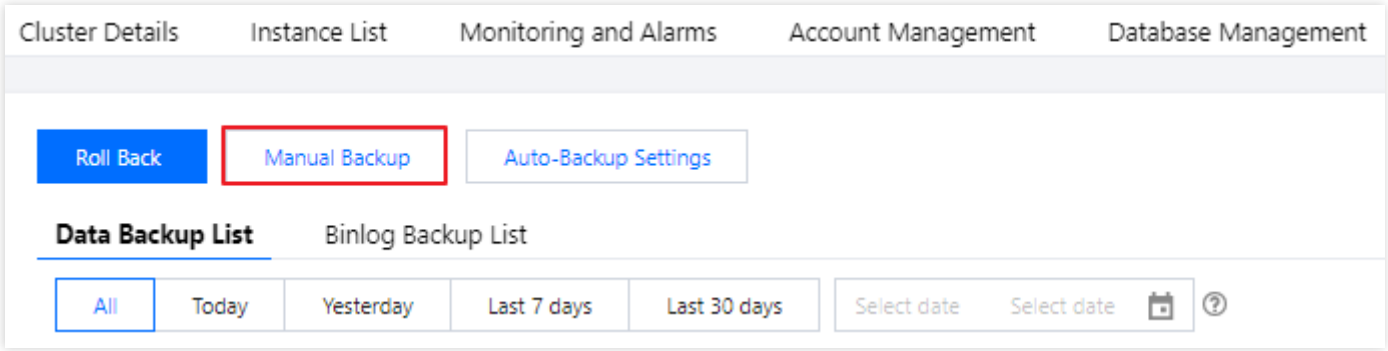
TDSQL-C for MySQL supports automatic backup and manual backup. This document describes how to create a manual backup in the console.

Note：

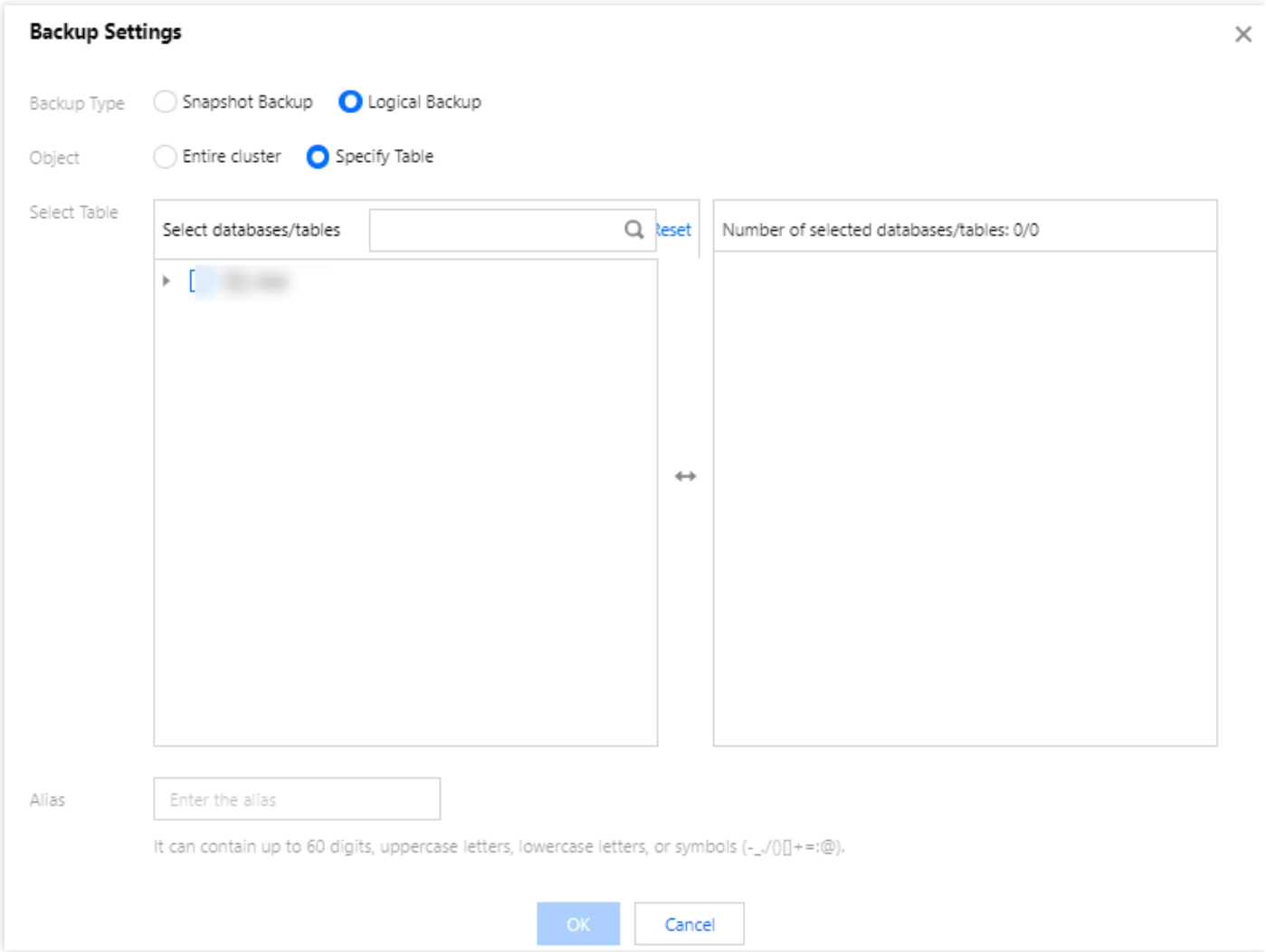
- With manual backup, you can use snapshot backup to manually back up the entire cluster or use logical backup to back up the entire cluster or specified databases/tables.
- You can delete manual backups from the backup list in the console to release backup space.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select the **Backup Management** tab and click **Manual Backup**.



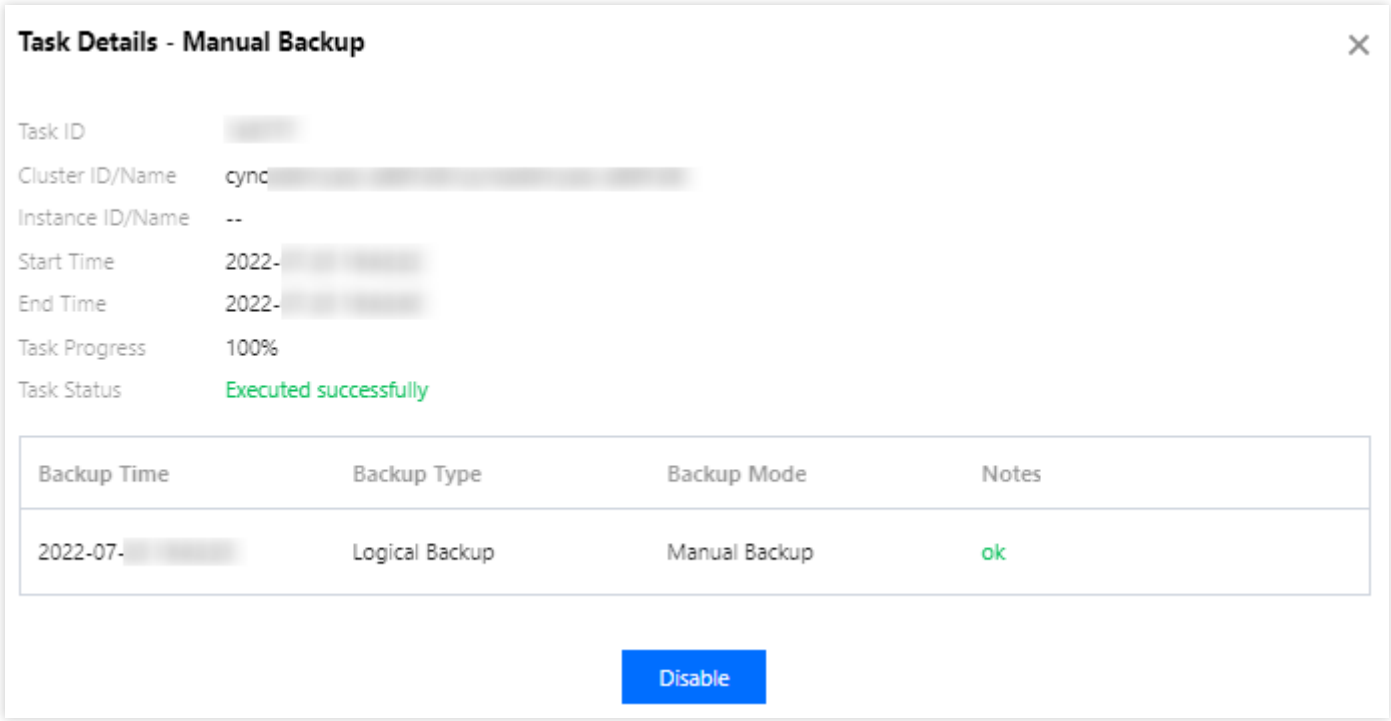
4. In the pop-up window, select the backup type and object, set the backup file alias, and click **OK**.



- **Backup Type**: Snapshot backup and logical backup are supported.
- **Object**: Snapshot backup applies to the entire instance, while logical backup applies to the entire instance or specified databases/tables.
- **Alias**: You can set a file alias when creating a manual backup, which can contain up to 60 digits, letters, or symbols (-_./()[]+=:;@).

> Note：
> We recommend you perform a manual logical backup task during off-peak hours, as it will lock the database and affect database use.

5. After creating a manual backup task, you can view the task progress in the **Task List** on the left sidebar. You can also view the execution status by clicking **Backup Details** in the *Operation column.



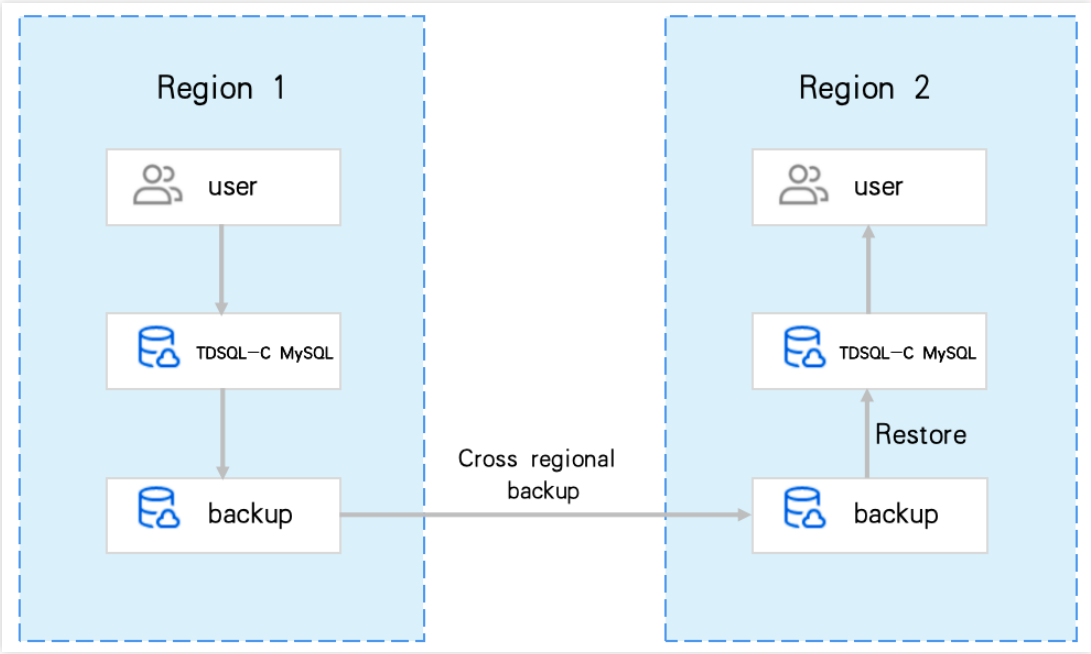
Cross-Region Backup

Last updated：2024-03-07 09:45:49

Cloud Native Database TencentDB for CynosDB provides a cross-region backup function, enabling users to store backup files in a different region for storage to enhance regulatory compliance and disaster recovery ability and improve data reliability. Currently, TencentDB for CynosDB cluster supports cross-region backup settings for automatic logical backup, binlog backup, and manual logical backup (for the entire cluster).
This article presents the function of cross-region backup.

Background

Data is an important part of enterprise operations. Although the information technology brings convenience, it also reveals that electronic data and stored information are very vulnerable to damage or loss. Any incident, such as natural disaster, system failure, maloperation, or virus, can cause interruption of business operations or even disastrous losses. Therefore, ensuring the security and integrity of core data is a top priority of every enterprise. The cross-region backup feature of Cloud Native Database TencentDB for CynosDB can be used to store backup files in another region so as to minimize data corruptions caused by natural disasters or system failures. This feature ensures the high availability, security, and recoverability of data and implement various features, such as remote backup and restoration, remote disaster recovery, long-term data archive, and regulatory compliance.



Cross-Region Backup Limitations

Currently, only high IO edition clusters support cross-region backup capabilities in automatic logical backup, binlog backup, and manual logical backup (for the entire cluster).

Notes on Cross-Region Backup

Cross-region backup can restore data to the region where the cluster or the backup is.
Cross-region backup doesn't affect the local default backup, and both coexist after cross-region backup is enabled.
Cross-region backup will be triggered after the local default automatic backup is complete, that is, the default automatic backup is dumped to the storage device for cross-region backup.

Billing Description

Cross-region backup is currently in the public testing phase, and is presently available for use without charge.

The Difference between Cross-region Logical Backup and Local Automatic Backup

Comparison Item	Cross-Region Logical Backup	Local Automatic Backup
Is it enabled by default	It is disabled by default and requires manual enabling.	Enabled by default
Backup storage region	Backup is stored in the specified remote region (up to three regions)	Backup is stored in the region where the cluster is located
Backup and restoration	Data can be restored to: Original cluster New cluster in the target region	Can be restored to: Original cluster New cluster in the current region

Supported Regions

The regions supporting cross-region backup span all the regions in TencentDB for CynosDB.

Enabling Cross-Region Backup for Automatic Logical Backup

Note:

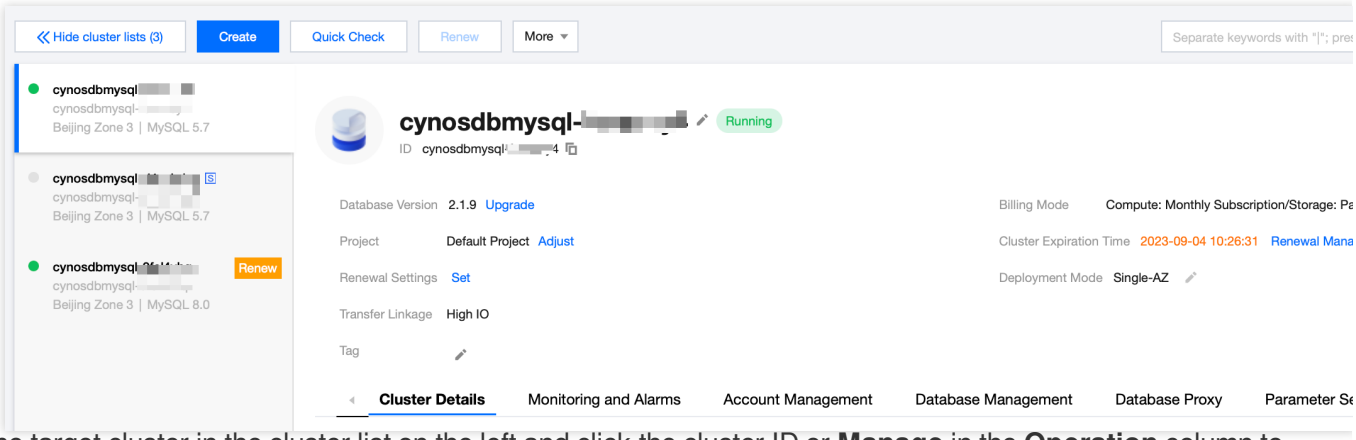
After the enabling of cross-region backup of automatic logical backup, it is recommended to synchronously enable the Binlog cross-region backup for the corresponding regions, providing convenience for time point recovery in the corresponding regions.

1. Log in to the TencentDB for CynosDB console.
2. At the top, select a region and operate according to the console view pattern currently in use.

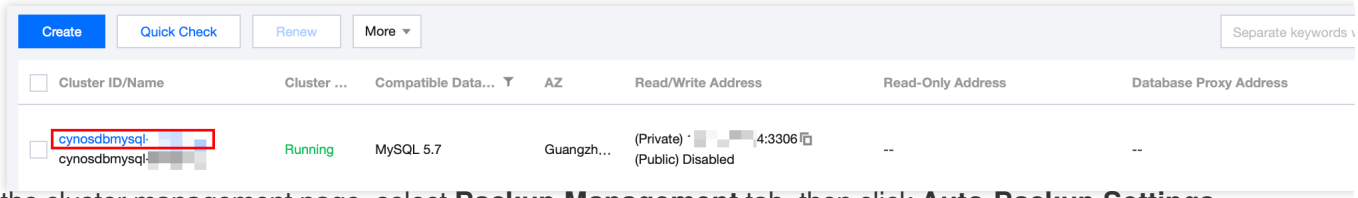
Tab View

List View

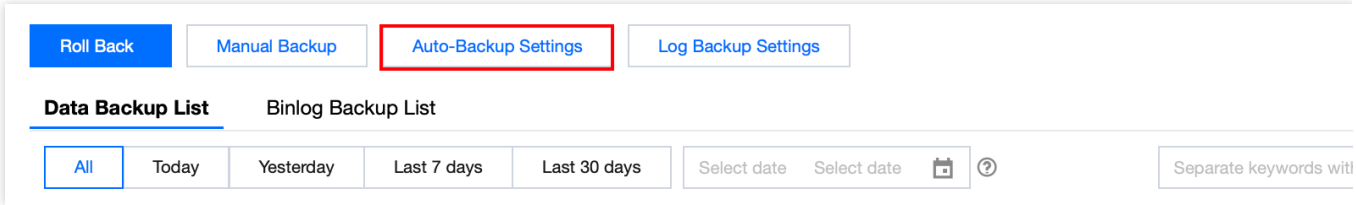
On the left side of the cluster list, click **Target Custer** to enter the cluster management page.



Find the target cluster in the cluster list on the left and click the cluster ID or **Manage** in the **Operation** column to access the cluster management page.



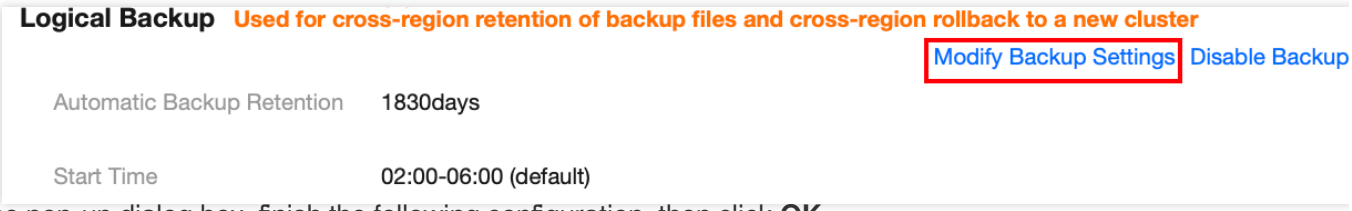
3. On the cluster management page, select **Backup Management** tab, then click **Auto-Backup Settings**.



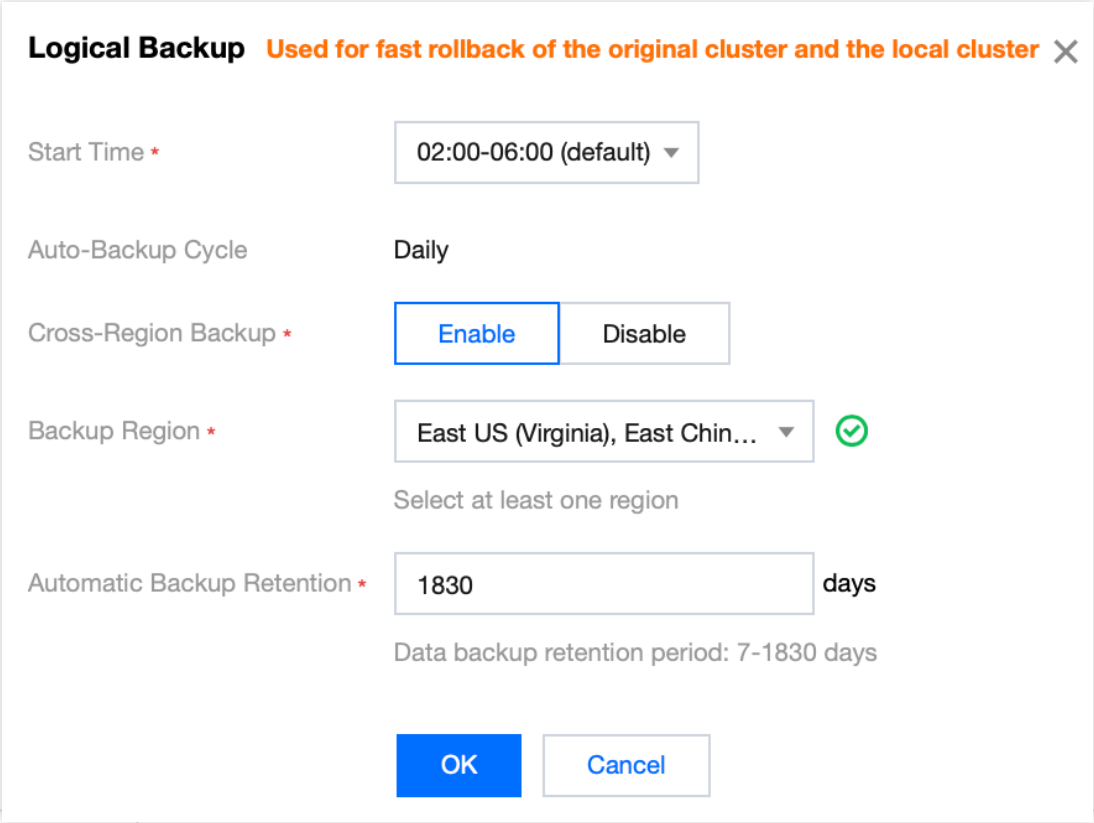
4. Under Logical Backup, click **Modify Backup Settings**.

Note:

If you have enabled automatic logical backup, but have not activated the cross-region backup feature for automatic logical backup, please click on **Modify Backup Settings**.



5. In the pop-up dialog box, finish the following configuration, then click **OK**.



Parameter	Description
Start time	Set the daily start time of automatic logical backup. Set by default from 02:00 to 06:00.
Automatic backup cycle	Daily is set by default. **Note:** The current version only supports an automatic backup cycle being daily. Due to logical backup speed limitations, the backup cycle for some larger data volume clusters is expected to extend to more than a day (if the logical backup for that day is not completed, the next automatic logical backup will not occur until the day after this logical backup is completed).
Cross-region backup	Select to enable cross-region backup.
Backup region	Choose the regions for backup storage, supporting up to three different regions.
Automatic	Backups can be retained from 7 to 1830 days.

backup retention	

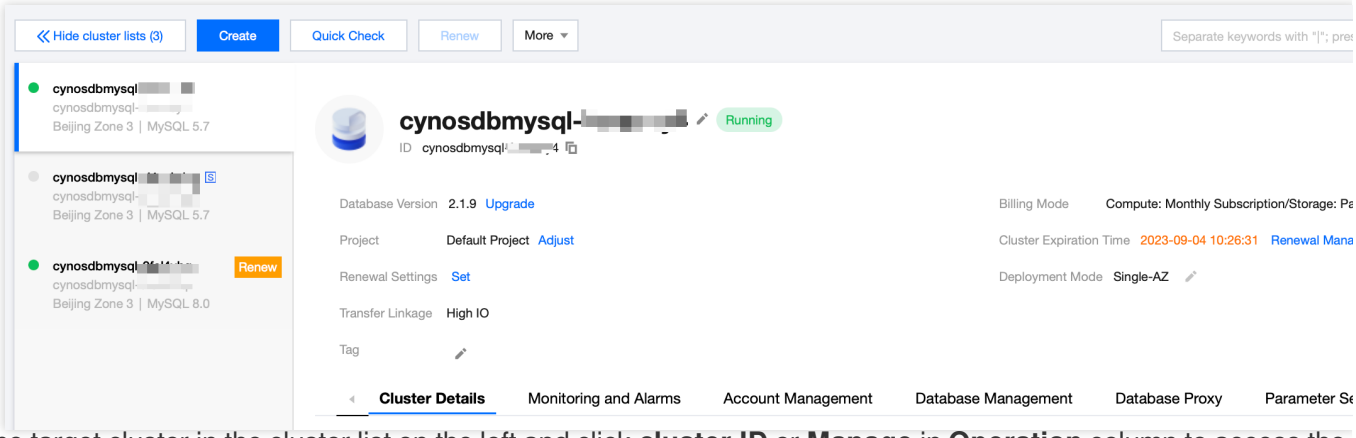
Enabling Cross-Region Backup of binlog Backup

1. Log in to the TencentDB for CynosDB console.
2. At the top, select the region and operate according to the console view pattern currently in use.

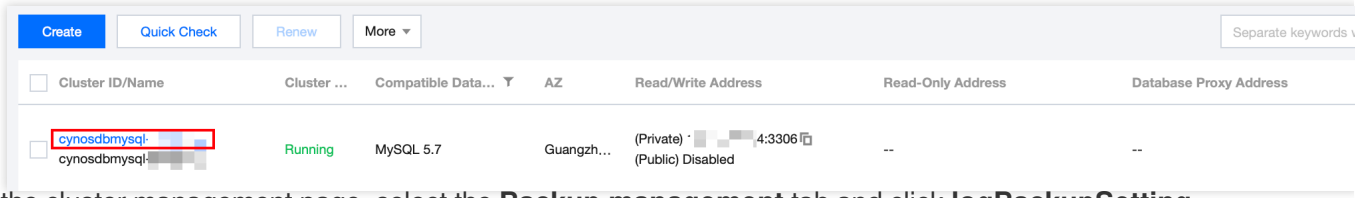
Tab View

List View

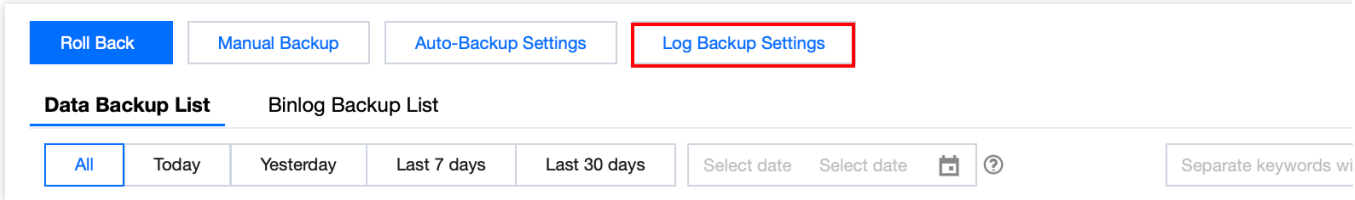
On the left side of the cluster list, click on the desired cluster to enter the cluster management page.



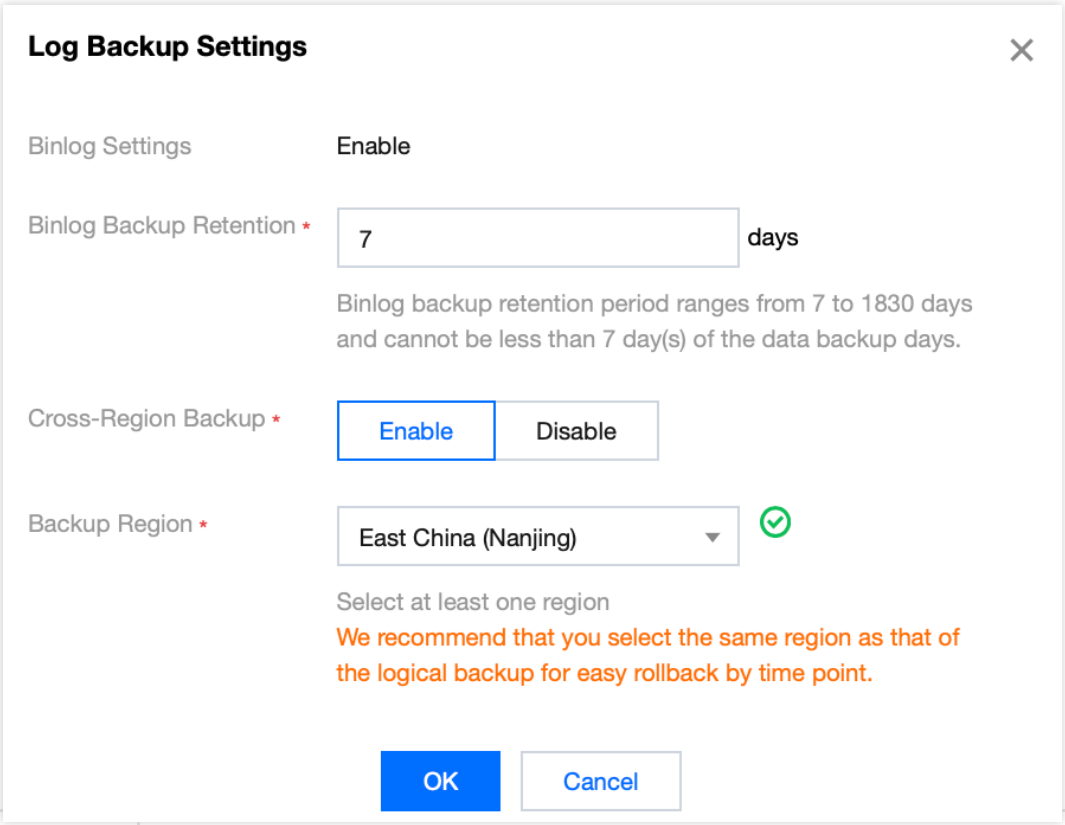
Find the target cluster in the cluster list on the left and click **cluster ID** or **Manage** in **Operation** column to access the cluster management page.



3. On the cluster management page, select the **Backup management** tab and click **logBackupSetting**.



4. In the pop-up dialog box, complete the following configuration and click **OK**.



Parameter	Description
binlog setting	Enabled by default.
Binlog backup retention	Binlog backup retention period lasts 7 to 1,830 days, which must not be less than the data backup retention period of 7 days.
Cross-region backup	Select if the cross-region backup should be enabled. If it is activated, binlog backups can be stored in other regions. If it is not activated, binlog backups will be kept in the region where the cluster is currently located by default.
Backup region	When enabling cross-region backup, you can select the region where the backup is stored, supporting up to three different regions.

Enabling Manual Logical Cross-Region Backup

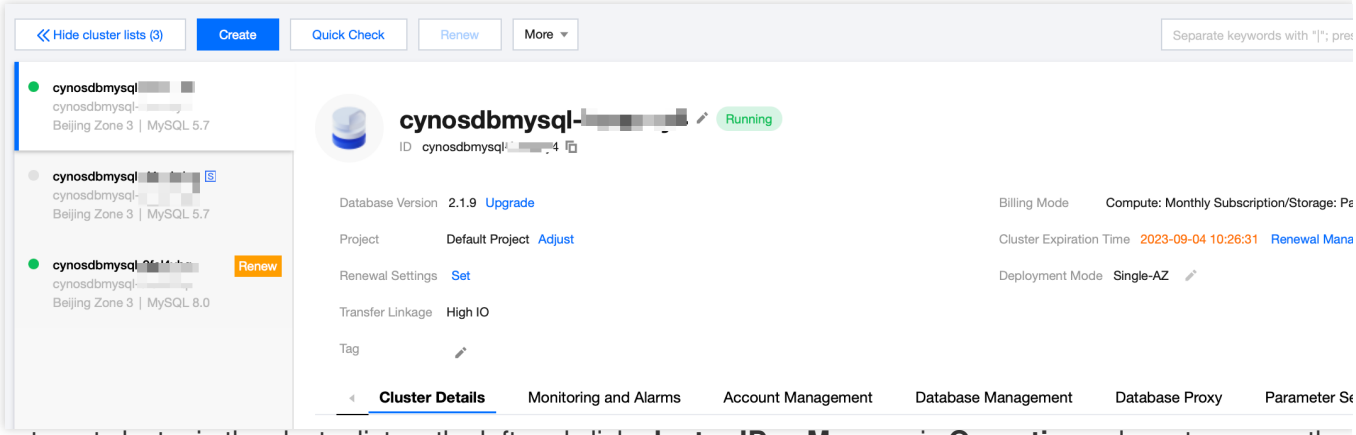
1. Log in to the TencentDB for CynosDB console.

2. At the top, select the region and operate according to the console view pattern currently in use.

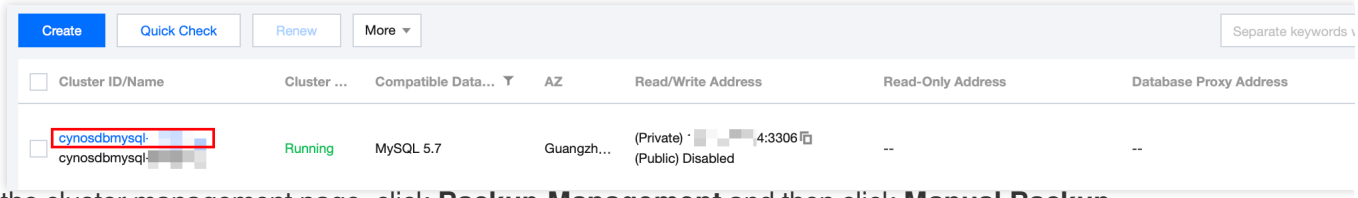
Tab View

List View

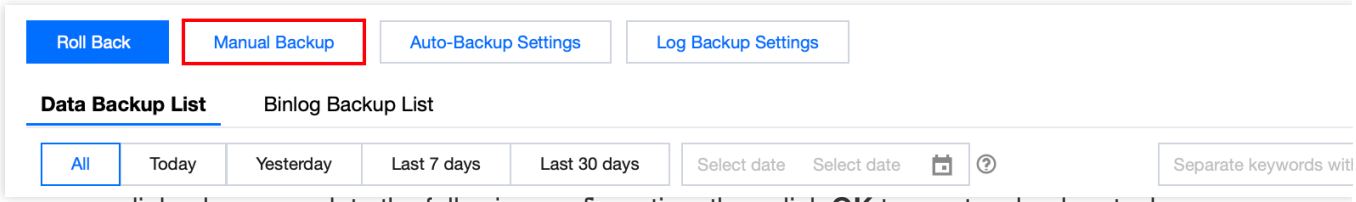
On the left side of the cluster list, click the desired cluster to enter the cluster management page.



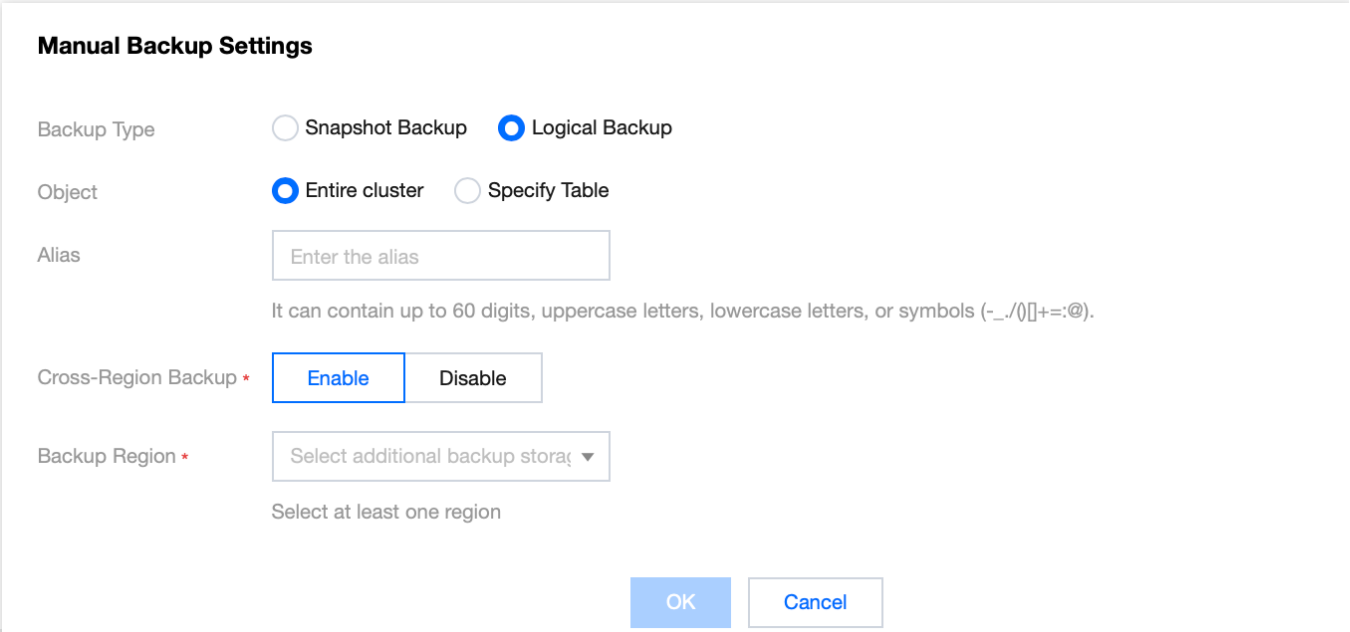
Find the target cluster in the cluster list on the left and click **cluster ID** or **Manage** in **Operation** column to access the Cluster Management page.



3. On the cluster management page, click **Backup Management** and then click **Manual Backup**.



4. In the pop-up dialog box, complete the following configuration, then click **OK** to create a backup task.



Parameter	Description
Backup type	Choose logical backup. **Note:** We recommend you perform a manual logical backup task during off-peak hours, as it will lock the database and affect database use.
Selecting object	In manual backup, the capacity for cross-region backup is currently only supported for selecting the entire cluster for cross-region backup.
Naming remarks	When a manual backup is created, the settings of the remark name for the backup file is supported. Only digits, English upper case and lower case letters, Chinese as well as special characters -_./()[]()+=:;@ are supported and the length cannot exceed 60.
Cross-region backup	Choose to enable cross-region backup. After that, you can save manual logical backups to other regions.
Backup region	Choose the region where the manual logical backup is stored; Up to three different regions can be selected.

Managing Backup
Viewing Backup List

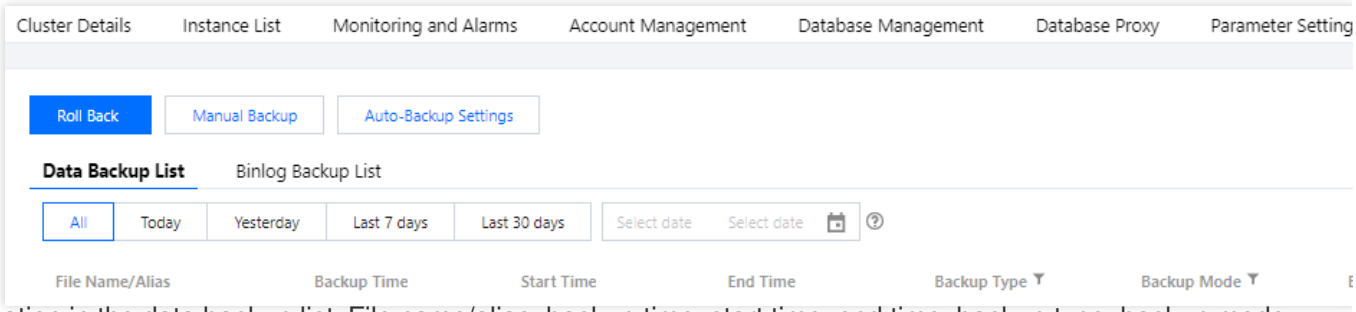
Last updated：2023-11-09 14:52:38

TDSQL-C for MySQL automatically backs up data based on the default backup settings. You can modify the automatic or manual backup settings in the console. You can also view the backup files and relevant information in the backup list.

This document describes how to view the list of backup files in the console.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the ID of the target cluster or **Manage** in the **Operation** column to enter the cluster management page.
3. On the **Backup Management** tab, view backup tasks in the data or binlog backup list.

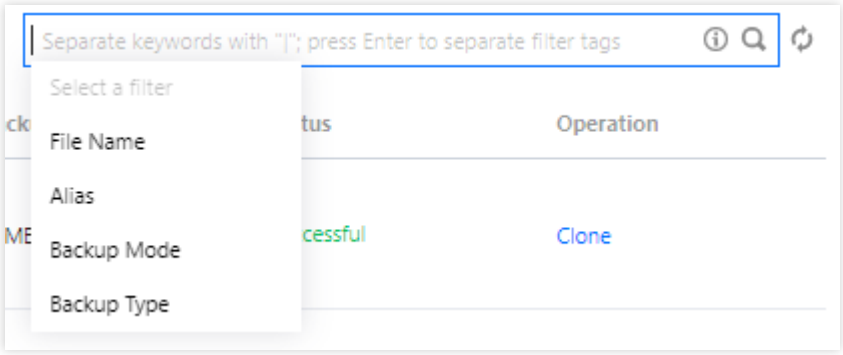


Information in the data backup list: File name/alias, backup time, start time, end time, backup type, backup mode, backup size, status, and operation.

Information in the binlog backup list: File name, binlog backup start time, binlog backup end time, backup size, and operation.

You can filter backup files by time (all, today, yesterday, past 7 or 30 days, or custom time period).

You can search for backup files in the search box on the right by file name, alias, backup mode, and backup type.



FAQs

Can I download or restore backup files that exceed the retention period?

Expired backup files (including data and binlog backup files) will be automatically deleted and cannot be downloaded or restored.

You can download logical backup files in the TDSQL-C for MySQL console.

You can also perform backup manually in the console.

Note:

Manual logical backups can be deleted manually, while snapshot backups cannot.

Can I delete backups manually?

Automatic backups cannot be deleted manually. They will be automatically deleted after the retention period ends, which is 7 days and can be customized as needed. For more information, see Setting Backup Retention Period.

Manual logical backups can be manually deleted from the backup list in the TDSQL-C for MySQL console.

How can I reduce the backup capacity cost?

Delete manual logical backups that are no longer used. For detailed directions, see Deleting Backup.

Reduce the automatic backup retention period as needed to implement the regular deletion of automatic backups. For more information, see Setting Backup Retention Period.

Setting Backup Retention Period

Last updated：2023-08-24 09:49:03

TDSQL-C for MySQL backups include data and binlog backups in automatic and manual modes. Automatic backups cannot be manually deleted. You can set and adjust the backup retention period as needed, after which they will be automatically deleted.

This document describes how to set the backup retention period in the console.

Note:

The binlog backup retention period cannot be shorter than the data backup retention period.

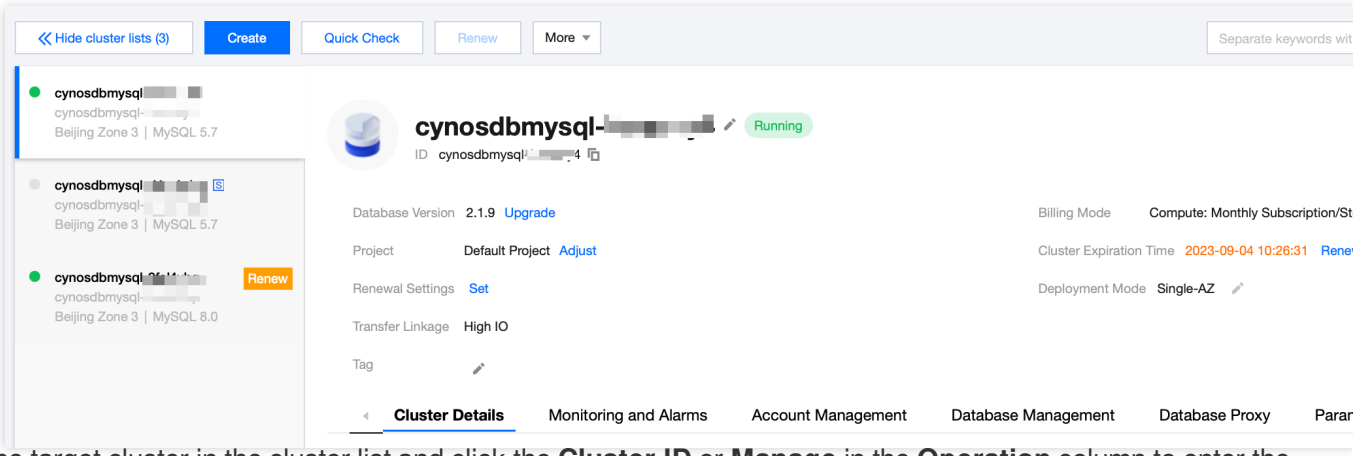
Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode.

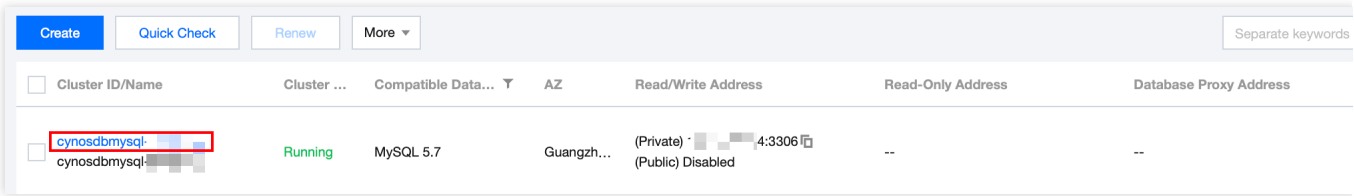
Tab view

List view

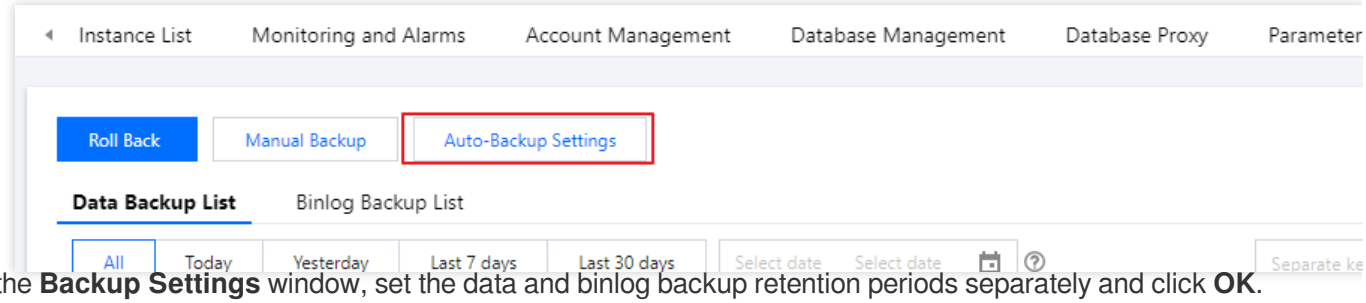
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



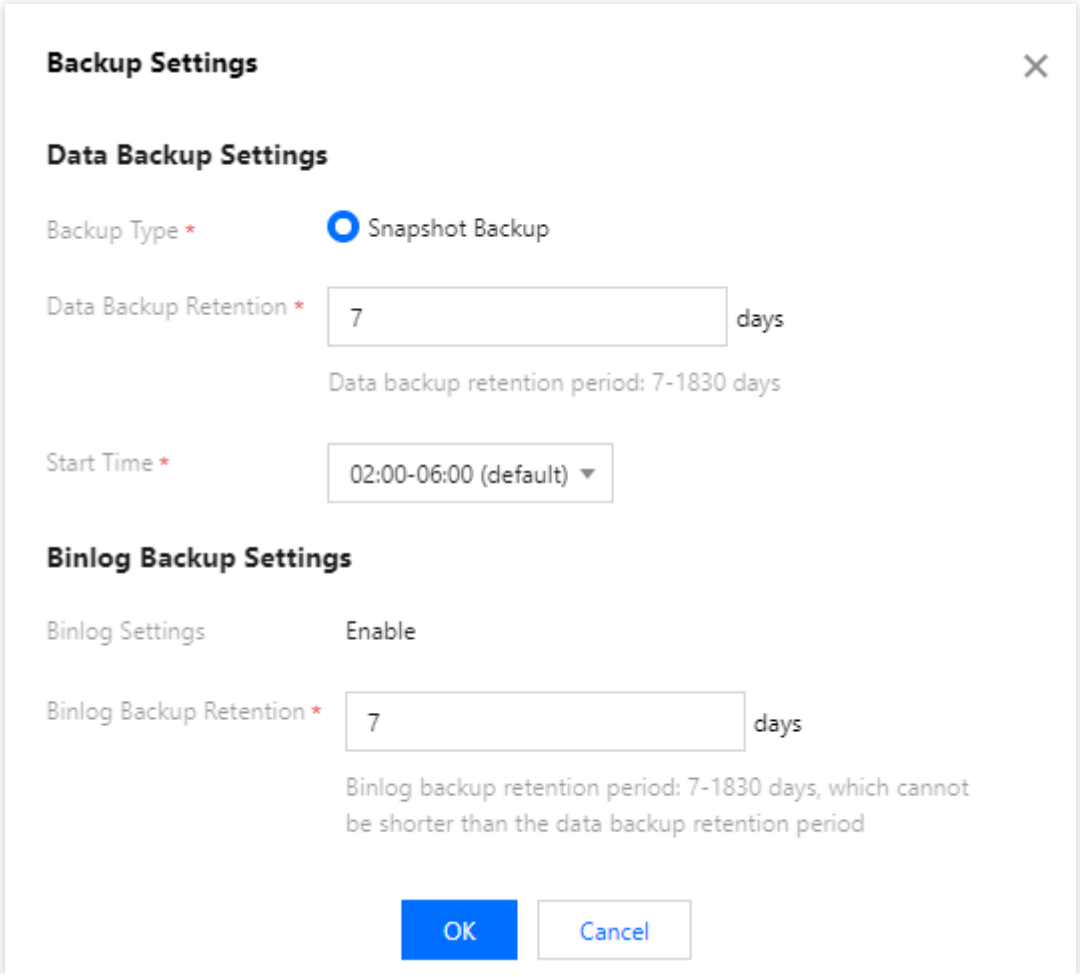
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Backup Management** tab and click **Auto-Backup Settings**.



4. In the **Backup Settings** window, set the data and binlog backup retention periods separately and click **OK**.



Downloading and Decompressing Backup

Last updated：2023-02-08 09:46:36

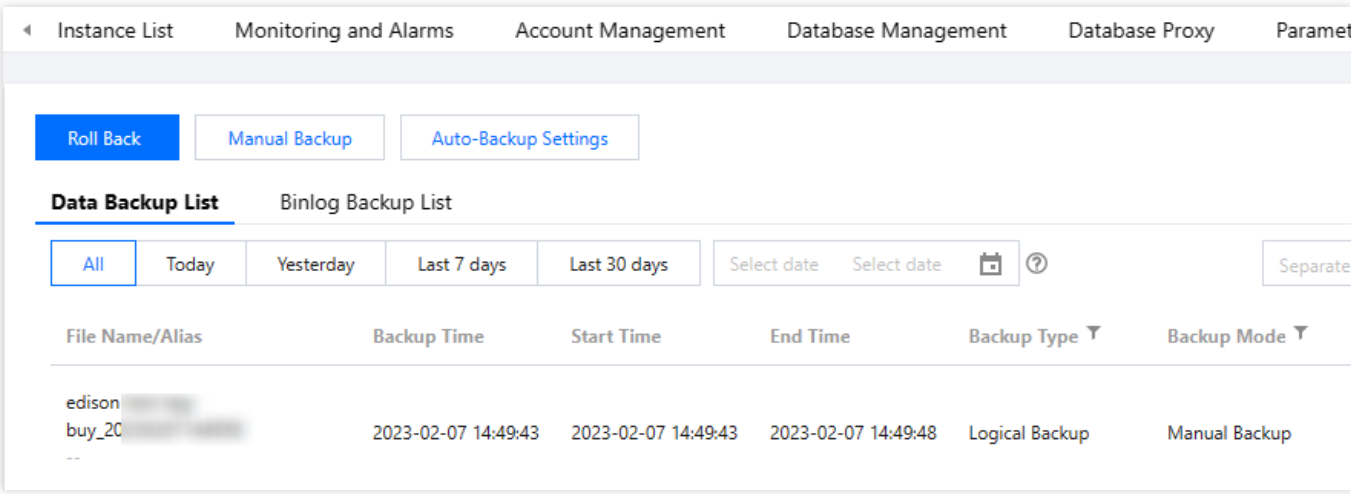
The TDSQL-C for MySQL console provides the list of logical backup files that can be downloaded. They may then be used to restore data from one database to another (such as a self-built database or a database provided by another cloud vendor).

File Types Supported for Download, Decompression, and Deletion

Category	Backup Type		Method	Download	Decompression Required After Download	Deletion
Data backup	Logical backup	Full backup	Manual	✓	✓	✓
Binlog backup	Binlog backup	Incremental backup	Automatic	✓	✓	x

Downloading a Data Backup File

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select **Backup Management** > **Data Backup List** to view the backup file.



4. In the **Operation** column of a data backup file, click **Download**.

5. In the pop-up window, click **Copy** to get the file download address for fast download by running the `wget` command, or directly click **Download**.

Note:

After copying the download address, log in to a Linux CVM instance as instructed in Customizing Linux CVM Configurations and run the `wget` command for download.

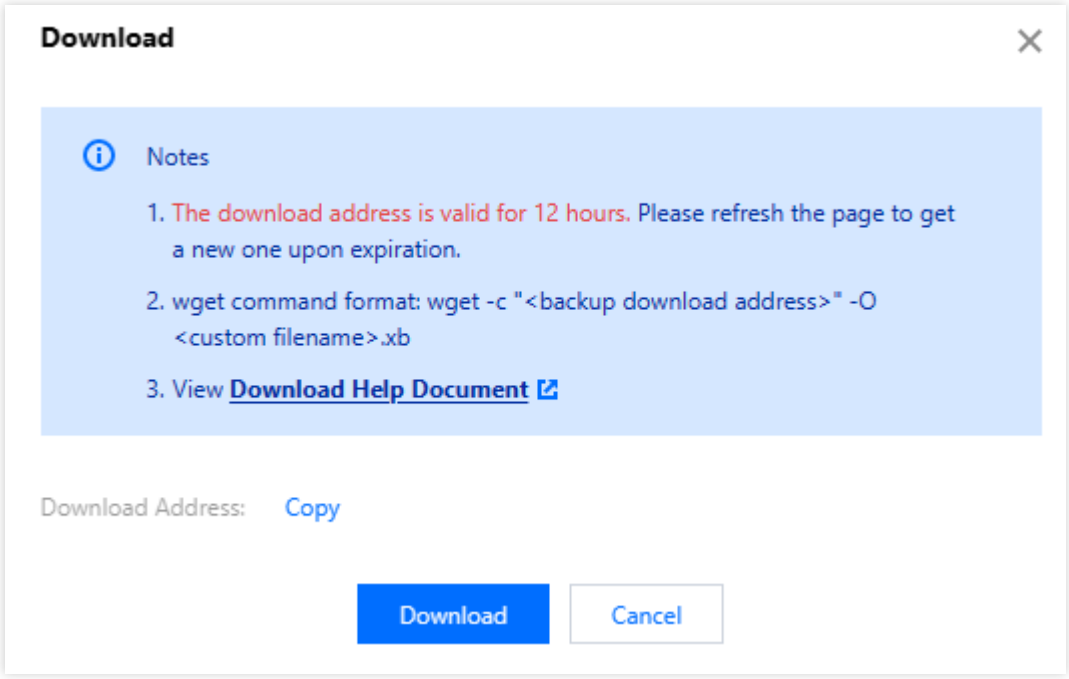
If the cluster and CVM instance are in the same region, the `wget` command can be used for fast download over the private network, no matter whether they are in the same or different VPCs.

If the cluster and CVM instance are in different regions, fast download over the private network is not supported, and the public IP must be enabled for the CVM instance before the `wget` command can be used for download.

The download address is valid for 12 hours, after which you will need to enter the download page again to get a new one.

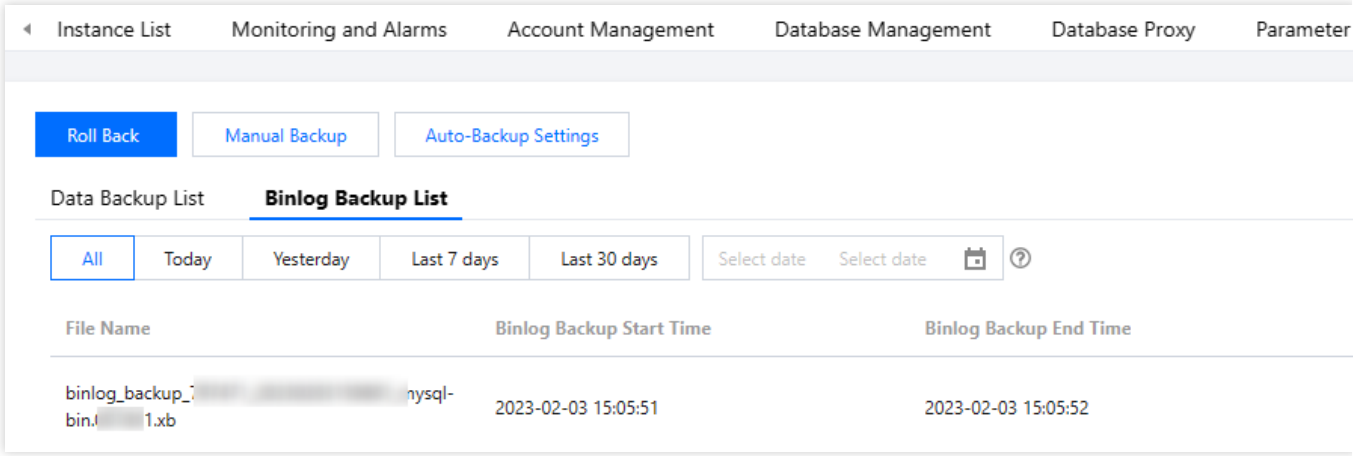
The URL must be enclosed with quotation marks when the `wget` command is used to download.

wget command format: wget -c " " -O .xb



Downloading a Binlog Backup File

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select **Backup Management** > **Binlog Backup List** to view the backup file.

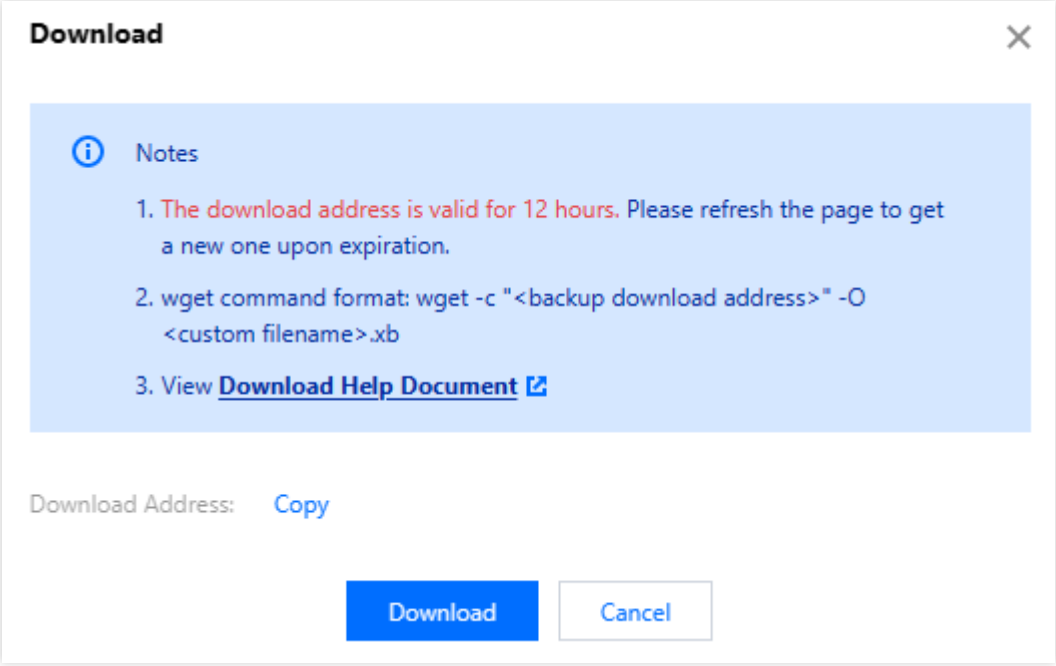


4. In the **Operation** column of a data backup file, click **Download**.
5. In the pop-up window, click **Copy** to get the file download address for fast download by running the `wget` command, or directly click **Download**.

Note:

After copying the download address, log in to a Linux CVM instance as instructed in Customizing Linux CVM Configurations and run the `wget` command for download.

If the cluster and CVM instance are in the same region, the `wget` command can be used for fast download over the private network, no matter whether they are in the same or different VPCs.
If the cluster and CVM instance are in different regions, fast download over the private network is not supported, and the public IP must be enabled for the CVM instance before the `wget` command can be used for download.
The download address is valid for 12 hours, after which you will need to enter the download page again to get a new one.
The URL must be enclosed with quotation marks when the `wget` command is used to download.
wget command format: wget -c " " -O .xb



Decompressing a File

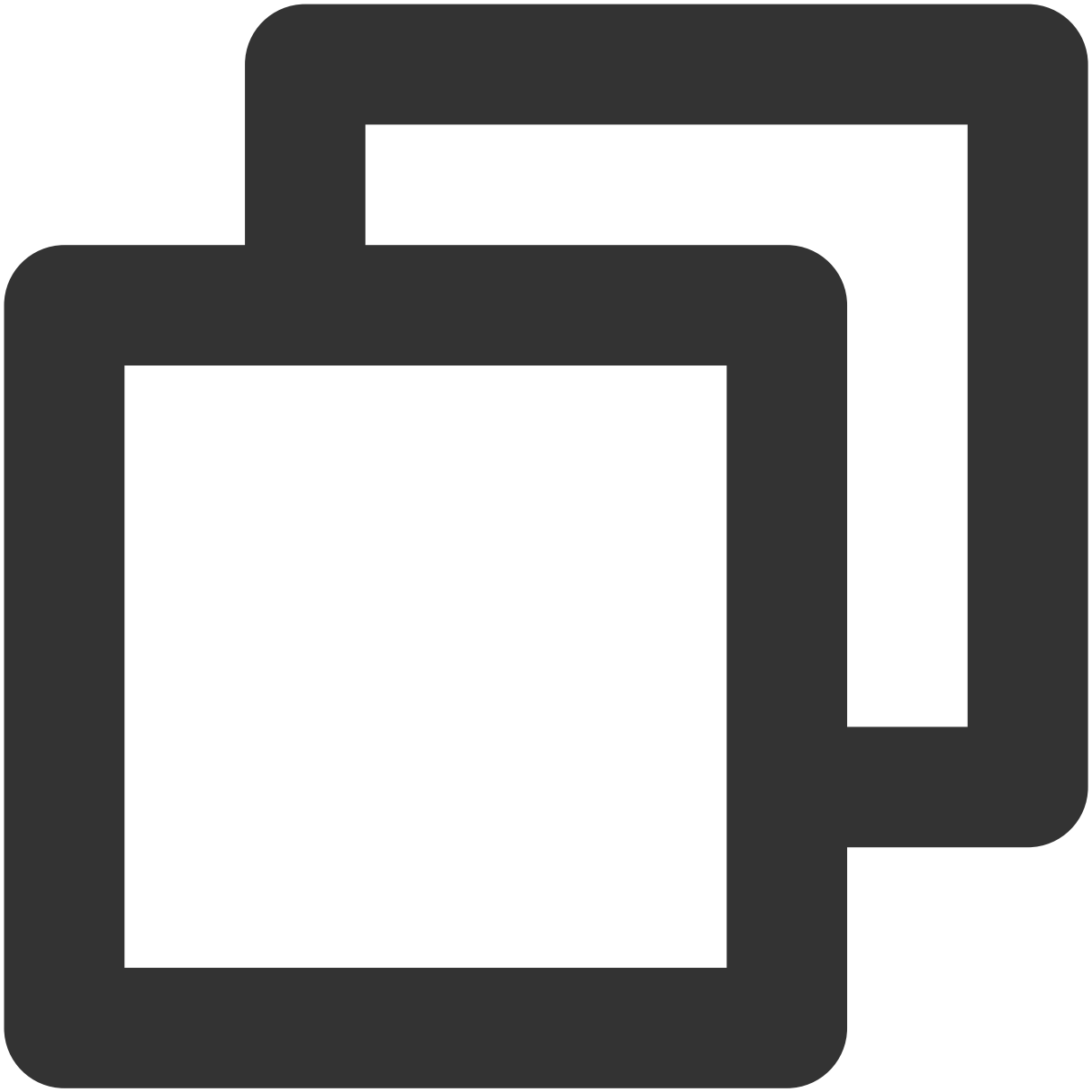
To save storage space, TDSQL-C for MySQL data and binlog backup files will be compressed with qpress and then packed with xbstream offered by Percona. Therefore, downloaded backup files can be imported to the target database only after being unpacked and decompressed.

Unpacking a backup file

Unpack the backup file with xbstream.

Note:

xbstream can be downloaded at Percona's official website. Select Percona XtraBackup v2.4.6 or later. For more information on installation, see Installing Percona XtraBackup on Red Hat Enterprise Linux and CentOS.



```
xbstream -x < test0.xb  -P
```

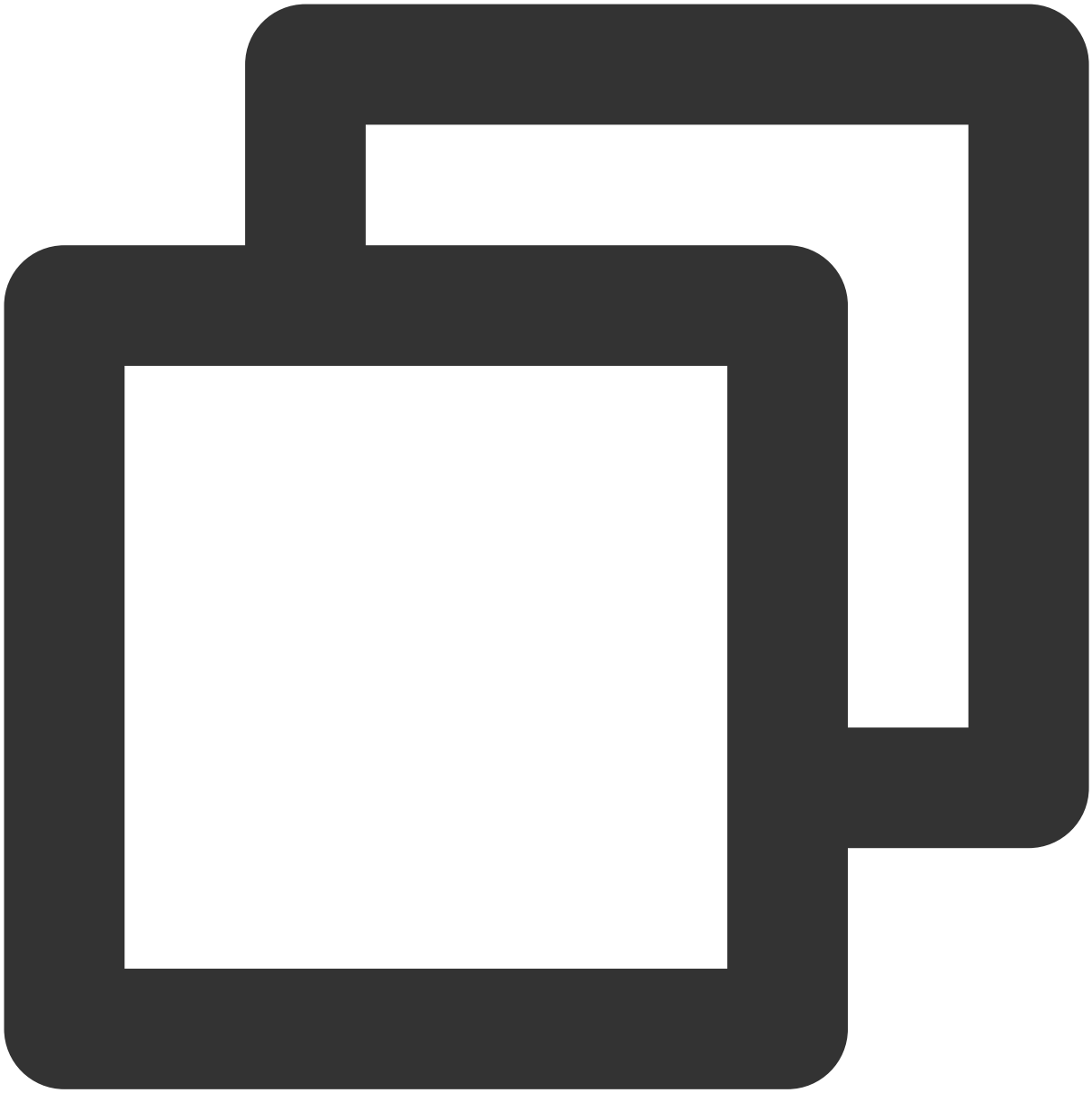
Note:

Replace `test0.xb` with your backup file name.

The unpacking result is as shown below:

Decompressing a backup file

1. Download qpress by running the following command.

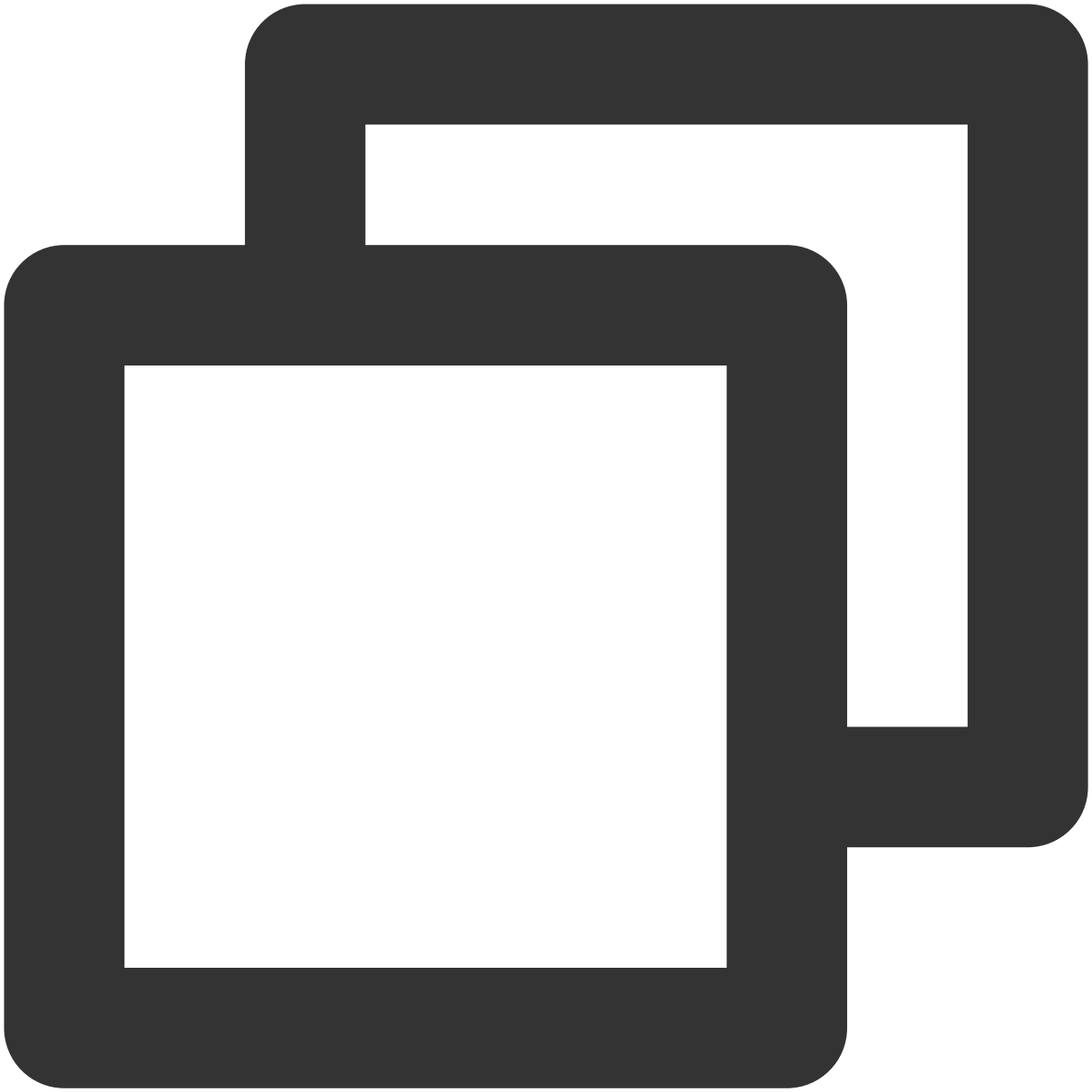


```
wget -d --user-agent="Mozilla/5.0 (Windows NT x.y; rv:10.0) Gecko/20100101 Firefox/
```

Note:

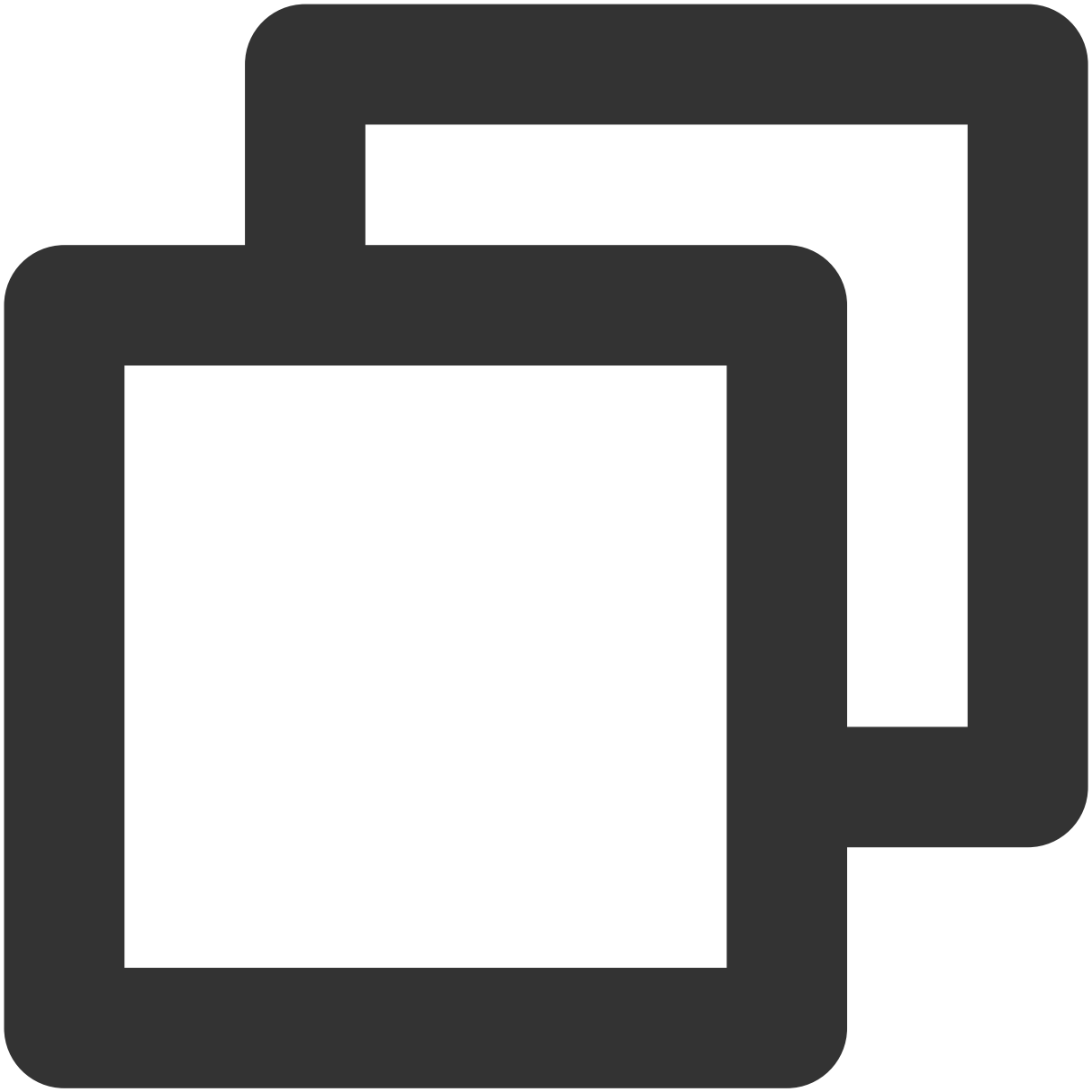
If an error is displayed during the `wget` download, you can click here to download qpress locally and upload it to the Linux CVM instance. For more information, see Uploading Files from Linux or MacOS to Linux CVM via SCP.

2. Extract the qpress binary files by running the following command:



```
tar -xf qpress-11-linux-x64.tar -C /usr/local/bin
source /etc/profile
```

3. Decompress the backup file with qpress.



```
qpress -d <backup file>.sql.qp .
```

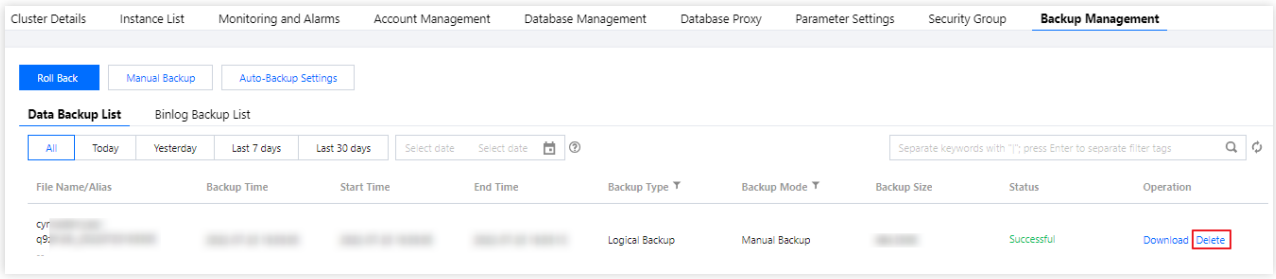
Deleting Backup

Last updated：2023-05-31 16:10:39

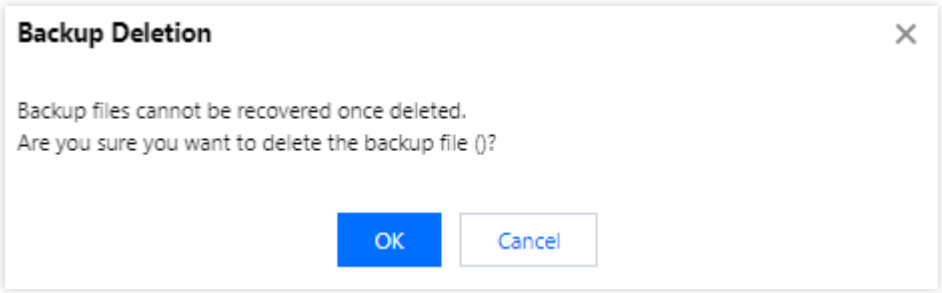
TDSQL-C for MySQL allows you to delete backups to save space. You can manually delete manual backups. For automatic backups, you can adjust their lifecycle as instructed in Setting Backup Retention Period.
This document describes how to delete manual backups in the console.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select the **Backup Management** tab, select the data or log backup list, find the target backup file, and click **Delete** in the **Operation** column.



4. In the pop-up window, click **OK**.



Note:

Note that backup files cannot be restored once deleted.

Rolling back
Restoring Data from Logical Backup
Restoring Data from Logical Backup

Last updated：2024-03-06 16:15:13

This document describes how to restore data from a logical backup file.

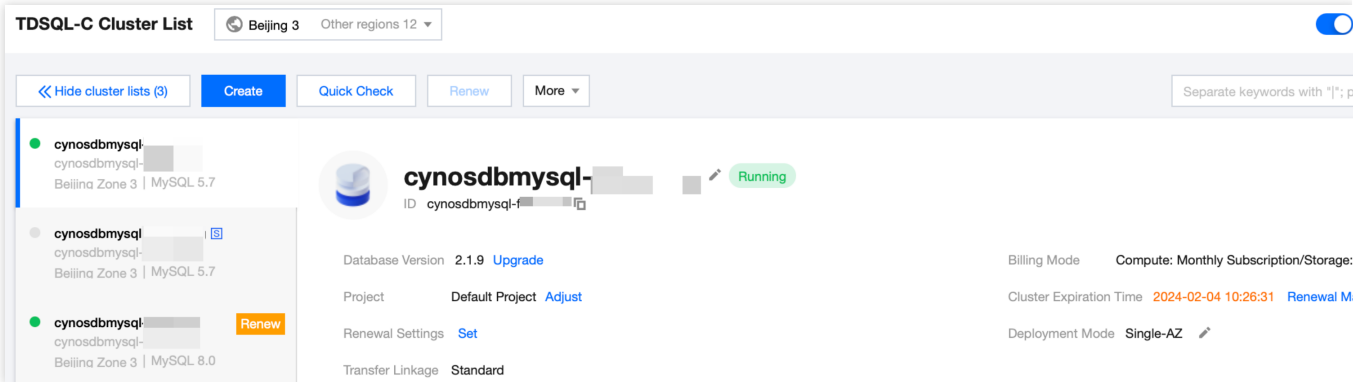
Overview

Note:
To save storage space, TDSQL-C for MySQL backup files will be compressed with qpress and then packed with xbstream offered by Percona.
TDSQL-C MySQL Edition supports logical backup methods, allowing users to manually generate logical backup files through the console. It also supports the download and acquisition of logical backup files for the entire cluster or partial databases. This document introduces data recovery using logical backup files on the Linux platform.

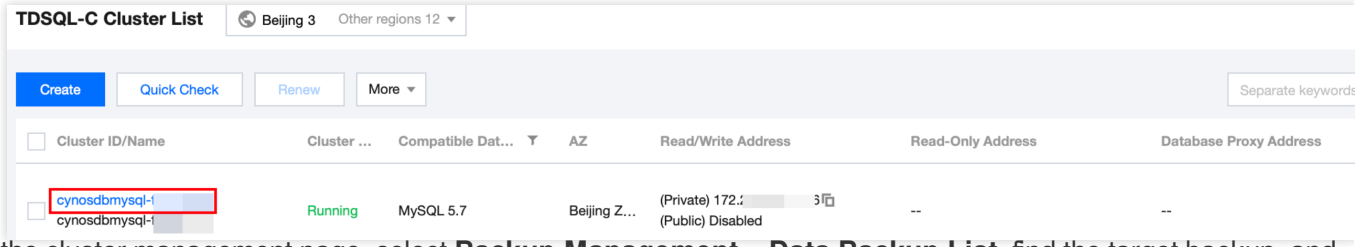
Directions

Step 1. Downloading the Backup File

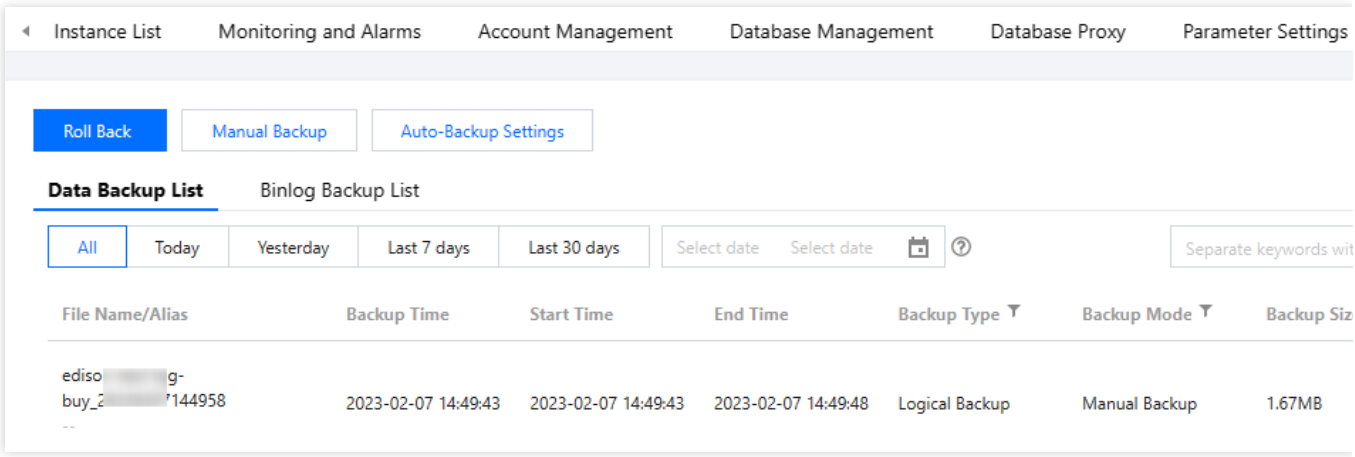
1. Log in to the TDSQL-C for MySQL console, and operate according to the actual console view mode.
Tabbed View
List View
In the cluster list on the left, click **Target Cluster** to enter the cluster management page.



Locate the target cluster in the list on the left, and click the Cluster ID or **Manage** in the **Operation** column to access the cluster management page.



2. On the cluster management page, select **Backup Management** > **Data Backup List**, find the target backup, and click **Download** in the **Operation** column.



3. Copy the download address in the pop-up window, log in to a Linux CVM instance as instructed in Customizing Linux CVM Configurations, and run the `wget` command for fast download.

Note:

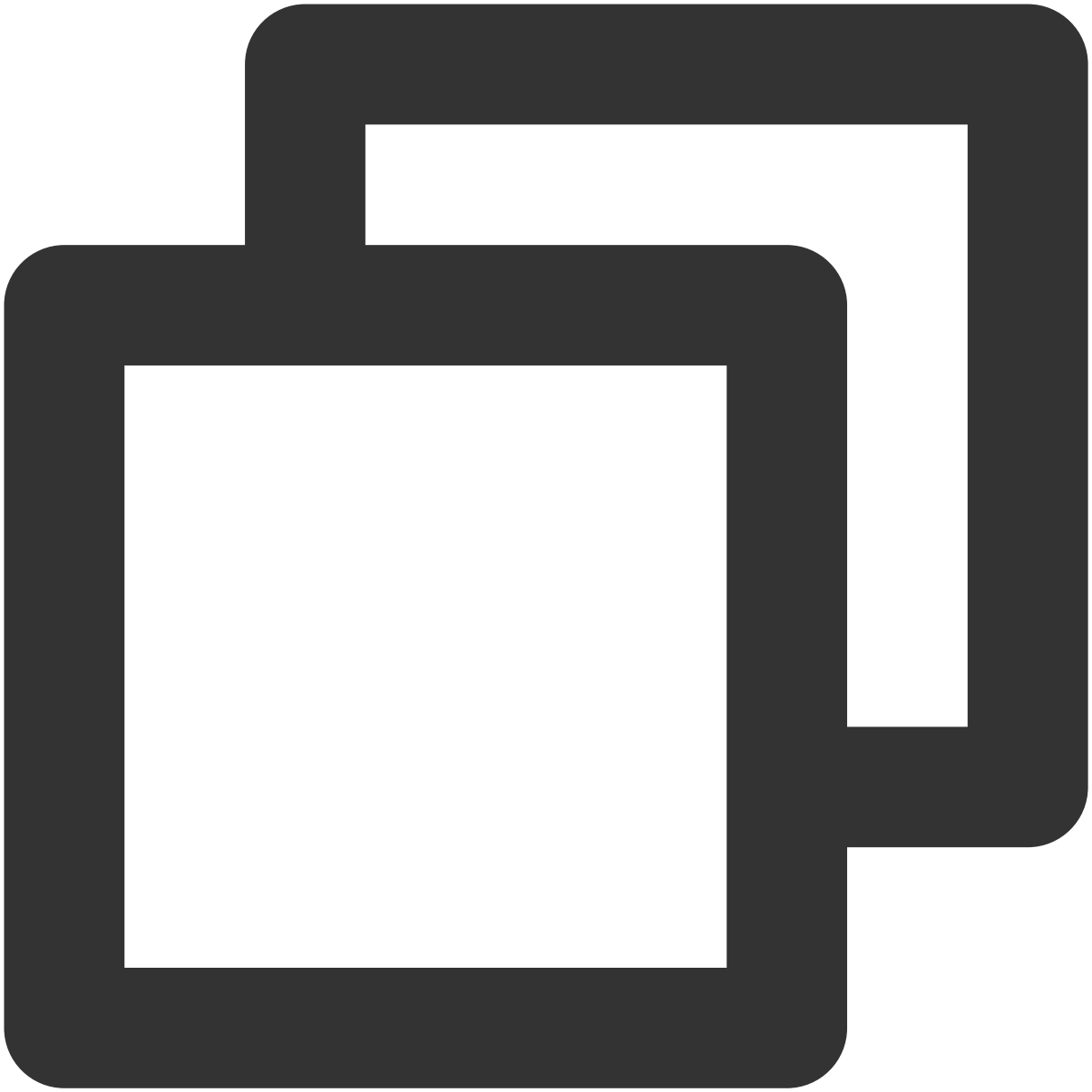
After copying the download address, log in to a Linux CVM instance as instructed in Customizing Linux CVM Configurations and run the `wget` command for download.

If the cluster and CVM instance are in the same region, the `wget` command can be used for fast download over the private network, no matter whether they are in the same or different VPCs.

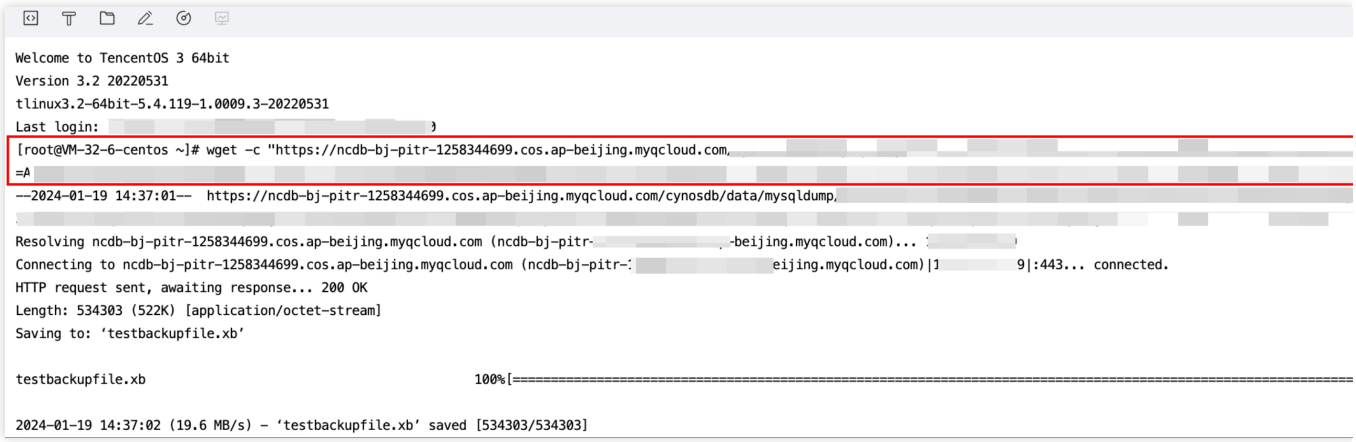
If the cluster and CVM instance are in different regions, fast download over the private network is not supported, and the public IP must be enabled for the CVM instance before the `wget` command can be used for download.

You can also click **Download** to download it directly. However, this may take a longer time.

The `wget` command format is as follows:



```
wget -c "<backup download address>" -O <custom filename>.xb
```

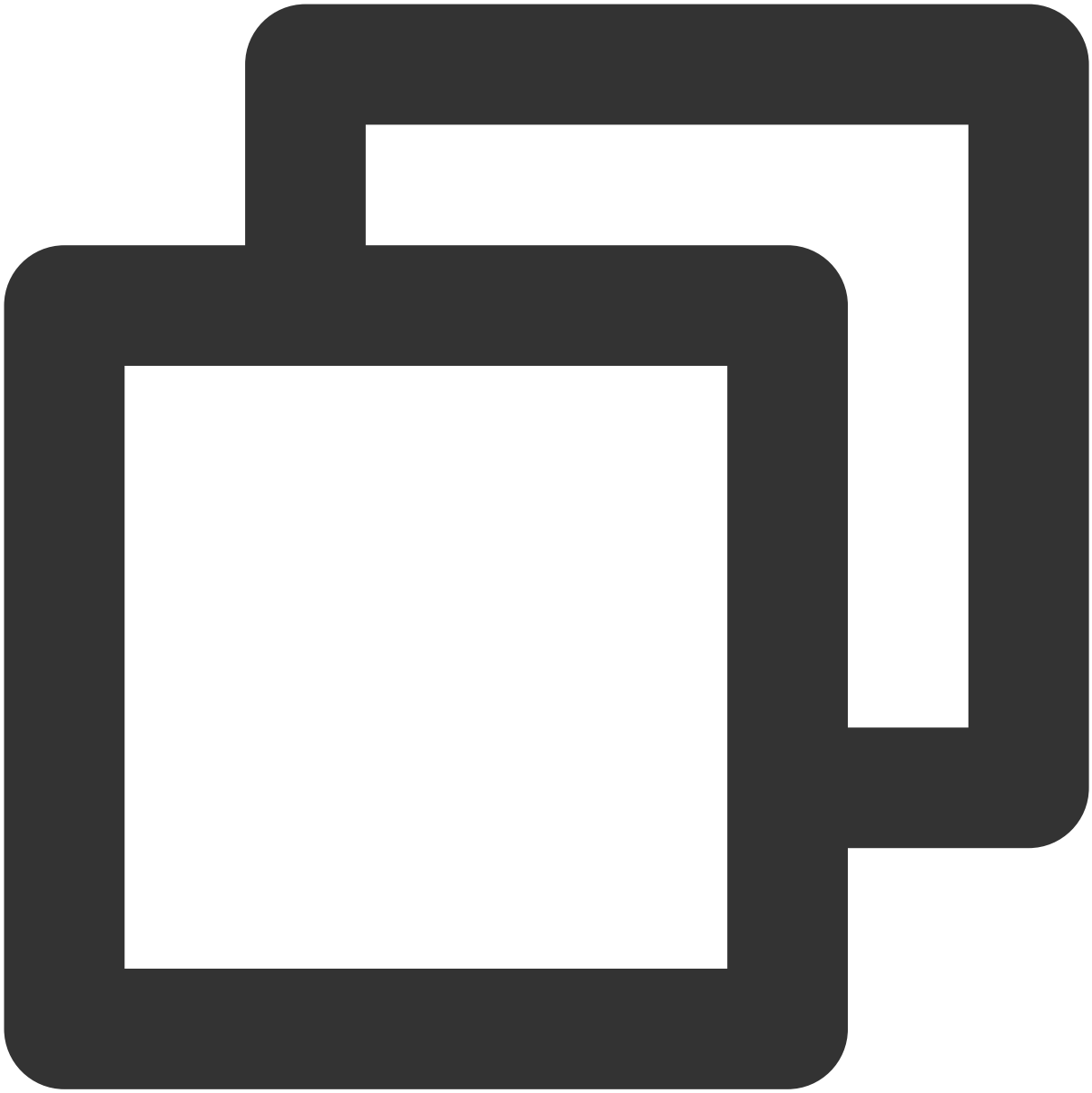


Step 2. Unpacking the Backup File

Unpack the backup file with xbstream.

Note:

xbstream can be downloaded at Percona's official website. Select Percona XtraBackup v2.4.6 or later. For more information on installation, see Installing Percona XtraBackup on Red Hat Enterprise Linux and CentOS.



```
xbstream -x < test0.xb  -P
```

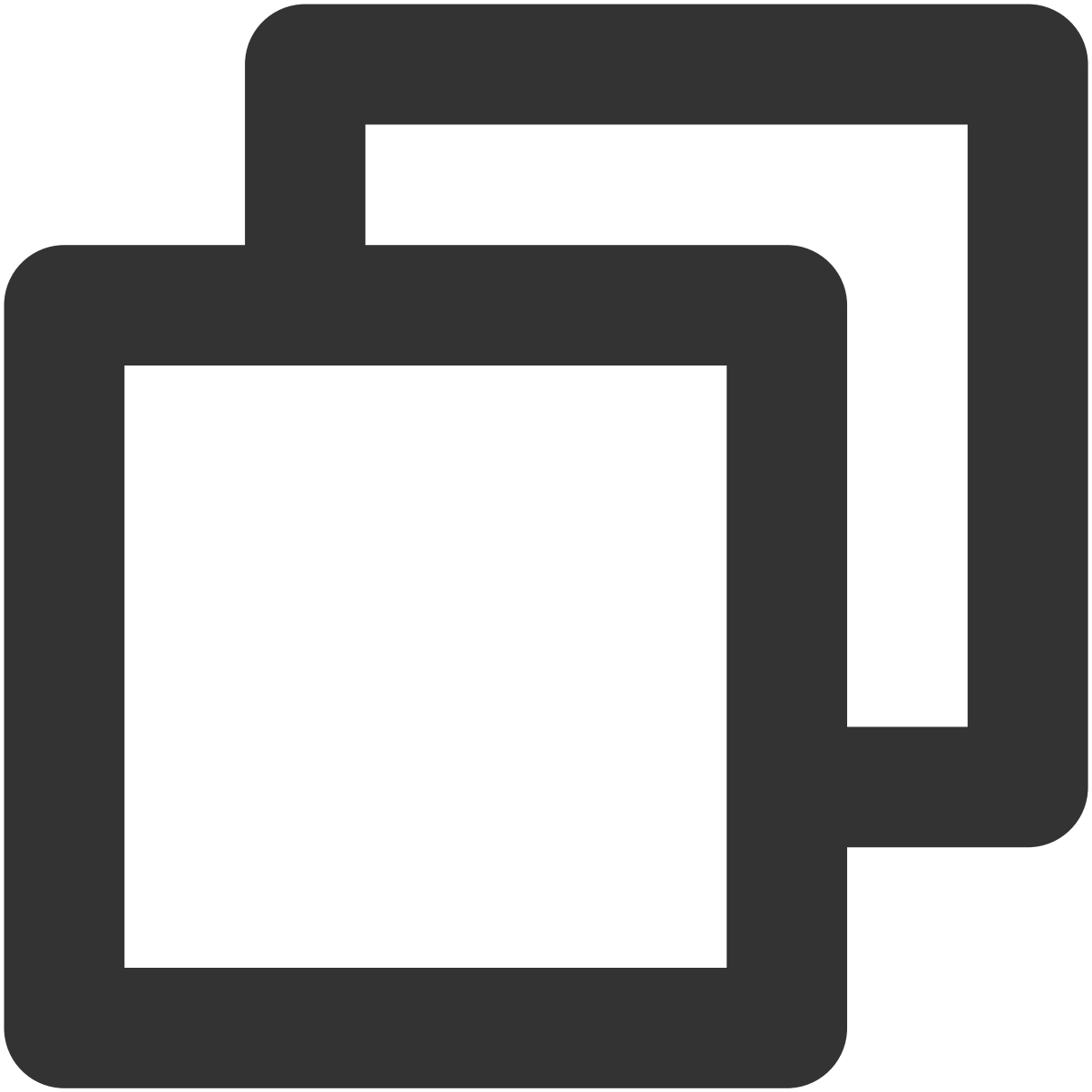
Note:

Replace `test0.xb` with your backup file name.

The unpacking result is as shown below:

Step 3. Decompressing the Backup File

1. Download qpress by running the following command.

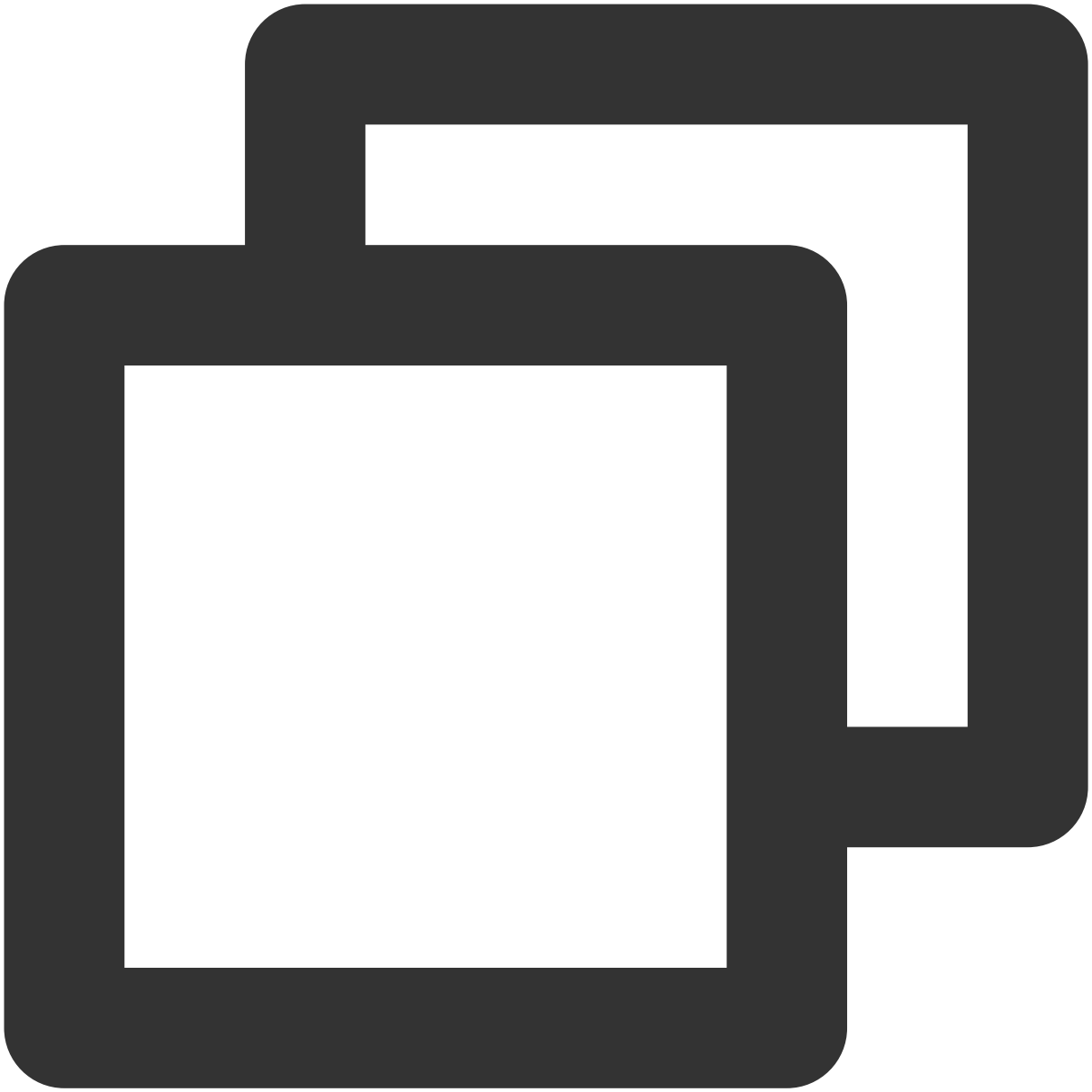


```
wget -d --user-agent="Mozilla/5.0 (Windows NT x.y; rv:10.0) Gecko/20100101 Firefox/
```

Note:

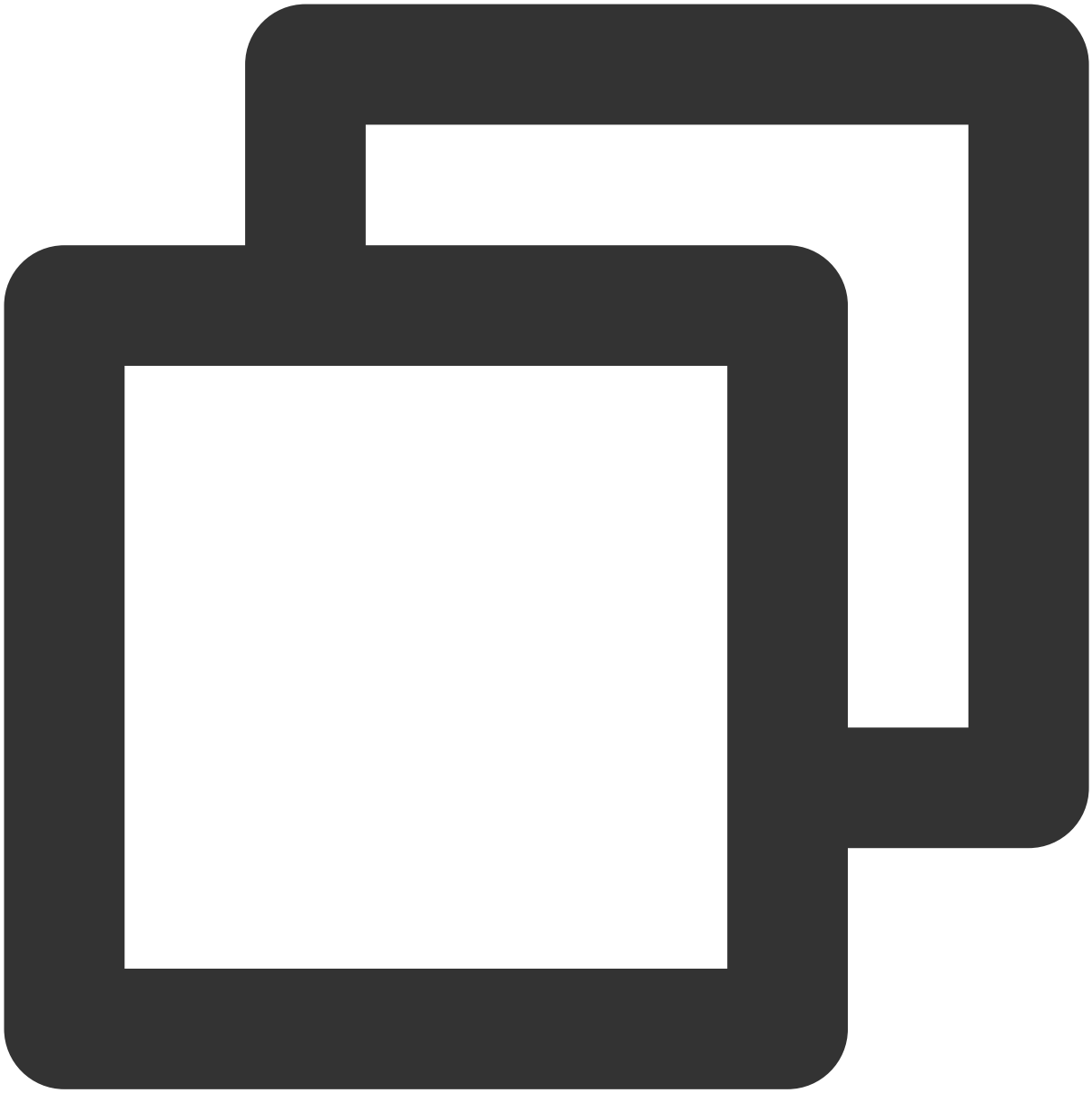
If an error is displayed during the `wget` download, you can click here to download qpress locally and upload it to the Linux CVM instance. For more information, see Uploading Files from Linux or MacOS to Linux CVM via SCP.

2. Extract the qpress binary files by running the following command:



```
tar -xf qpress-11-linux-x64.tar -C /usr/local/bin
source /etc/profile
```

3. Decompress the backup file with qpress.

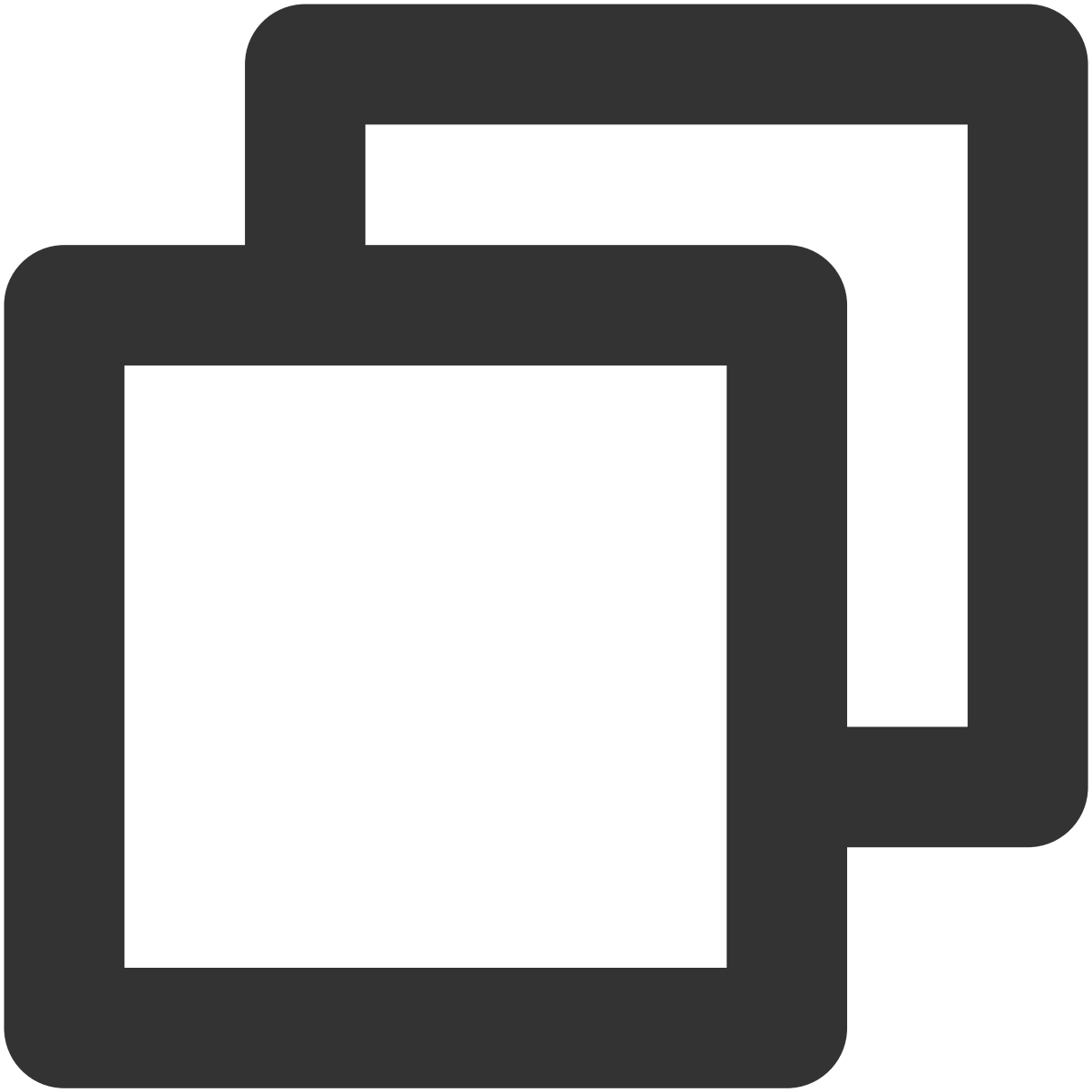


```
qpress -d <backup file> <unzipped file>
```

<Backup file>: Replace with the corresponding unpacked backup filename. The angle brackets <> in the command should be removed, retaining only the actual parameters.
<Unzipped file>: Specify the custom filename after decompression, which should immediately follow the backup file name in the command. The angle brackets <> in the command should be removed.

Step 4. Importing the Backup File into the Target Database

Import the .sql file into the target database by running the following command:



```
mysql -u<account name> -P<port> -h<target database's private network address> -p <
```

Restoring Data from Snapshot Backup
Restoring Data to Original Cluster

Last updated：2023-03-01 14:33:46

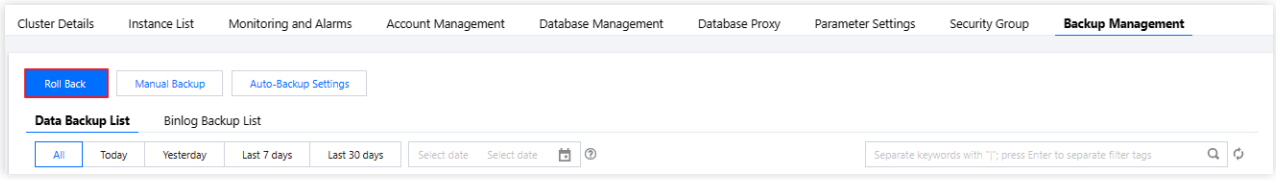
TDSQL-C for MySQL can roll back databases/tables to an original cluster and roll back an entire cluster (clone) to a new cluster as instructed in Restoring Data to New Cluster. You can choose different rollback methods based on your business needs. This document describes how to roll back certain databases/tables to the original cluster.

Rollback methods

Rollback by backup file: This method restores the cluster to the dataset state of a backup file. The selection range of the backup file is determined by the data backup retention period you set.
Rollback by time point: This method restores the cluster to any time point within the log backup retention period you set.

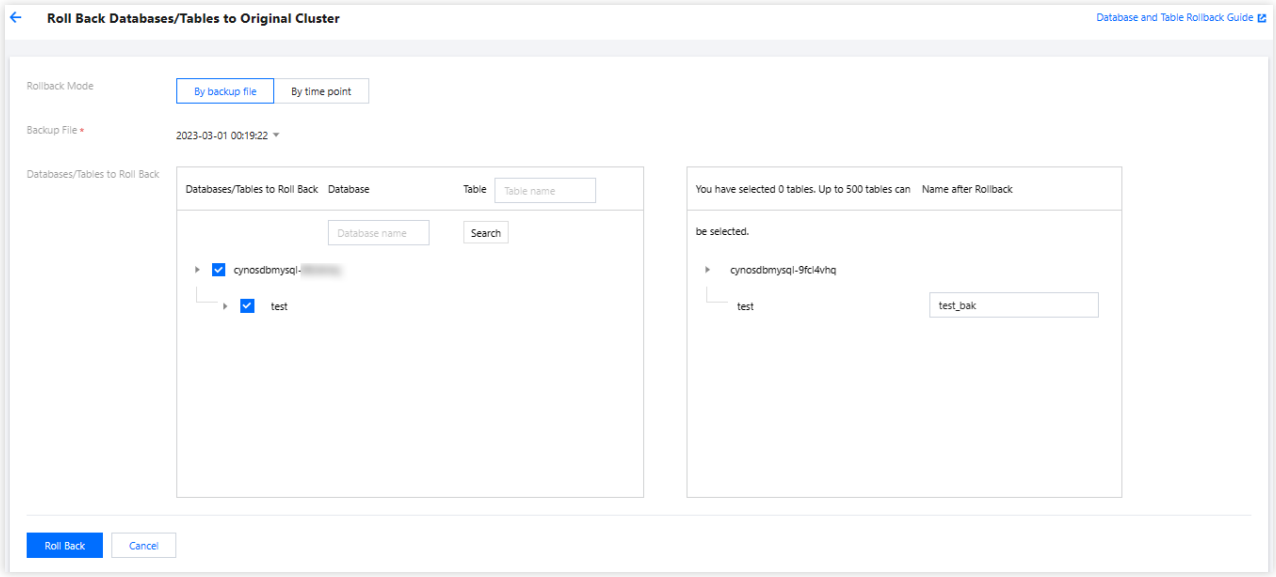
Directions

1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list to enter the cluster management page.
2. On the cluster management page, select the **Backup Management** tab and click **Roll Back**.



3. On the database/table rollback page, select the database/table to be rolled back, rename it, and click **Roll Back**.
For database/table rollback, you need to specify the database/table to be rolled back. If you are not sure about that, we recommend you clone the original cluster first and then migrate back to it later when you can specify the rollback objects.
If the database/table to be rolled back does not exist at the specified time point, the rollback will fail.
If the database/table to be rolled back does not exist or has been dropped, you need to log in to the database and create a database/table first before performing rollback in the console.

If there are primary or foreign key constraints in the specified database/table to be rolled back, make sure that the associated databases/tables exist during the rollback process; otherwise, the rollback will fail.
Up to 500 databases or tables can be rolled back at a time.



4. In the pop-up window, confirm that everything is correct and click **OK** to initiate the rollback task.
5. After the task is submitted, you can click **View Rollback Task** or go to the task list to view the rollback progress and task details.
6. After the rollback is completed, you can see the new restored database/table in the original cluster and perform further operations.

Restoring Data to New Cluster

Last updated：2024-03-26 17:00:10

TDSQL-C for MySQL can roll back databases/tables to the original cluster and roll back an entire cluster (clone) to a new cluster. You can choose different rollback methods based on your business needs. This document describes how to clone a cluster and quickly restore data to the newly purchase cluster in the console.

TDSQL-C for MySQL provides the clone feature to restore a cluster to any time point in the log backup retention period or to the backup set of the specified backup file.

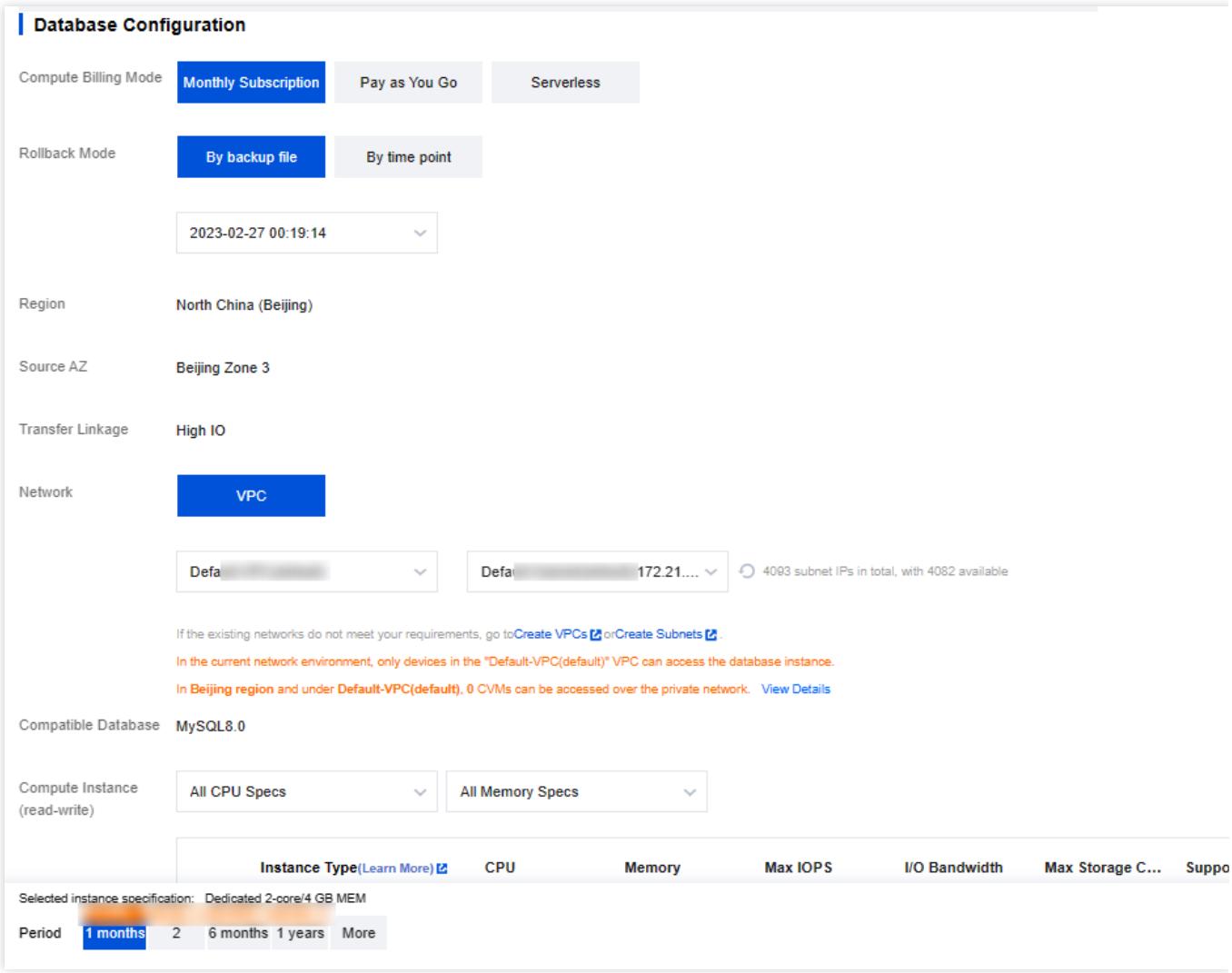
The clone will create a new cluster based on your choice. After the new cluster is verified, you can either migrate the data back to the source cluster through DTS or use the new cloned cluster directly. For more information, see Migrating to TDSQL-C for MySQL.

Note:

Billing will start for the cloned new cluster after the clone succeeds. For more information on billing and pricing, see Billing Overview.

Directions

1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list to enter the cluster management page.
2. On the cluster management page, select the **Backup Management** tab and click **Clone** in the **Operation** column in the backup list.



3. On the purchase page, complete the **Database Configuration** settings based on he selected database mode, and click **Next**.

For Provisioned Resource instance mode

Instance Mode: Provisioned Resources

Rollback Mode: Clone by backup file and clone by time point are supported.

By backup file: A new cluster can be created from the specified backup set for restoration, and the selection range is based on the time when the backup is completed and the retention period.

By time point: A new cluster can be created from the specific time point for restoration, and the time point selection range is based on the backup retention period.

Region: Same as that of the cloned cluster.

AZ: Same as that of the cloned cluster.

Transfer Linkage: High IO.

Network: Select VPC.

Database Version: Same as that of the cloned cluster.

Compute Instance (read/write)Select the compute specifications of the read-write instance. Only one read-write instance can be configured.
Compute Instance (read-only): Select the compute specifications of the read-only instance. You need to select at least two read-only instances to ensure that the cluster is highly available. After the cluster is created, you can expand the read capability of the cluster by adding read-only instances.
Compute Billing Mode: The billing mode is pay-as-you-go by default. Currently, clone doesn't support monthly subscription.
Storage Billing Mode: The billing mode is pay-as-you-go by default.

For serverless instance mode

Instance Mode: Serverless
Rollback Mode: Clone by backup file and clone by time point are supported.
By backup file: A new cluster can be created from the specified backup set for restoration, and the selection range is based on the time when the backup is completed and the retention period.
By time point: A new cluster can be created from the specific time point for restoration, and the time point selection range is based on the backup retention period.
Region: Same as that of the cloned cluster.
AZ: Same as that of the cloned cluster.
Transfer Linkage: High IO.
Network: Select VPC.
Database Version: Same as that of the cloned cluster.
Compute Unit: Select the upper and lower limits of the TDSQL-C compute unit (CCU), and the instance will be automatically and elastically scaled within the selected resource range. CCU is the computing and billing unit for the serverless mode. A CCU is approximately equal to 1 CPU core and 2 GB memory. The number of CCUs used in each billing cycle is the greater of the number of CPU cores used by the database and 1/2 of the memory size. For more information, see Compute Unit.
Auto-Pause: Configure the automatic pause time of the instance. If there is no connection to access the database within the set time, the instance will be automatically paused, with billing stopped.
Billing Mode: Resource pack (monthly subscription) and pay-as-you-go billing are supported.

Note:

The compute resource pack will be used preferably for the deduction of actual usage of pay-as-you-go products. When the pack is used up, the resource usage will be pay-as-you-go. Compute resource packs are deducted based on the actual CCU used per second. The resource pack mode is more cost-effective and flexible than the pay-as-you-go option.
Compute resource pack (displayed when selecting **Resource Pack** in the compute billing mode): You can bind all valid compute resource packs available in the selected region under the current account. If no resource pack is available, you can purchase a resource package first.
Storage Billing Mode: Resource pack (monthly subscription) and pay-as-you-go billing are supported.

Note:

The storage resource pack will be used preferably for the deduction of actual usage of pay-as-you-go products. When the pack is used up, the resource usage will be pay-as-you-go. Storage resource packs are deducted based on the actual storage used per hour. The resource pack mode is more cost-effective and flexible than the pay-as-you-go option.

Storage resource pack (displayed when selecting **Resource Pack** in the storage billing mode): You can bind all valid storage resource packs available in the selected region under the current account. If no resource pack is available, you can purchase a resource package first.

4. Configure the **Basic Info** and **Advanced Configuration** settings on the page redirected to, such as cluster name, default character set, security group, parameter template, and project. Then, click **Buy Now**.

5. Return to the cluster list, and you can see the new cluster created through clone. After its status changes from **Cloning** to **Running**, it can be used normally.

Note:

After confirming that the data in the new cluster created through clone is correct, you can modify the cluster VIP or click **Delete** in the **Operation** column to delete the old cluster on the cluster details page as needed.

Operation Log
Slow Log Overview

Last updated：2022-08-02 17:11:20

Overview

Slow logs are also called slow SQL queries. They are a type of logs provided by TDSQL-C for MySQL to record SQL statements with a longer response time than the threshold in the database process. Specifically, SQL statements with a longer execution time than the `long_query_time` value are recorded in slow logs.

Note:
The slow log feature is currently in beta test and only available in Beijing region. It will be made available in more regions.

Parameters

The `long_query_time` value of TDSQL-C for MySQL defaults to 10, indicating that SQL statements running more than 10 seconds will be recorded in slow logs. You can adjust the value in the range of 0.000000 to 3,600.000000. For more information, see Setting Instance Parameters.

Purpose

After setting this parameter in TDSQL-C for MySQL, you can identify and optimize inefficient SQL statements. Simply put, slow logs are used to troubleshoot SQL statement issues and check the current performance.

Relevant operations

Operation	Use Case	Method
Prevent slow logs from being generated	Get suggestions from DBbrain for slow SQL statement optimization.	-
Set slow	Set the value of `long_query_time` to record slow logs.	Setting

log parameters		Instance Parameters
View slow log monitoring alarms	View slow instance queries through the instance monitoring metric (Slow Queries) and set alarms to receive notifications.	Alarm Policies (Cloud Monitor)
Query and download slow log details	On the **Operation Log** page, query the following slow log information: execution time, SQL statement, client address, username, database, execution duration (s), lock duration (s), parsed rows, and returned rows. You can download them in CSV or native formats.	-
Analyze and optimize slow logs	View the complete SQL template, SQL sample, and optimization suggestion and description through DBbrain to analyze and optimize SQL statements.	Slow SQL Analysis

Querying and Downloading Slow Log Details

Last updated：2022-09-20 00:03:27

The slow log is used to record query statements that take more time than the specified value to execute in TDSQL-C for MySQL read-write and read-only instances. You can find out inefficient query statements to optimize by slow log details. TDSQL-C for MySQL allows you to download such details for easier analysis and optimization.
This document describes how to query and download slow log details.

Querying a slow log

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select the **Instance List** tab and click the ID of the target read-write or read-only instance to enter the instance details page.
4. On the instance details page, select **Operation Log** > **Slow Log Details**.
You can filter slow log details by time (all, today, yesterday, last 7 or 30 days, or custom time period).
You can query slow log details by field (client address, username, or database name) and export them into a list file.
You can query and sort the following details: execution time, SQL statement, client address, username, database, execution duration (s), lock duration (s), parsed rows, and returned rows.

Downloading slow log details

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select the **Instance List** tab and click the ID of the target read-write or read-only instance to enter the instance details page.
4. On the instance details page, select **Operation Log** > **Slow Log Details**.
5. Query the target slow log details by time or keyword and click **Export**.
6. In the pop-up window, select the file format and click **OK** for download.
You can export the filtered or retrieved results in CSV or native format (supported by open-source analysis tools).
CSV format: You can perform quick check and optimization.
Native format: The exported file can be recognized by open-source analysis tools.

You can export up to 2,000 records at a time. To download more than 2,000 records, select a shorter time range and export a part of the records you need. Repeat the steps until all of the records are exported.

CLS

Last updated：2024-03-07 09:38:49

TencentDB for CynosDB provides the Cloud Log Service (CLS) feature to gather slow log data stemming from TencentDB for CynosDB instances. This data is subsequently shipped to CLS for thorough analysis, which encompasses everything from log gathering and storage to log search, facilitating rapid monitoring and problem identification in the business operations. This article outlines the process of enabling or disabling the CLS feature through the console.

Note:

This service is in the process of a phased rollout. If the region of your instance currently does not support CLS, please submit a ticket to provide feedback.

Prerequisites

Before using this feature, please ensure that you have enabled CLS.

Definition of Slow Log

Field Value	Type	Description
TIMESTAMP	-	Reserved field of CLS, signifying the generation time of the log.
instance_id	String	Database instance ID, such as cynosdbmysql-ins-xxxx.
db_name	String	Database name.
rows_examined	Long	Number of scanned rows.
rows_sent	Long	Number of returned rows.
query_time	Double	Execution time (in seconds).
lock_time	Double	Lock waiting time (in seconds).
user_host	String	Client information.
user_name	String	The username employed in client connection to the database instance.

query_sql	String	Slow log SQL.

Enabling CLS

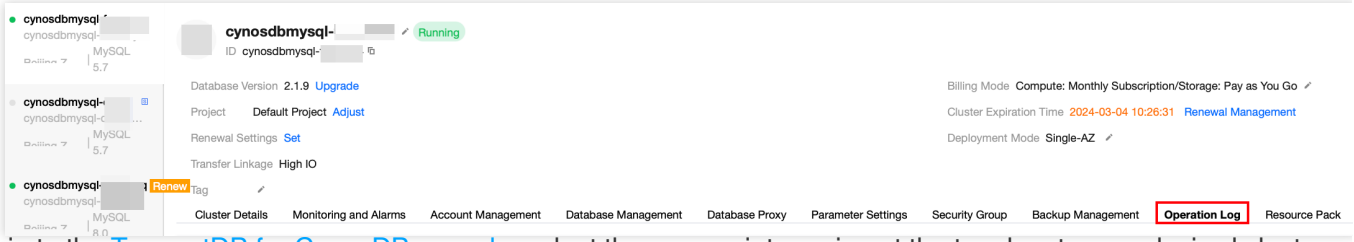
1. In the console, proceed to the cluster operation log page according to the actually used view pattern.

Tab View

List View

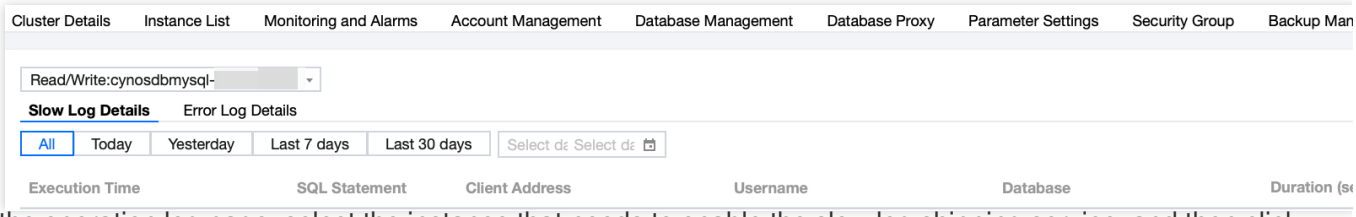
1. Log in to the TencentDB for CynosDB console, locate the cluster list on the left, and click the target cluster to enter the cluster management page.

2. On the cluster management page, select the **Operation Log** page.

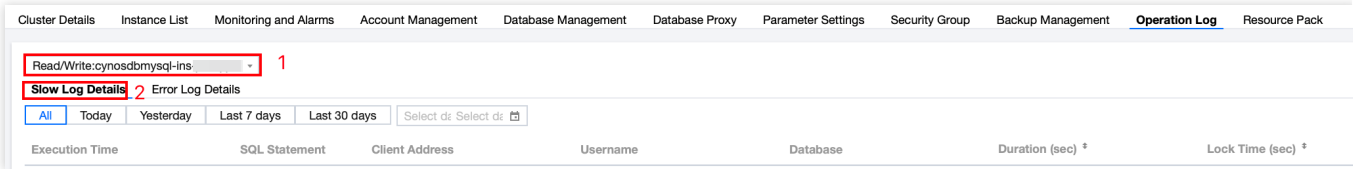


1. Log in to the TencentDB for CynosDB console, select the appropriate region at the top, locate your desired cluster, and click the Cluster ID or **Manage** in the **Operation** column. This will lead you to the cluster management page.

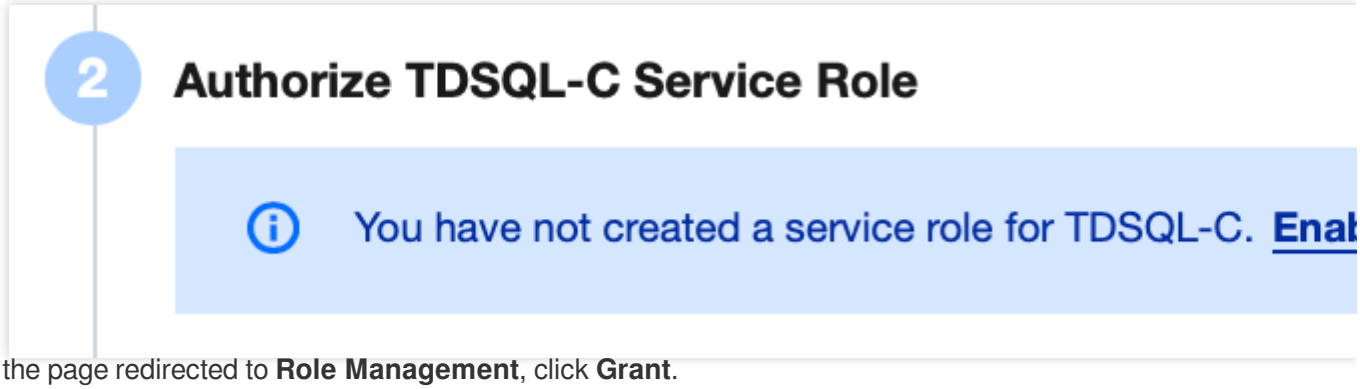
2. On the cluster management page, select the **Operation Log** page.



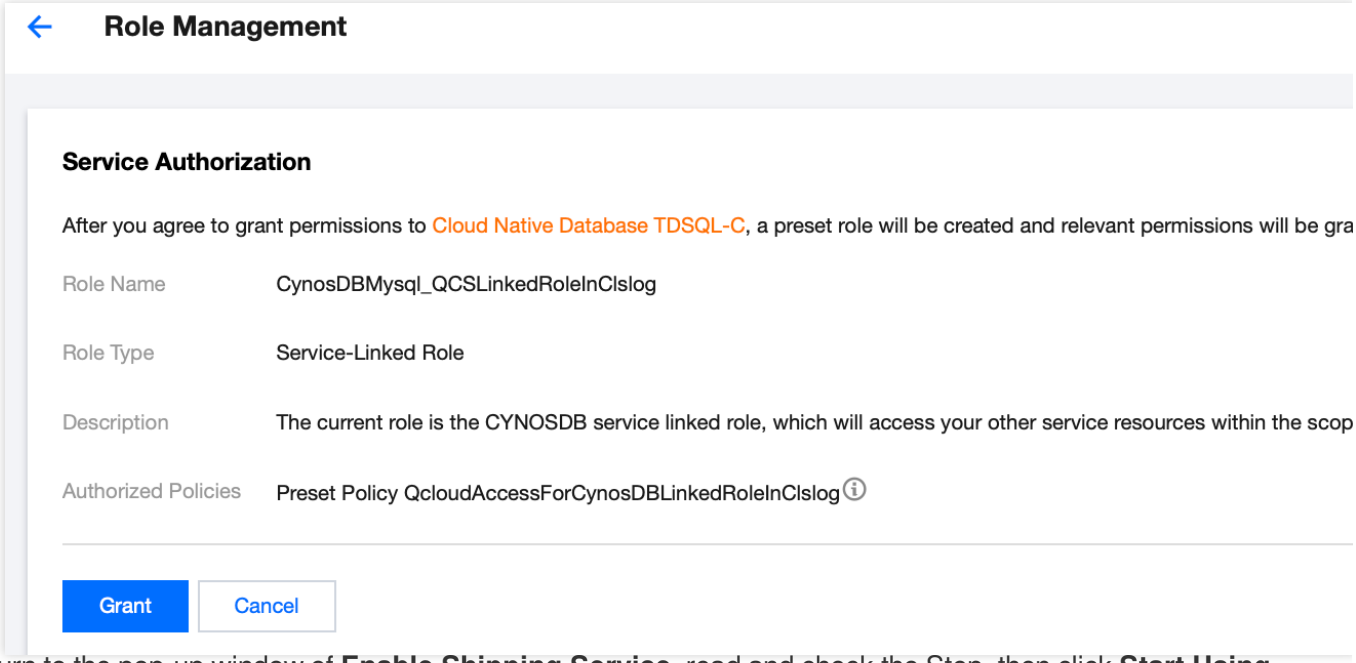
2. On the operation log page, select the instance that needs to enable the slow log shipping service, and then click **Slow Log Details** > **CLS (Activate Service)**.



3. Within the **Enable Shipping Service** pop-up window, click **Enable Now** to authorize the TDSQL-C service role (if you have already authorized, you can skip Steps 3 to 5).



4. On the page redirected to **Role Management**, click **Grant**.



5. Return to the pop-up window of **Enable Shipping Service**, read and check the Step, then click **Start Using**.

6. Click on

Add shipping logset

to conduct log shipping configuration.

7. Complete the following configuration in the pop-up window, then click **Activate Now**.

Parameter	Description
Region	Select the region for the logset shipping.
Logset operation	Logsets categorize log topics for your convenience in management. You can choose existing logsets or create new ones.
Logset	Select an existing logset: Filter through the search box to choose an existing logset for categorizing slow log shipping. Create a logset: You can create a new logset for the classification of slow log shipping, entering the name in the **cloud_cynos_custom_logset** format. The customised part only supports English alphabets, numbers, underscores, and hyphens.

Log topic operation	Log topic is the fundamental cell for log data collection, storage, retrieval and analysis. An existing log topic can be selected, or a new one can be created.
Log topic	Select an existing log topic: Only when the log set operation selects an existing log set, can this option be set. You may filter log topics under the selected log set in the search box. Create a log topic: Create a new log topic under the selected log set, the naming format for creating a log topic is **cloud_cynos_custom_topic**, where the custom part only supports English letters, numbers, underscores, and hyphens.

8. Once the CLS is successfully enabled, you can see the created **Logset**/ **Log Topic** in the log configuration pop-up window. Click **Go to CLS** to jump to the Cloud Log Service console for subsequent analysis and management.

Note:

Currently, when log shipping is initiated upon CLS is enabled, the index feature is not enabled by default. However, index configuration is a prerequisite for using CLS for search and analysis. A log can only be subjected to search and analysis if index is enabled. To activate the index feature, please refer to index configuration (CLS).

Disabling CLS

Note:

Upon disabling, log shipping to corresponding topics will cease. Though, shipped logs will be retained. If you want to delete logs, please proceed to the CLS console for deletion.

1. Refer to Enabling CLS > Step 1 to access the log operation page.
2. On the operation log page, select the instance that needs to disable the slow log shipping service, then click **Slow Log Details** > **CLS (Activate Service)**.
3. In the log configuration pop-up window, find the target **logset/log topic**, then click **Disable**.
4. In the pop-up window, click **OK**.

Feature Description of Log Configuration Page

Feature	Description
Enabled	Re-enable an existing log shipping configuration (currently disabled), and ship logs to CLS again. During this period, the log storage fees will be continuously charged.
Delete	Delete an existing log shipping configuration (currently disabled). However, historical logs will continue to be stored in the log topics until they expire. During this period, and the log storage fees will be continuously charged. If you do not want to continue storing historical logs, you may visit the CLS console to delete the log topics.

Add shipping logset	Create a new shipping logset. For operation steps, refer to Enabling CLS > Steps 6 to 8.

Documentation

Billing Overview
Logset
Dashboard
Data Processing
Search Analysis

Data Migration
Migrating with DTS

Last updated：2023-03-01 14:33:46

You can import your data to TDSQL-C for MySQL by using the data migration feature of DTS. For more information, see Migration from MySQL/MariaDB/Percona to TDSQL-C for MySQL.

Migrating with Command Line Tool

Last updated：2023-11-22 14:40:59

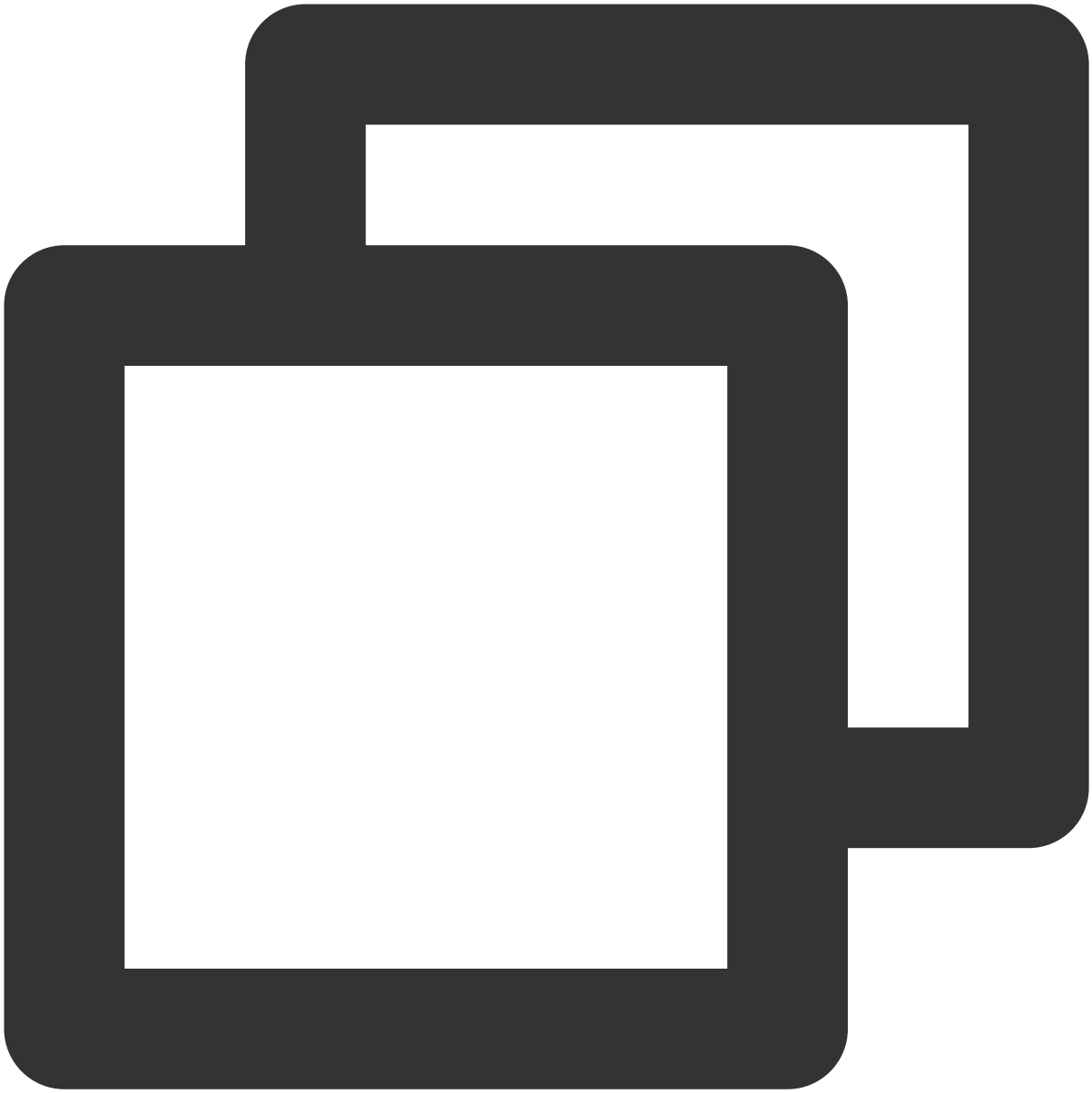
TDSQL-C for MySQL supports data migration through a command line tool.

Data Migration with the Command Line Tool

1. Generate the SQL file to be imported with the MySQL command line tool `mysqldump` in the following way:

Note:

The data files exported by using `mysqldump` must be compatible with the SQL specification of your purchased TDSQL-C for MySQL database. You can log in to the database and get the MySQL version information by running the `select version();` command. The name of the generated SQL file can contain letters, digits, and underscores but not "test".

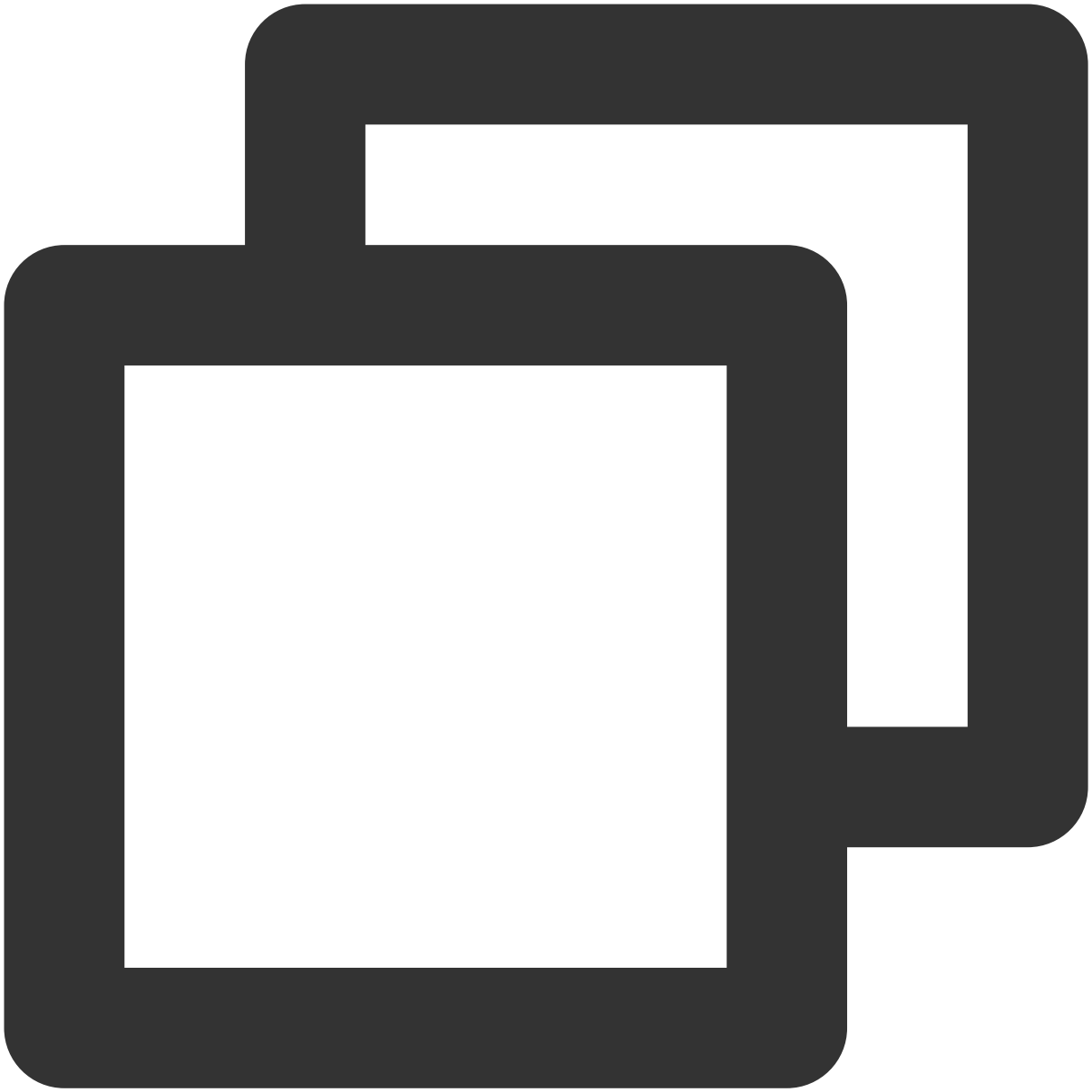


```
shell > mysqldump [options] db_name [tbl_name ...] > bak_pathname
```

Here, `options` is the export option, `db_name` is the database name, `tbl_name` is the table name, and `bak_pathname` is the export path.

For more information on how to export data with mysqldump, see mysqldump — A Database Backup Program.

2. A database can be restored with the MySQL command line tool by running the following command:



```
shell > mysql -h hostname -P port -u username -p < bak_pathname
```

Here, `hostname` is the target server for data restoration, `port` is the port of target server, `username` is the username of the database on the target server, and `bak_pathname` is the full path to the backup file.

Migrating Data on Windows

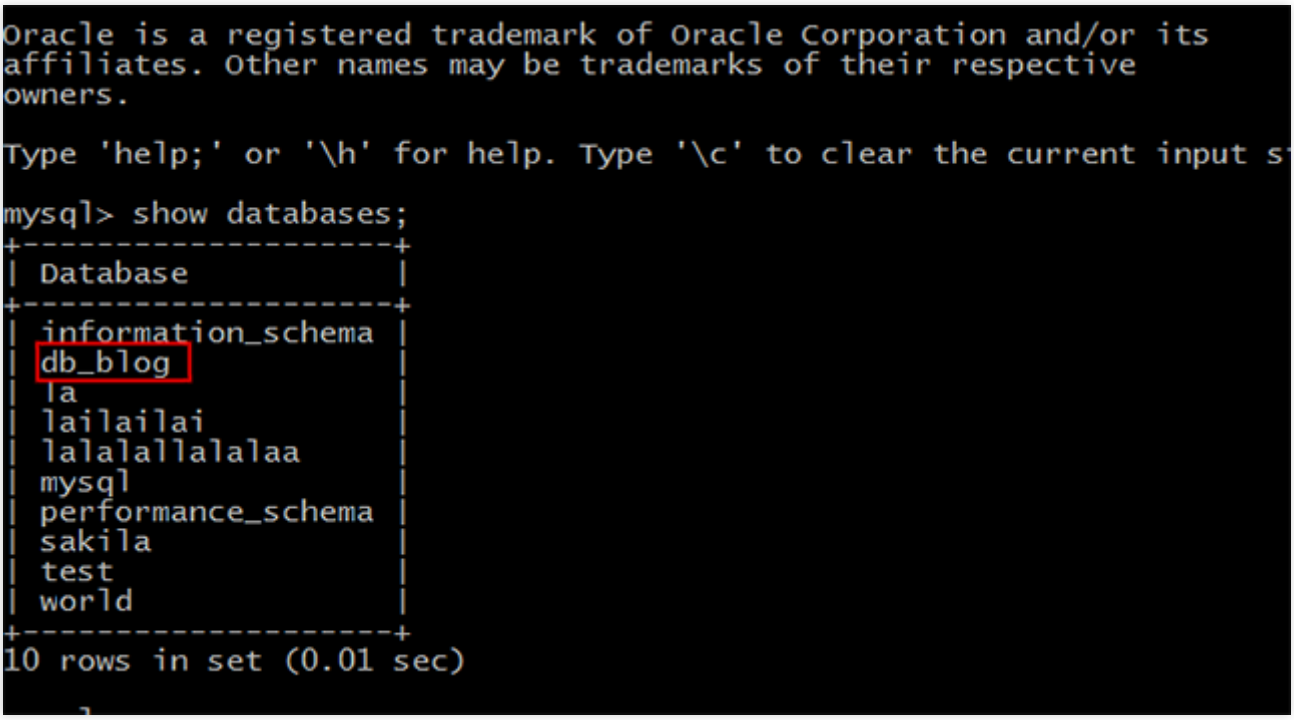
1. Use the Windows version of mysqldump to dump the data. For more information, see the description in Data Migration with the Command Line Tool.

Note:

Make sure that the same source and target database versions, mysqldump tool versions, and source and target database character sets are used. You can specify the character set using the parameter ``--default-character-set`` .

2. Enter the command prompt and restore the data with the MySQL command line tool.

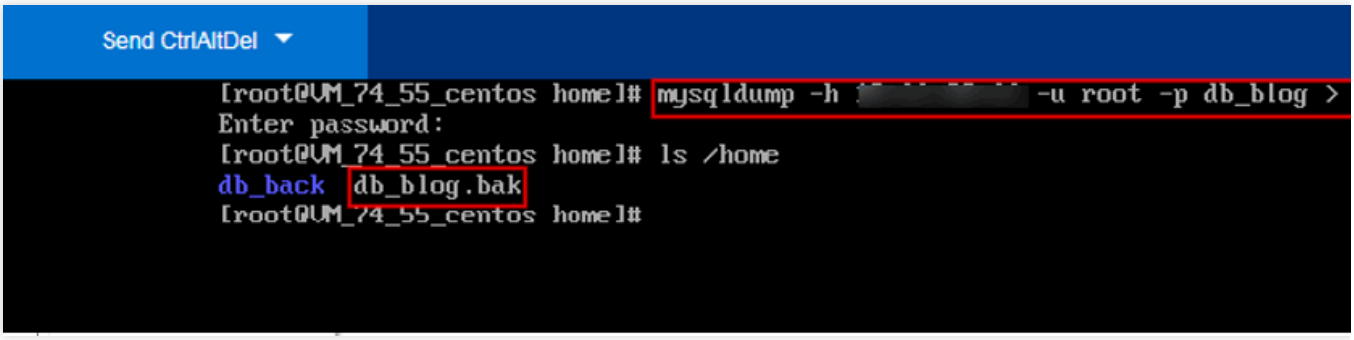
3. Log in to the MySQL database and you can see the backed up database has already been restored to the server.



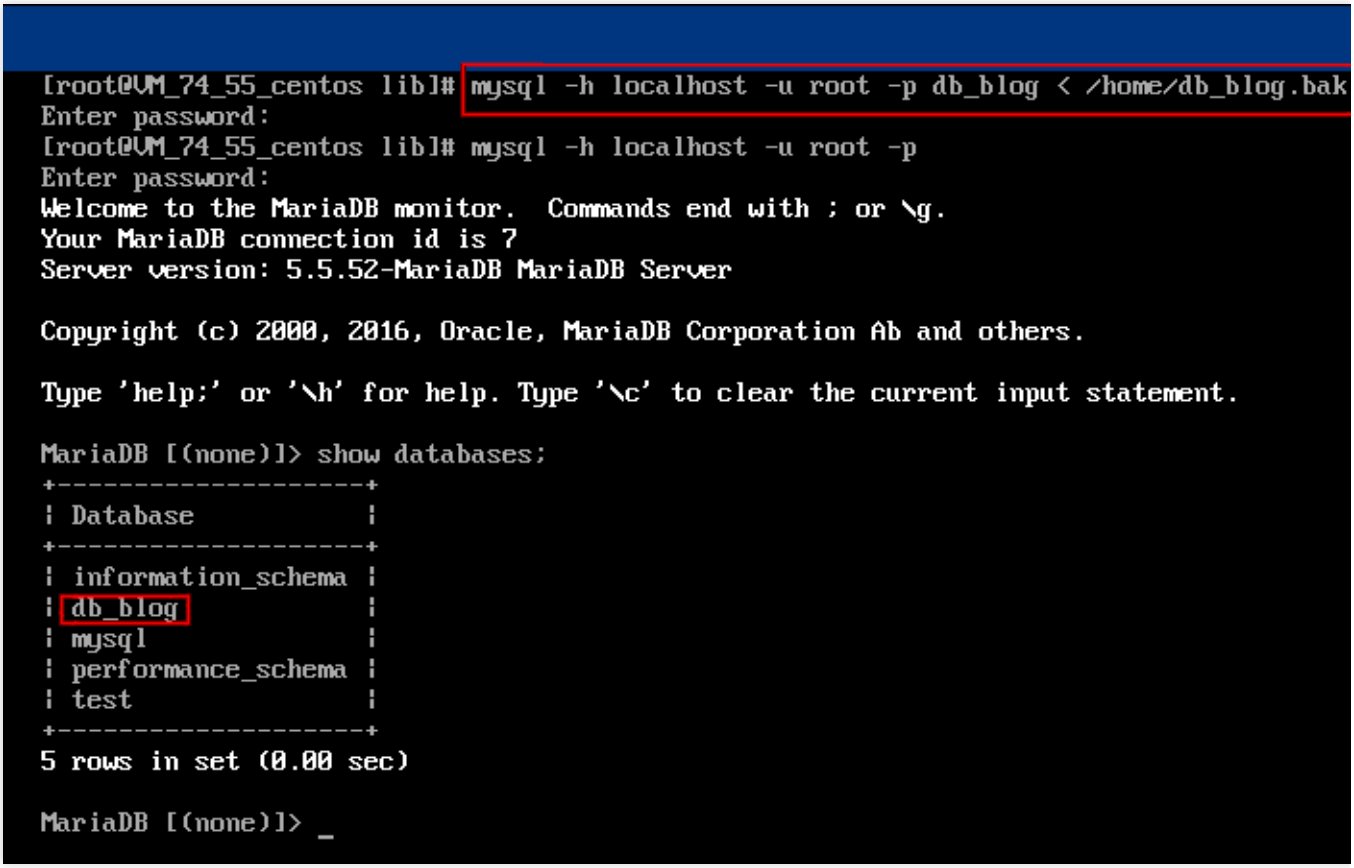
Migrating data on Linux CVM instance

For more information on how to access a database on a CVM instance, see Connecting to Cluster via the Console.

1. Taking the `db_blog` database in TencentDB for example, log in to the CVM instance and generate the SQL file to be imported with the MySQL command line tool "mysqldump".



2. Restore the data with the MySQL command line tool. In this example, data is restored to the CVM instance. You can see that the backed up database has been imported to the database corresponding to the target CVM instance.



Issues with the Character Set of Imported Data Files

1. If no character set is specified during data file import into the database, the one configured by the database will be used.
2. Otherwise, the specified character set will be used.
3. If the specified character set is different from that of TencentDB, garbled text will be displayed.

For more information, see the character set description in Use Limits.

Parallel Query Overview Background

Last updated：2023-01-06 11:35:30

This document describes the background of the parallel query capability of TDSQL-C for MySQL, which helps improve the efficiency in querying large tables with a large amount of computed data.

Feature background

Traditional MySQL-based relational databases still face storage redundancy and lack elastic loading after cloud-native transformation and cloud deployment. In contrast, TDSQL-C for MySQL adopts the "log as a storage" architecture to separate computing from storage and allows compute nodes to share underlying storage through the distributed file system. This reduces user costs, makes compute nodes stateless, and implements elastic expansion and failover of computing resources within seconds.
TDSQL-C for MySQL goes beyond traditional MySQL-based relational databases in terms of computing, storage, disaster recovery, and elastic expansion; however, it still faces the following challenges:
As the internet develops, cloud-native databases become more capable of storing data, and TDSQL-C for MySQL can store petabytes of data. Forms are carrying more and more data, with some containing terabytes of data online. When it comes to large table queries, SQL statements tend to be slow due to existing technical bottlenecks, which adversely affects the business process.
The current market environment sees an increasing number of AP queries, such as report statistics and other analytical queries. Although not large in number, they involve a high data volume and are quite sensitive to query time. Gradually, data analysis capability, especially heterogeneous data processing, has become a must-have.
The above challenges are caused by the traditional technical implementation mode in the MySQL ecosystem. In particular, open-source releases support only the single-thread query mode, where only one thread (called user thread) is responsible for the parsing, optimization, and execution of a SQL statement. This mode cannot make full use of the hardware resources of modern multi-core CPUs and large memory devices, leading to a resource waste. Therefore, it is important to streamline analysis and enhance performance by using multi-core services in the query of a large amount of data, which is also the key to query acceleration, cost reduction, and efficiency improvement. This is the so-called **parallel query (PQ)**, one of the core techniques to build HTAP product forms.

Strengths

Performance enhancement at no extra costs: You can upgrade the kernel capabilities at no extra costs, so that you can get the most out of the instance CPU computation for quicker statement response and higher computing performance.
Transparent process monitoring: You can use various metrics to monitor the parallel query, so that you can keep track of any exceptions and keep your cluster running stably.
Support for common statements: You can use most common SQL statements in virtually any business scenarios. This helps you accelerate your business smoothly.
Flexible parameter settings: You have many parameters at hand to control the conditions of enabling or disabling parallel query. This helps you make queries smarter and more adaptable to your business scenarios with no transformation needed.

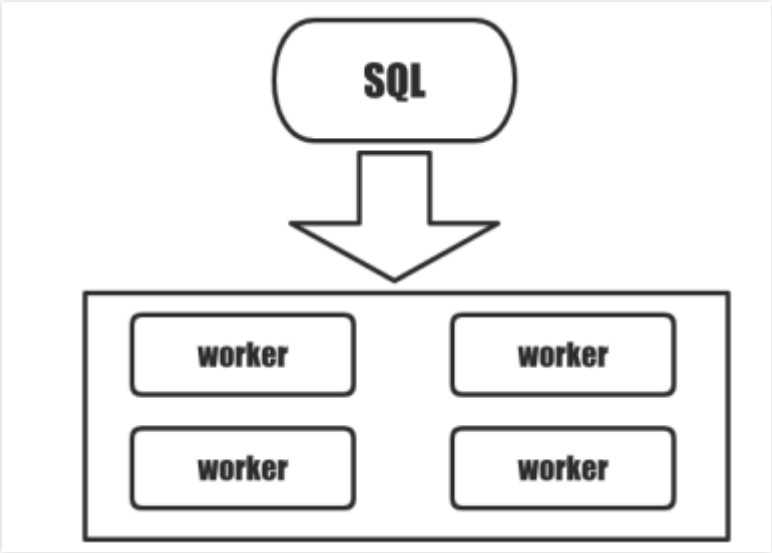
How It Works

Last updated：2023-01-06 11:35:30

This document describes how the parallel query feature works.

How parallelism works

TDSQL-C for MySQL offers the parallel query feature to push complete data partitions down to different threads for parallel computing, aggregate the result in the user thread, and return it to you. This greatly improves the query efficiency.

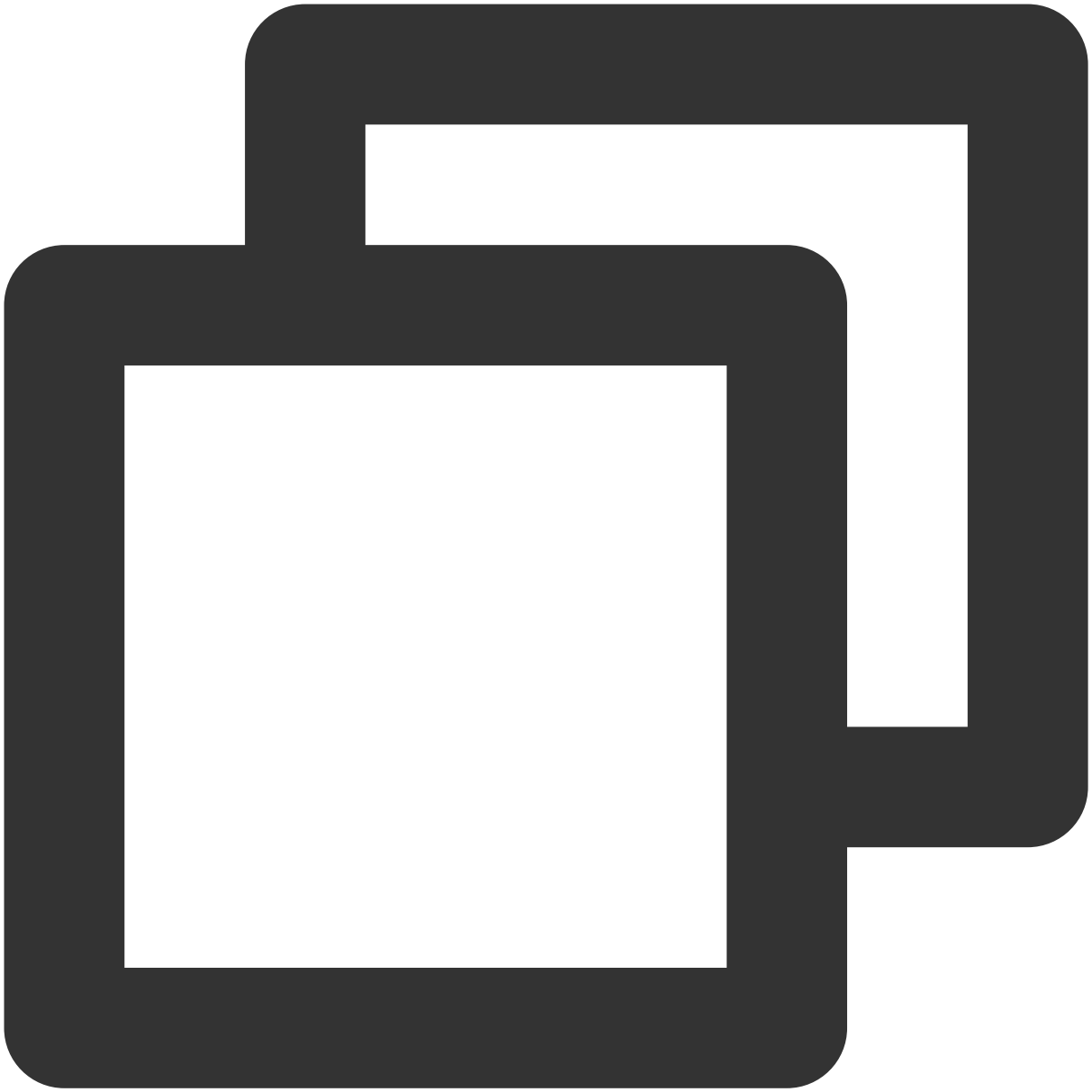


Below is an example of how parallel query works.

Note:

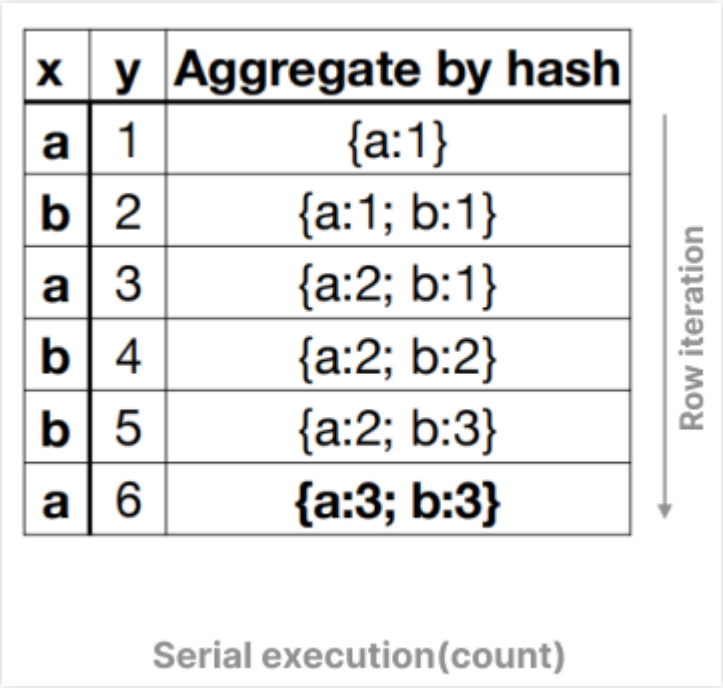
MySQL/InnoDB uses segmentation and paging physical storage, with a page being a partition. Here, a row is used to represent a partition.

Specify the `t` table for grouped and aggregated query as follows.

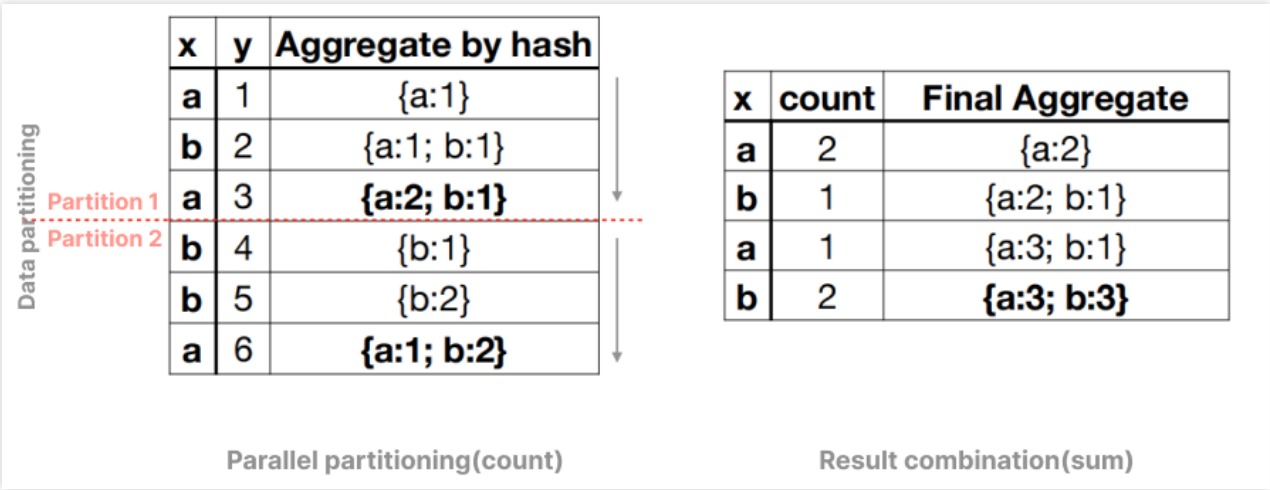


```
select x, count(*) from t group by x;
```

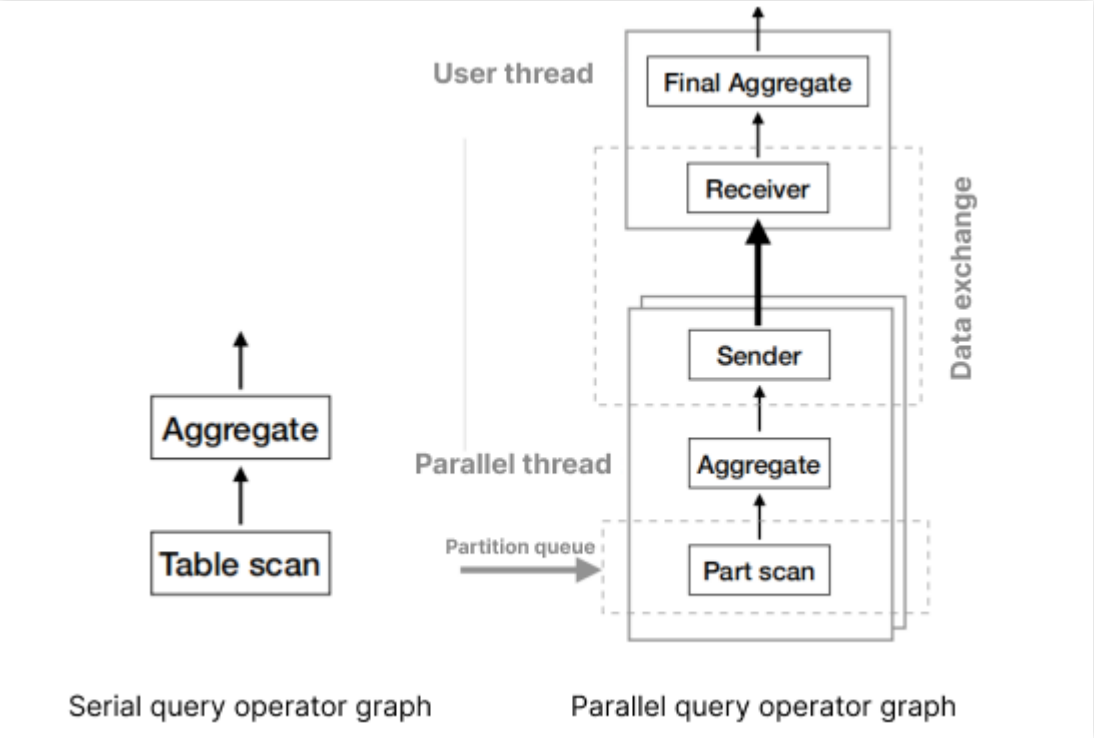
The hash aggregate algorithm (iterative evaluation) is performed as follows. When each row is iterated, the group aggregate status is updated. After all data rows are iterated, the group aggregate result is obtained. If the aggregate status update is fixed, then the time complexity of the algorithm is O(n).



If k threads are used to accelerate the query (here, k is `2`), it is better to first divide the data table into k * p partitions (here, p is `1`). In this way, each thread can process p partitions to generate an tentative local result. Then, local results are combined in the same user thread to get the final result.



Based on the above example, we can get the following operator graphs of serial and parallel queries.



As can be seen, in parallel query, data falls into different partitions that do not overlap based on parallelism. This process is called data partitioning.

After data is partitioned, special operations in the original plan are split, which is called task splitting. Then, data is scanned and executed in multiple worker threads (parallel threads), which aggregate the results to the user thread through the exchange operator. This process depends on data exchange.

After data exchange, the user thread completes the aggregation and outputs the complete result. Here, the user thread is responsible for data partitioning and task splitting and also works as the coordinator (also known as the coordinator thread) for the parallel execution of subtasks by worker threads. In addition, it combines the results and sends the final result to you. Worker threads execute parallel subtasks and exchange intermediate results through the data channel.

The core elements of parallel query can be concluded as follows:

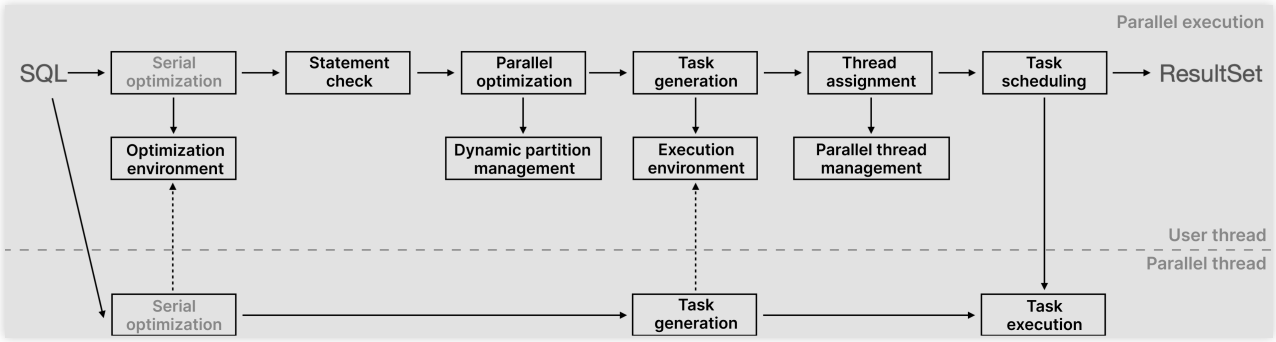
Data partitioning: Data in the original table is partitioned and read by partition.

Task splitting: Special operations in the original plan are split into "local-global" segments. In addition, the data exchange operator is inserted to support cross-thread data transfer.

Data exchange: Data is transferred among different threads.

Parallel query process

Based on the above principle, TDSQL-C for MySQL implements a complete parallel query plan to expand serial processing as shown below.



In a traditional MySQL serial process, serial optimization is first performed on a SQL statement to output a serial execution plan. Then, the iterative model is executed to output the result. The efficiency is low throughout the process. To implement parallel query, TDSQL-C for MySQL has redesigned SQL statement processing:

1. In parallel query, all processes are performed by the user thread and worker threads (parallel threads). After a SQL statement starts, the user thread analyzes it against the parameters in the optimizer and generate an execution plan. You can simply consider the SQL optimization environment as a highly abstract deterministic compute module, with the SQL and optimization environment as the input and the execution plan as the output. For the same input, the optimization path and output are the same, ensuring the accuracy of the result.
2. After the execution plan is generated, the statement check starts. At this point, the computing layer checks whether the statement meets the criteria for parallel query. The statement-level check includes dynamic query, data isolation level check (RC or RR), and execution cost check. At the execution plan level, the iteration operator and function are checked for parallel support. If the criteria are not met, the statement goes back to serial execution; otherwise, it proceeds to parallel query and optimization. For more information, see Supported Statement Scenarios and Restricted Scenarios.
3. During parallel optimization, the computing layer partitions the data in the target table as needed and splits the tasks of aggregation or sorting to be executed by worker threads. As optimization involves data partitioning, TDSQL-C for MySQL leverages the dynamic partition management capability to balance data in each thread, so as to execute multiple tasks in each thread and minimize data skew.
4. Then, the computing layer generates parallel query tasks, pushes them down to worker threads as task replicas, and configures the number of tasks on worker threads based on the configured parameters. Then, worker threads start working in parallel and push their results up to the user thread, which eventually returns the aggregated result to you. At this point, the parallel query of a statement ends.

In summary, after receiving the SQL statement, the user thread performs common operations such as parsing, verification, and optimization and generates a serial execution plan. In addition, it searches for the information (optimization environment) necessary for optimization. Then, it analyzes the serial execution plan (statement check,

i.e., check of the operator tree, execution environment, and optimization cost) and decides whether to start parallel optimization. During parallel optimization, the serial operator tree is divided into several coarse-grained tasks, the algorithm is determined for data exchange between parallel tables (dynamic partitions) and tasks, and a task dependency graph is drawn. At this point, the user thread is ready, which means execution can start after enough worker threads are obtained from the available thread manager. After worker threads complete their work, data is exchanged to the user thread for aggregation. Then, the final result is returned to you.
In the above process, TDSQL-C for MySQL provides various parameters to adjust parallel query capabilities and control the resource load caused by the statement execution cost and parallel query. It also offers many metrics for you to monitor parallel query information in real time. For more information, see Enabling/Disabling Parallel Query.

Supported Statement Scenarios and Restricted Scenarios

Last updated：2023-01-06 11:35:30

This document describes the supported statement scenarios and restricted scenarios of the parallel query.

Supported statement scenarios

TDSQL-C for MySQL has implemented the parallel query feature for SQL statements with the following characteristics, with more to come.

Single-table scan: Full-table scan, index scan, index range scan, and index REF query in ascending or descending order are supported.

Multi-table join: The nested-loop join (NLJ) algorithm as well as semi join, anti join, and outer join are supported.

Subquery: Parallel query is supported for derived tables.

Data type: Different data types can be queried, such as integer, string, floating point, time, and overflow (with a runtime size limit).

There are no restrictions on common operators and functions.

COUNT, SUM, AVG, MIN, and MAX aggregate functions are supported.

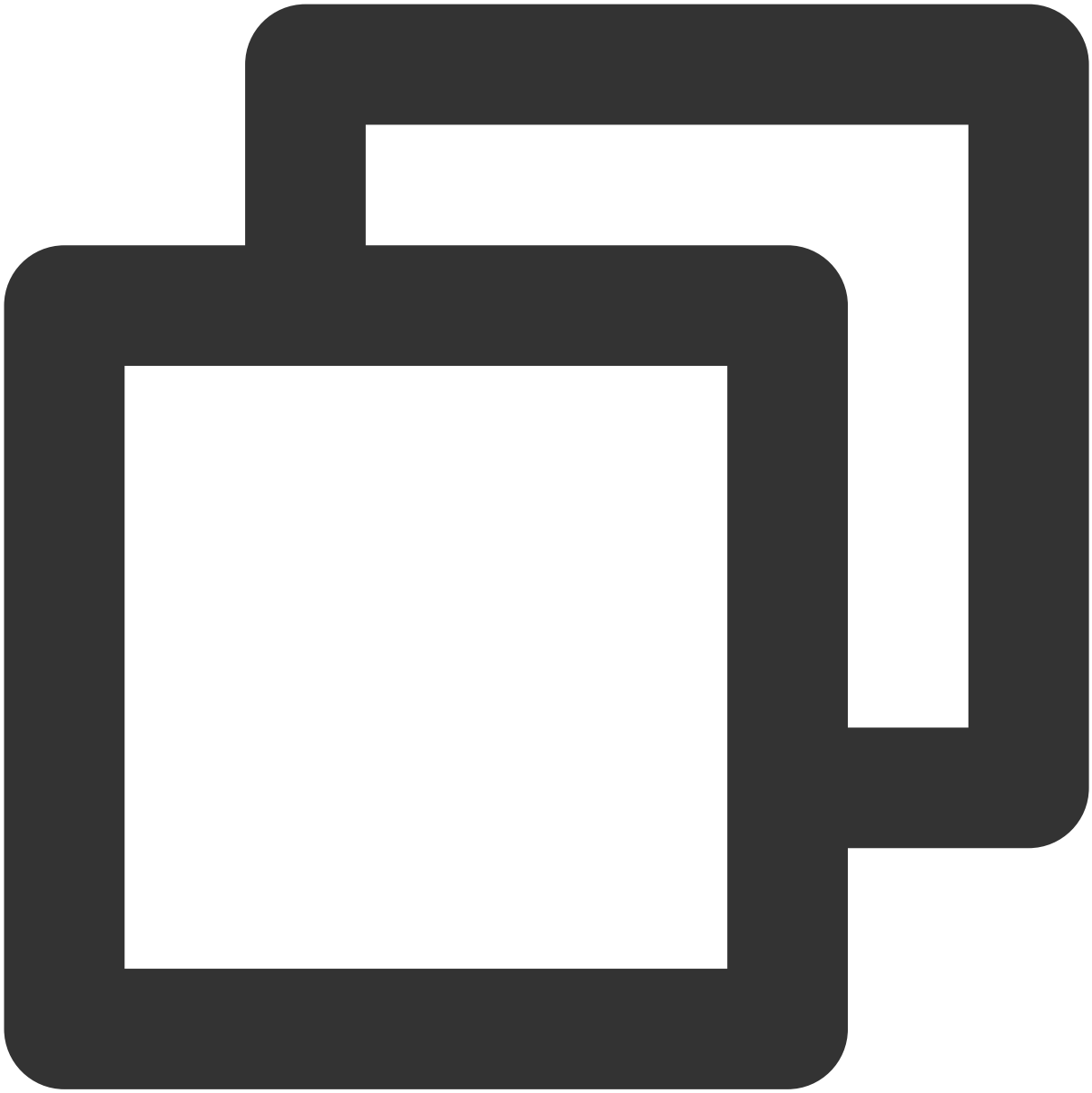
UNION and UNION ALL queries are supported.

Traditional (default), JSON, and tree EXPLAIN formats are supported.

Statement performance improvement

If `SF` is `100` and `Dop` is `16`, the acceleration ratio is as follows.

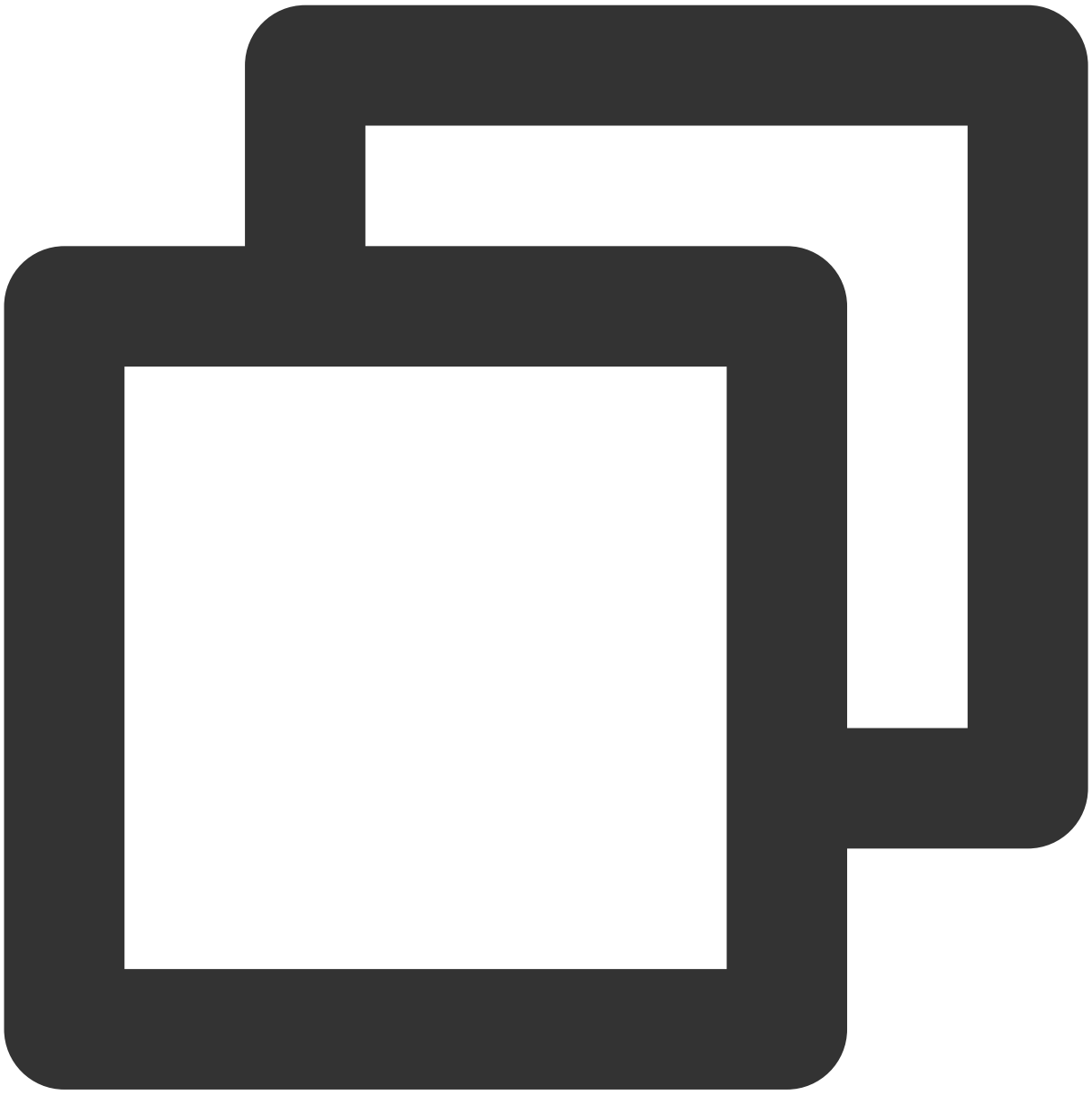
SUM, AVG, or COUNT aggregate function



```
SELECT l_returnflag, l_linestatus,
Sum(l_quantity) AS sum_qty,
Sum(l_extendedprice) AS sum_base_price,
Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
Avg(l_quantity) AS avg_qty,
Avg(l_extendedprice) AS avg_price,
Avg(l_discount) AS avg_disc,
Count(*) AS count_order FROM
lineitem WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag, l_linestatus ;
```

The execution takes 1,376.96 seconds and 107.25 seconds before and after parallel query is enabled, indicating an acceleration ratio of 12.84.

ORDER BY or GROUP BY statement

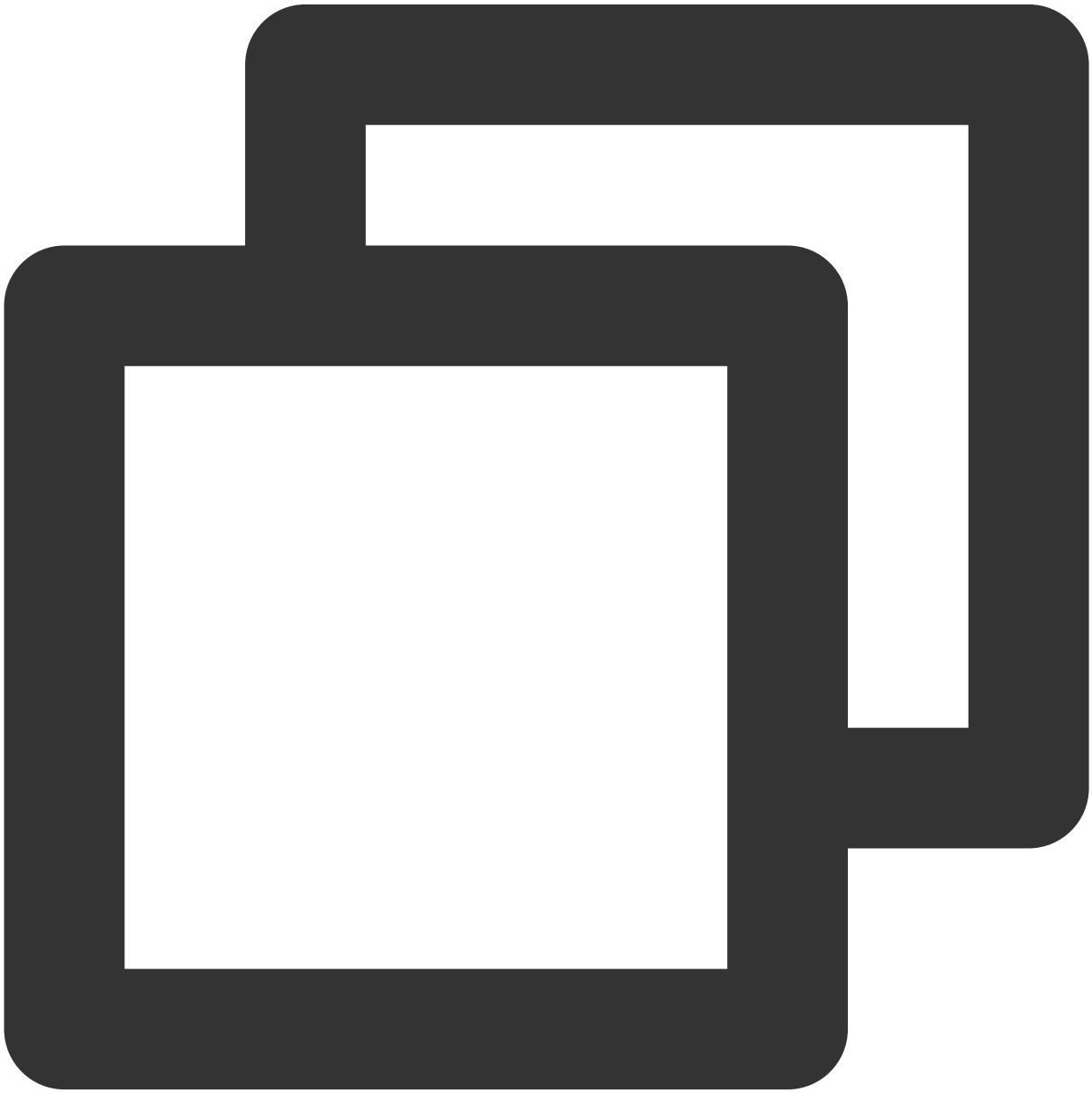


```
SELECT l_returnflag, l_linestatus,
Sum(l_quantity) AS sum_qty,
Sum(l_extendedprice) AS sum_base_price,
Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
Avg(l_quantity) AS avg_qty,
```

```
Avg(l_extendedprice) AS avg_price,
Avg(l_discount) AS avg_disc,
Count(*) AS count_order FROM
lineitem WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag, l_linestatus ;
```

The execution takes 1,376.96 seconds and 107.25 seconds before and after parallel query is enabled, indicating an acceleration ratio of 12.84.

JOIN, BETWEEN, or IN statement



```
select
```

```
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#12'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
and l_quantity >= 6
and l_quantity <= 6 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#13'
and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
and l_quantity >= 10 and l_quantity <= 10 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#24'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
and l_quantity >= 21
and l_quantity <= 21 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
);
```

The execution takes 20.55 seconds and 1.87 seconds before and after parallel query is enabled, indicating an acceleration ratio of 11.

Restricted scenarios

The parallel query feature of TDSQL-C for MySQL is not supported in the following scenarios.

Restriction	Description

Statement compatibility restriction	Parallel query is not supported for non-query statements, including INSERT ... SELECT and REPLACE ... SELECT.
	Parallel query is not supported for statements in a stored program.
	Parallel query is not supported for prepared statements.
	Parallel query is not supported for statements in serial isolation-level transactions.
	Parallel query is not supported for locking reads, such as SELECT FOR UPDATE and SELECT ... FOR SHARE.
	Parallel query is not supported for CTEs.
Table/Index compatibility restriction	Parallel query is not supported for system, temp, and non-InnoDB tables.
	Parallel query is not supported for space index.
	Parallel query is not supported for full-text index.
	Parallel query is not supported for partitioned tables.
	Parallel query is not supported for tables in `index_merge` scan mode.
Expression/Field compatibility restriction	Parallel query is not supported for tables containing generated columns or BLOB, TEXT, JSON, BIT, and GEOMETRY fields.
	Parallel query is not supported for aggregate functions of the BIT_AND, BIT_OR, or BIT_XOR type.
	Parallel query is not supported for DISTINCT aggregations, such as SUM(DISTINCT) and COUNT(DISTINCT).
	Parallel query is not supported for GIS functions such as SP_WITHIN_FUNC and ST_DISTANCE.
	Parallel query is not supported for custom functions.
	Parallel query is not supported for JSON functions such as JSON_LENGTH, JSON_TYPE, and JSON_ARRAYAGG.
	Parallel query is not supported for XML functions such as XML_STR.
	Parallel query is not supported for user-lock functions such as IS_FREE_LOCK, IS_USED_LOCK, RELEASE_LOCK, RELEASE_ALL_LOCKS, and GET_LOCK.
	Parallel query is not supported for SLEEP, RANDOM, GROUP_CONCAT, SET_USER_VAR, and WEIGHT_STRING functions.

	Parallel query is not supported for certain statistical functions such as STD, STDDEV, STDDEV_POP, VARIANCE, VAR_POP, and VAR_SAMP.
	Parallel query is not supported for subqueries.
	Parallel query is not supported for window functions.
	Parallel query is not supported for ROLLUP.

Besides the above examples in Supported statement scenarios, you can also check the parallel query execution plan and thread working status to see whether a statement can be queried parallelly. For more information, see Viewing Parallel Query.

Setting Parallel Query
Enabling/Disabling Parallel Query

Last updated：2023-01-06 11:35:30

This document describes how to enable or disable the parallel query feature of TDSQL-C for MySQL via the console or command line.

Prerequisites

Database version: TDSQL-C for MySQL 8.0 on kernel version 3.1.8 or later.

Parameters

Note:
The parallel query feature can be enabled for both read-write and read-only instances, as long as their number of CPU cores is greater than or equal to 4.
You can enable the parallel query feature for the current instance by setting the `txsql_max_parallel_worker_threads` and `txsql_parallel_degree` parameters to a value other than `0` via the console or command line. Parameters and suggested settings are as follows:

Parameter information

Parameter	Variable Type	Scope	Default Value	Value Range	Description
txsql_max_parallel_worker_threads	Integer	Global	0	0–1024	The total number of threads of the instance node that can be used for parallel query. If it is set to `0`, no parallel thread is available, indicating to disable the parallel query feature.
txsql_parallel_degree	Integer	Global/session	0	0–	The maximum

				1024	number of threads (default parallelism) that can be used during the parallel query of a single statement. `0` indicates to disable the parallel query feature.

Suggested settings

Parallelism limit: `txsql_parallel_degree` indicates the maximum number of threads for the parallel query of a single statement, i.e., the default parallelism. We recommend you limit this value to half of the CPU core quantity of the instance. To ensure the stability, the parallel query feature is disabled for small clusters with fewer than four CPU cores, and you cannot adjust parallel query parameters via the console or command line.

During the parallel query of a SQL statement, the parallelism set by `txsql_parallel_degree` will be used by default, which can be adjusted through the HINT statement. For more information, see HINT Statement Control.

`txsql_max_parallel_worker_threads` indicates the number of threads of the instance that can be used for parallel query, and `txsql_max_parallel_worker_threads / txsql_parallel_degree` indicates the maximum number of SQL statements allowed in a parallel query.

`txsql_max_parallel_worker_threads` and `txsql_parallel_degree` control the status of the parallel query feature. When either of them is `0`, the feature is disabled.

TDSQL-C for MySQL offers various parameters for you to set the execution conditions of parallel query for business adaptation and stability. After conditions are set, TDSQL-C for MySQL will check whether each SQL statement can be executed against such conditions like execution cost, number of table rows, and memory usage for the parallel statement execution. Parameters are described as follows:

Parameter	Variable Type	Scope	Default Value	Value Range
txsql_parallel_cost_threshold	Numeric	Global/Session	50000	0–9223372036854775807

txsql_parallel_table_record_threshold	Integer	Global/Session	5000	0–9223372036854775807
txsql_optimizer_context_max_mem_size	Integer	Global	8388608	0–9223372036854775807
txsql_parallel_execution_max_lob_size	Integer	Global/Session	65536	128–9223372036854775807

Note:

Parallel query parameters take effect immediately after being set, with no instance restart required.

If the scope of a parameter is session, it takes effect only for the statement.

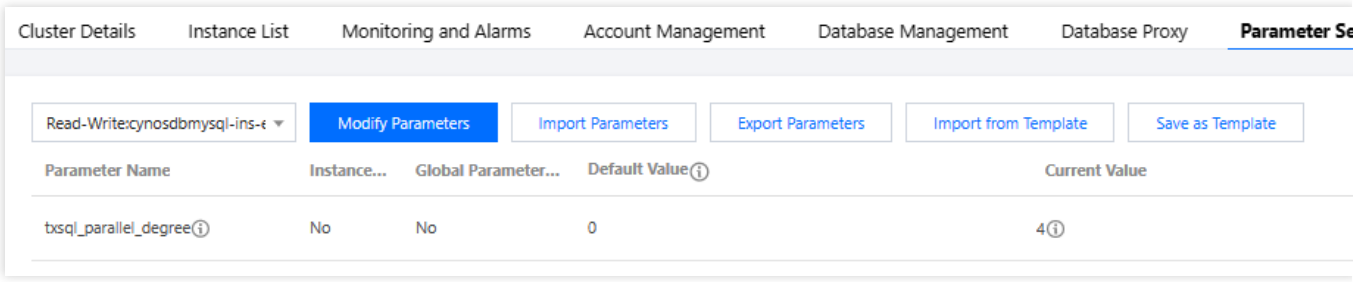
Enabling or disabling parallel query in the console

You can enable or disable the feature by setting parameters on the **Parameter Settings** page in the TDSQL-C for MySQL console.

Set `txsql_max_parallel_worker_threads` and `txsql_parallel_degree` to a value other than `0` to enable parallel query.

Set `txsql_max_parallel_worker_threads` or `txsql_parallel_degree` to `0` to disable parallel query.

You can also set execution conditions on the **Parameter Settings** page. For detailed directions, see Setting Instance Parameters.



Specifying the parallel execution mode of a SQL statement through the HINT statement

TDSQL-C for MySQL allows you to specify the parallel execution mode of a SQL statement through the HINT statement. For detailed directions, see HINT Statement Control.

References

Viewing Parallel Query

Parallel Query Metrics

HINT Statement Control

Last updated：2023-02-23 11:00:59

TDSQL-C for MySQL allows you to enable or disable the parallel query feature by adjusting parameters. Specifically, you can enable or disable the feature for all SQL statements, set execution conditions, or specify the execution mode of a specific SQL statement through the HINT statement in the console.

Note:

The HINT statement can specify whether to execute a SQL statement and apply session parameters to the statement. In addition, it also supports querying the specified parallel table.

HINT statement usage

Feature	Command Line	Description
Enable parallel query	`SELECT /*+PARALLEL(x)*/ ... FROM ...;`	`x` indicates the parallelism for the SQL statement, which should be greater than `0`.
Disable parallel query	`SELECT /*+PARALLEL(x)*/ ... FROM ...;`	If `x` is set to `0`, it indicates to disable parallel query.
Specify the parallel table	You can specify the table to be included in or excluded from the parallel query execution plan in either of the following ways: Specify the table to be included in the plan through `PARALLEL`. `SELECT /*+PARALLEL(t)*/ ... FROM ...;` Specify the table to be excluded from the plan through `NO_PARALLEL`. `SELECT /*+NO_PARALLEL(t)*/ ... FROM ...;`	`t` is the table name.
Specify both the parallel table and parallelism	`SELECT /*+PARALLEL(t x)*/ * ... FROM ...;`	`x` indicates the parallelism for the SQL statement, which should be greater than `0`. `t` is the table name.

Set the session parameter through the HINT statement, which takes effect only for the specified SQL statement	`SELECT /*+SET_VAR(var=n)*/ * ... FROM ...;`	`var` is the parallel query parameter in the `session` scope.

HINT statement use cases

Use case 1: `select /*+PARALLEL()*/ * FROM t1,t2;`
Set the parallelism to the value of `txsql_parallel_degree` (default) for the parallel query. If a statement does not meet the parallel query execution condition, serial query will be used.
Use case 2: `select /*+PARALLEL(4)*/ * FROM t1,t2;`
Set the parallelism of the statement to `4` regardless of the default value, i.e., `txsql_parallel_degree = 4`. If the statement does not meet the parallel query execution condition, serial query will be used.
Use case 3: `select /*+PARALLEL(t1)*/ * FROM t1,t2;`
Include the `t1` table in the parallel query and use the default parallelism. If `t1` is smaller than the value of `txsql_parallel_table_record_threshold`, serial query will be used.
Use case 4: `select /*+PARALLEL(t1 8)*/ * FROM t1,t2;`
Include the `t1` table in the parallel query and set the parallelism to `8`. If `t1` is smaller than the value of `txsql_parallel_table_record_threshold`, serial query will be used.
Use case 5: `select /*+NO_PARALLEL(t1)*/ * FROM t1,t2;`
Exclude the `t1` table from the parallel query. If `t1` is greater than the value of `txsql_parallel_table_record_threshold`, serial query will be used.
Use case 6: `select /*+SET_VAR(txsql_parallel_degree=8)*/ * FROM t1,t2;`
Set the parallelism of the statement to `8` regardless of the default value, i.e., `txsql_parallel_degree = 8`.
Use case 7: `select /*+SET_VAR(txsql_parallel_cost_threshold=1000)*/ * FROM t1,t2`
Set `txsql_parallel_cost_threshold=1000` for the statement. If its execution penalty is greater than `1000`, parallel query can be used.
Use case 8: `select /*+SET_VAR(txsql_optimizer_context_max_mem_size=500000)*/ * FROM t1,t2`
Set `txsql_optimizer_context_max_mem_size=500000` for a statement, which means to adjust the maximum memory size it can apply for in the parallel query plan environment to `500000`.

References

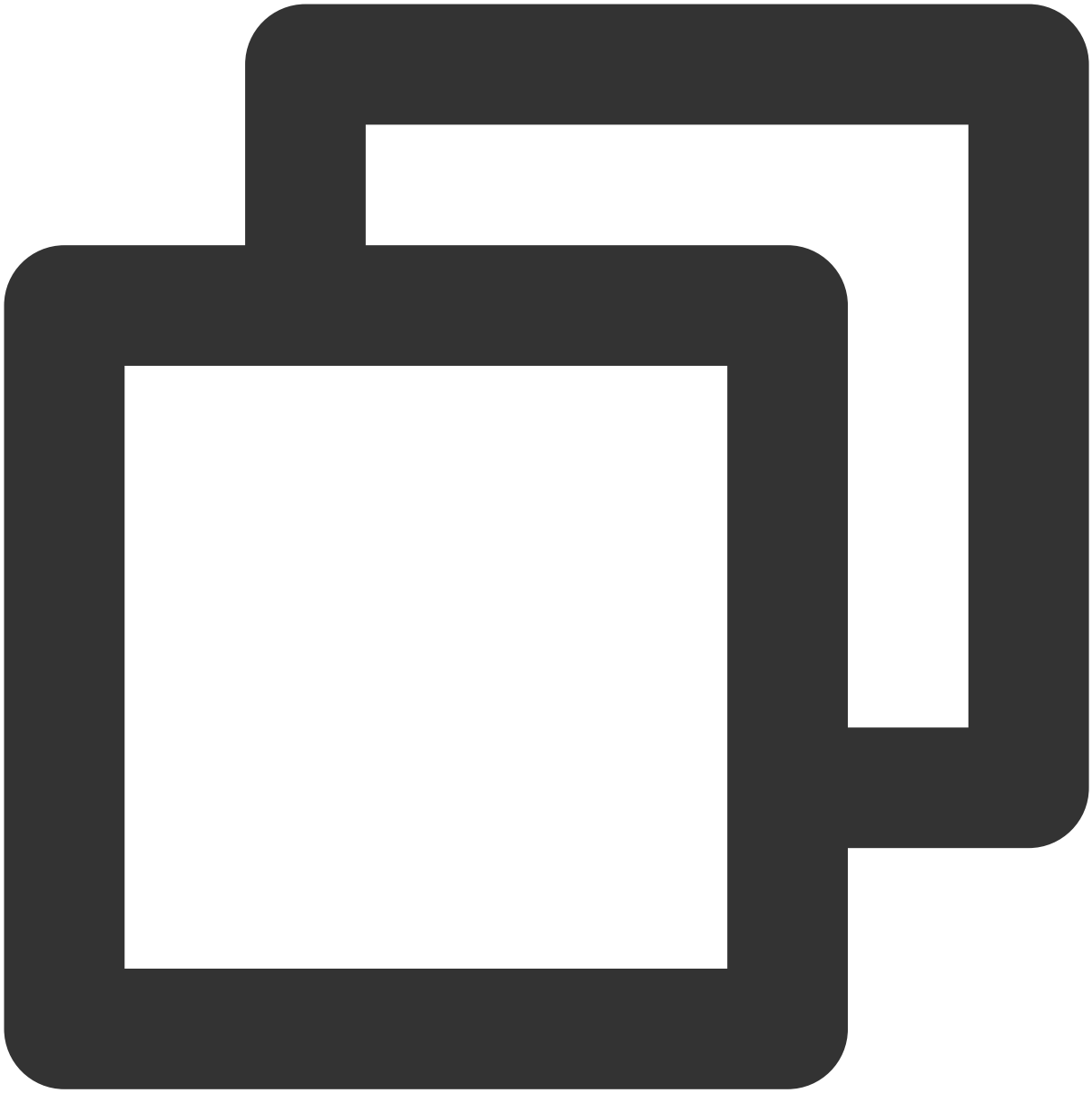
Viewing Parallel Query

Last updated：2023-01-06 11:35:30

TDSQL-C for MySQL allows you to view the parallel query execution plan and threads in the plan, so that you can clearly know how parallel query takes effect in a database and quickly troubleshoot issues.
This document describes two common methods for viewing parallel queries.

Option 1: Using the EXPLAIN statement

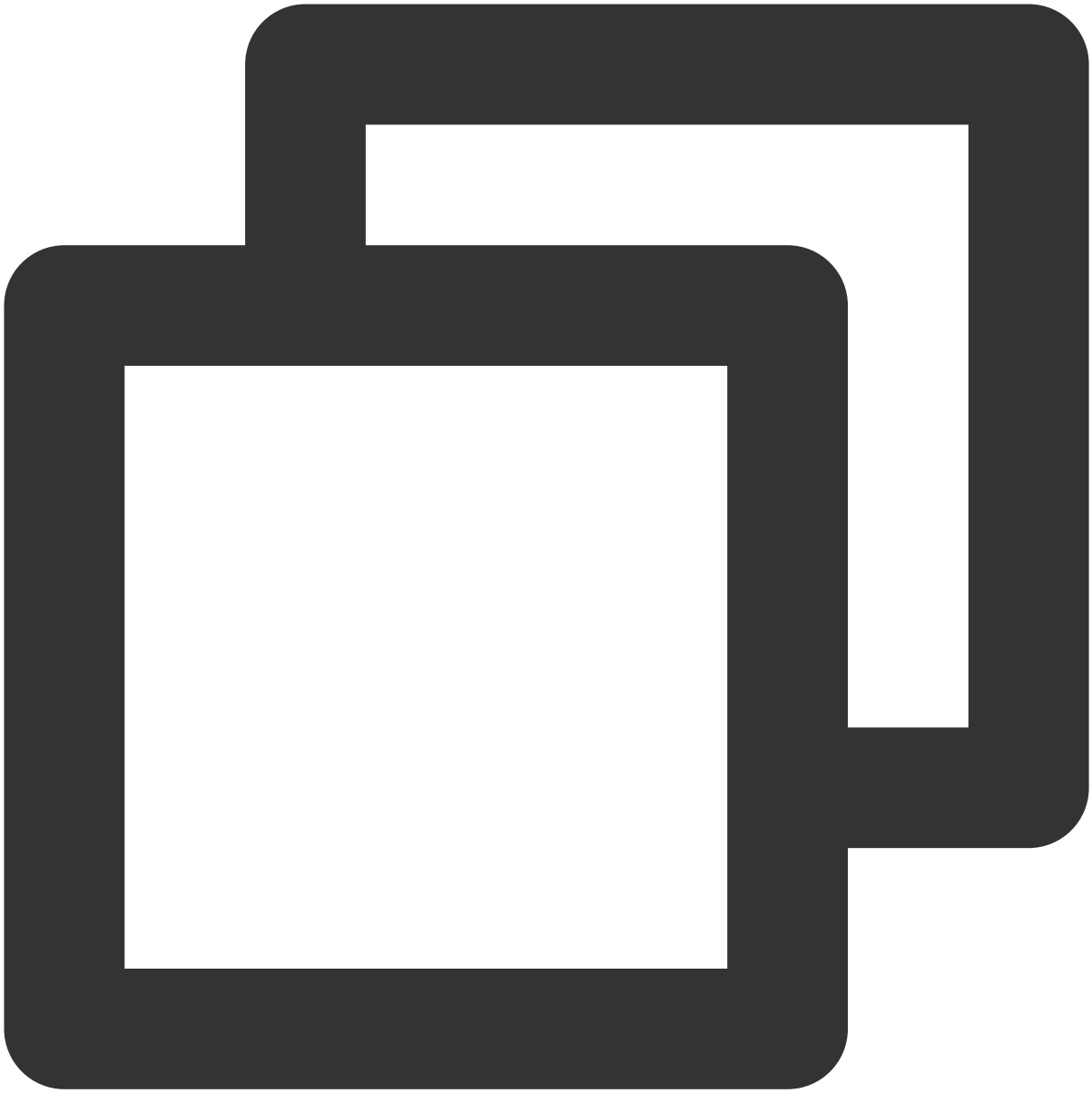
Sample SQL statement:



```
SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

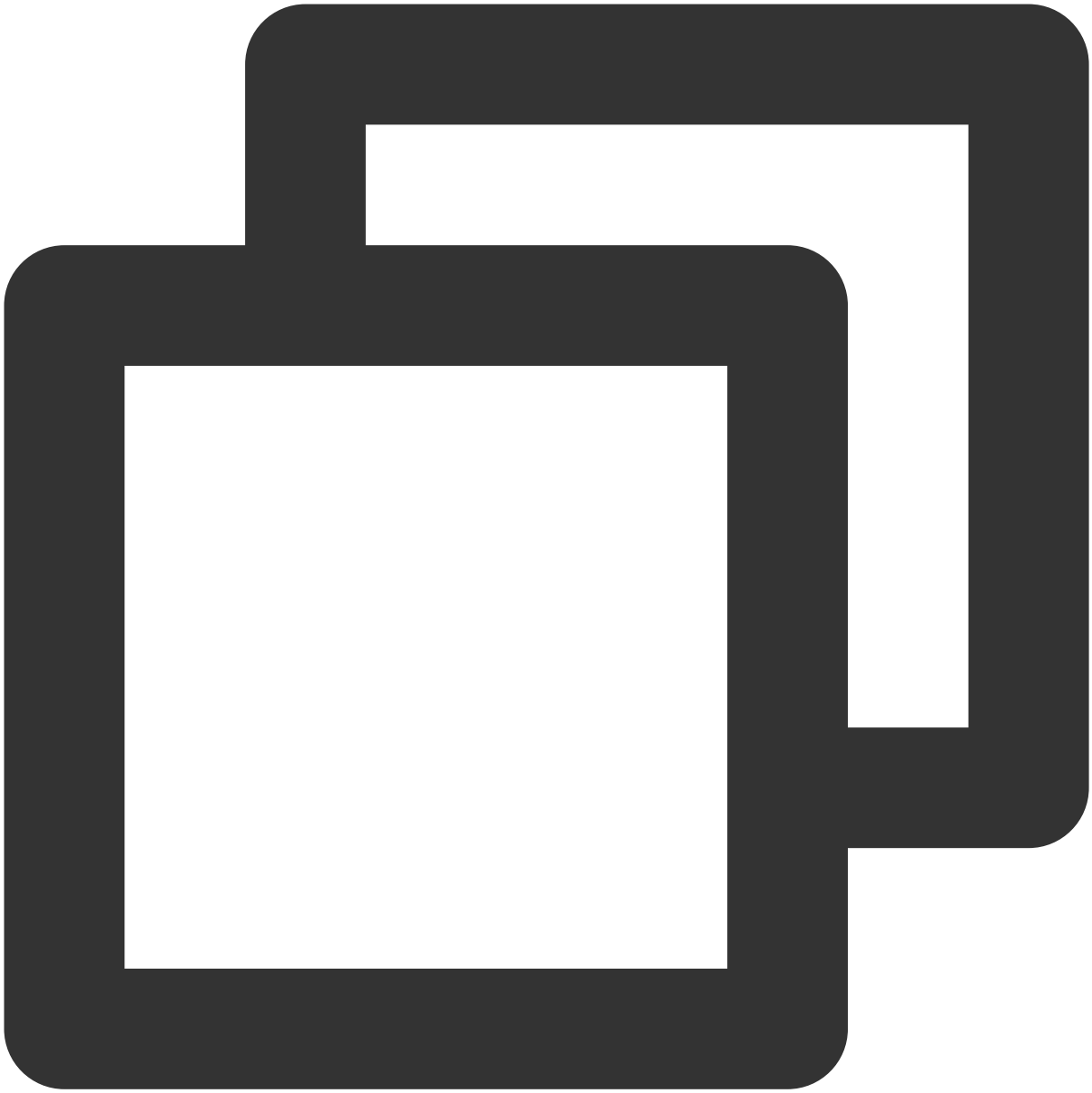
This sample is a simplified version of TPC-H Q1, a typical report operation.

EXPLAIN statement:



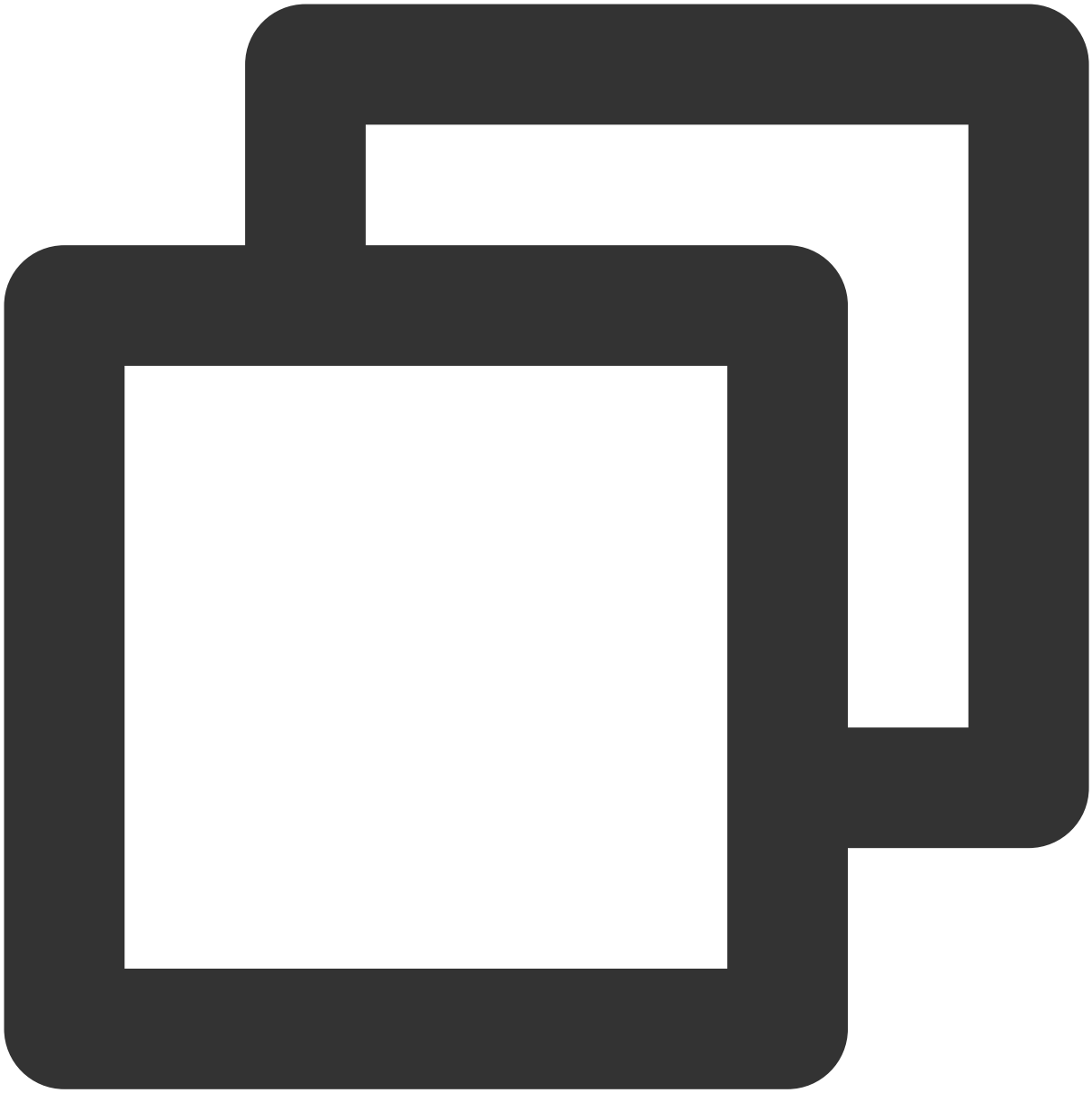
```
EXPLAIN  SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

Query result:



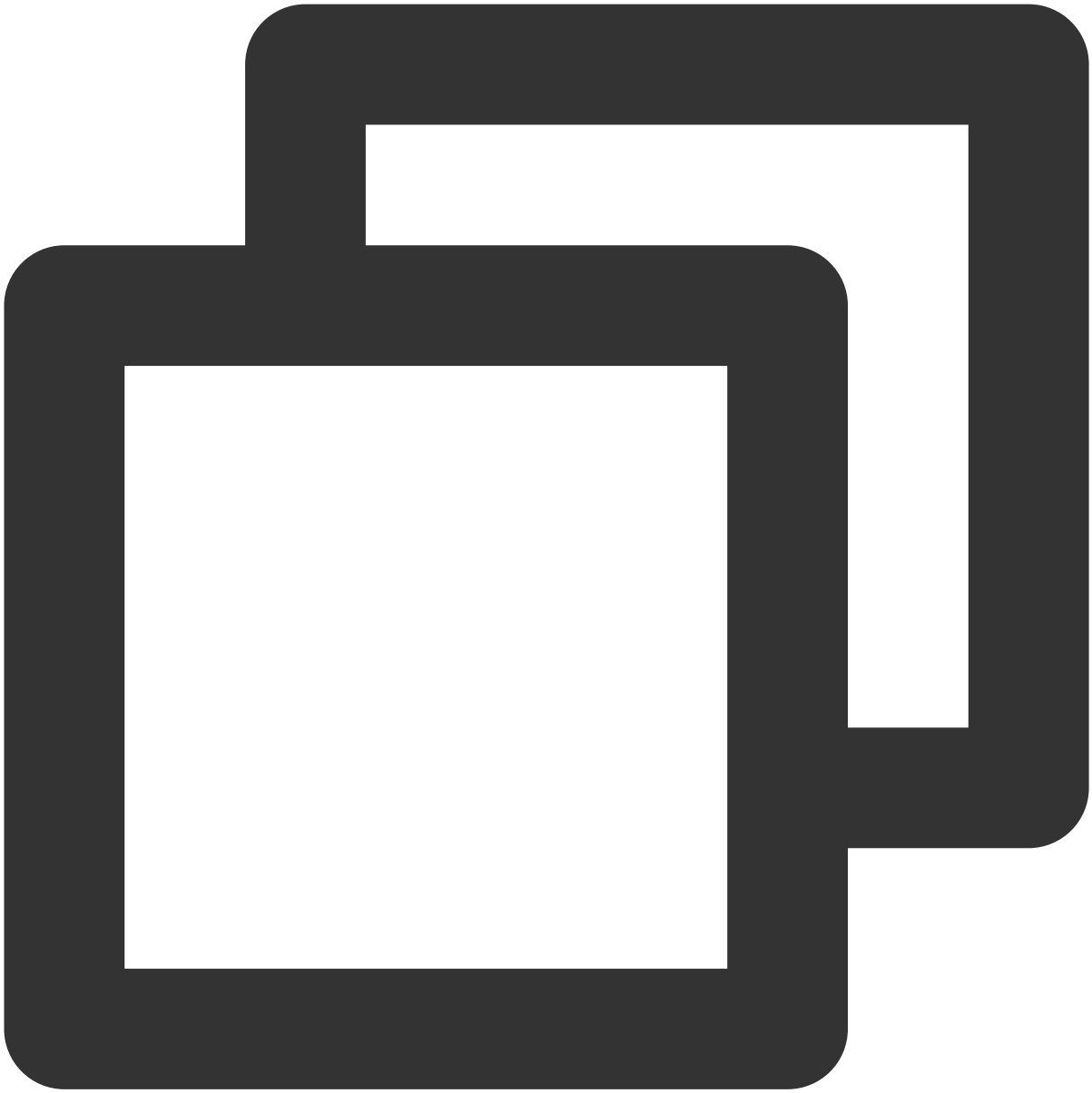
```
MySQL [tpch100g]> explain SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum
+----+-------------+-------------+------------+------+---------------+------+------+
| id | select_type | table       | partitions | type | possible_keys | key  | key_l
+----+-------------+-------------+------------+------+---------------+------+------+
|  1 | SIMPLE      | lineitem    | NULL       | ALL  | i_l_shipdate  | NULL | NULL
|  1 | SIMPLE      | <sender1>   | NULL       | ALL  | NULL          | NULL | NULL
|  1 | SIMPLE      | <receiver1> | NULL       | ALL  | NULL          | NULL | NULL
+----+-------------+-------------+------------+------+---------------+------+------+
3 rows in set, 1 warning (0.00 sec)
```

EXPLAIN format=tree:



```
EXPLAIN format=tree query  SELECT l_returnflag, l_linestatus, sum(l_quantity) as su
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

Query result:



```
MySQL [tpch100g]> explain format=tree SELECT l_returnflag, l_linestatus, sum(l_quan
*************************** 1. row ***************************
EXPLAIN: -> Sort: lineitem.L_RETURNFLAG, lineitem.L_LINESTATUS
    -> Table scan on <temporary>
        -> Final Aggregate using temporary table
            -> PX Receiver (slice: 0; workers: 1)
                -> PX Sender (slice: 1; workers: 4)
                    -> Table scan on <temporary>
                        -> Aggregate using temporary table
                            -> Filter: (lineitem.L_SHIPDATE <= DATE'1998-09-02')  (
                                -> Parallel table scan on lineitem  (cost=65449341.
```

```
1 row in set (0.00 sec)
```

As can be seen from the above result:

The parallel query plan assigns the statement to four worker threads for computing.

Aggregate operations are split into two segments that are executed by the user and parallel threads respectively.

The parallel scan operator is used for the `lineitem` table.

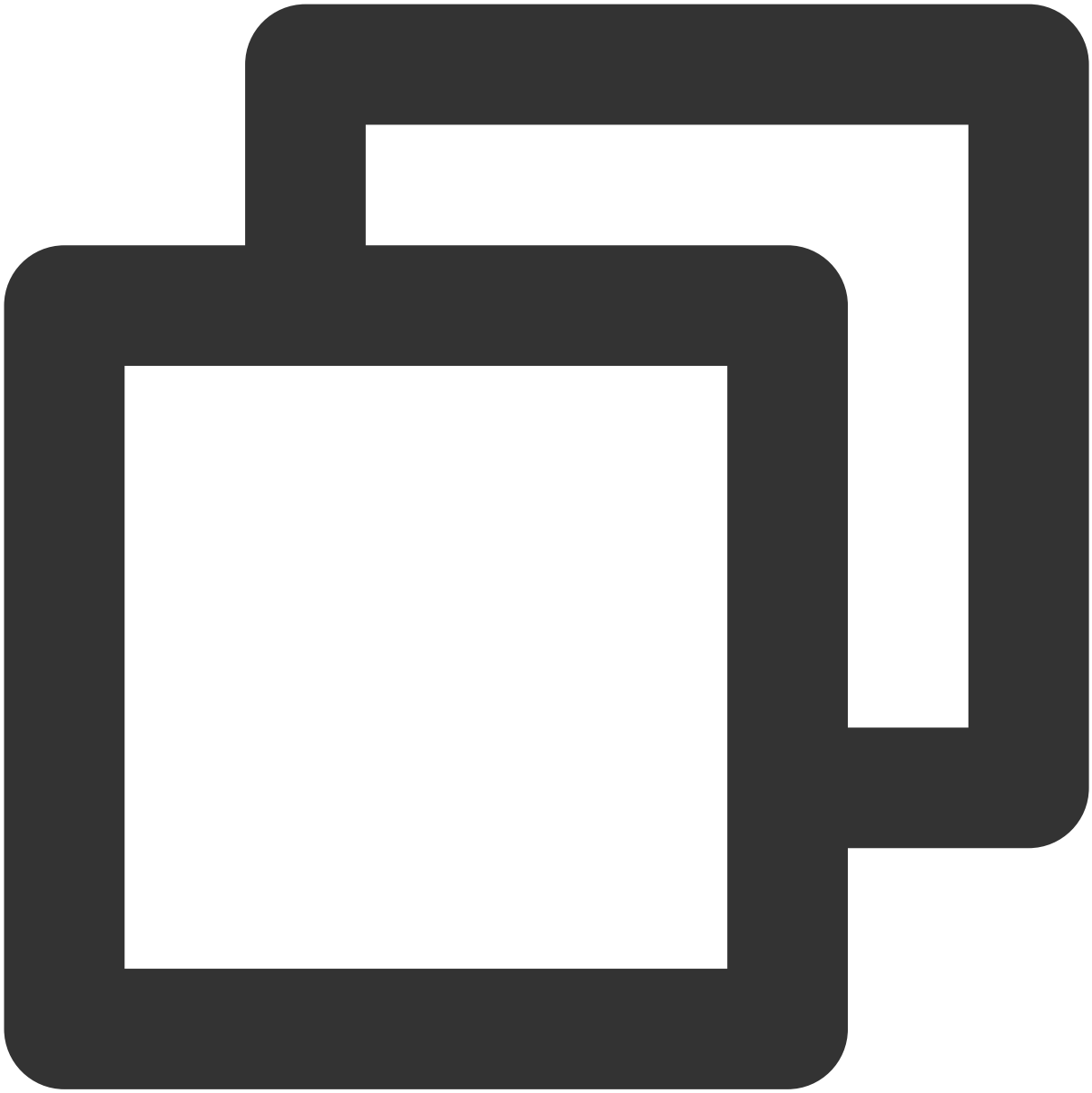
EXPLAIN format=tree query works better than the traditional EXPLAIN.

Option 2: Viewing in the thread list

The result of the `show processlist` command displays which threads are running. You can view not only the total number of current connections but also the connection status to identify abnormal query statements.

Based on the `show processlist` command, TDSQL-C for MySQL offers the proprietary `show parallel processlist` statement, which displays only the threads related to parallel query and filters out irrelevant threads.

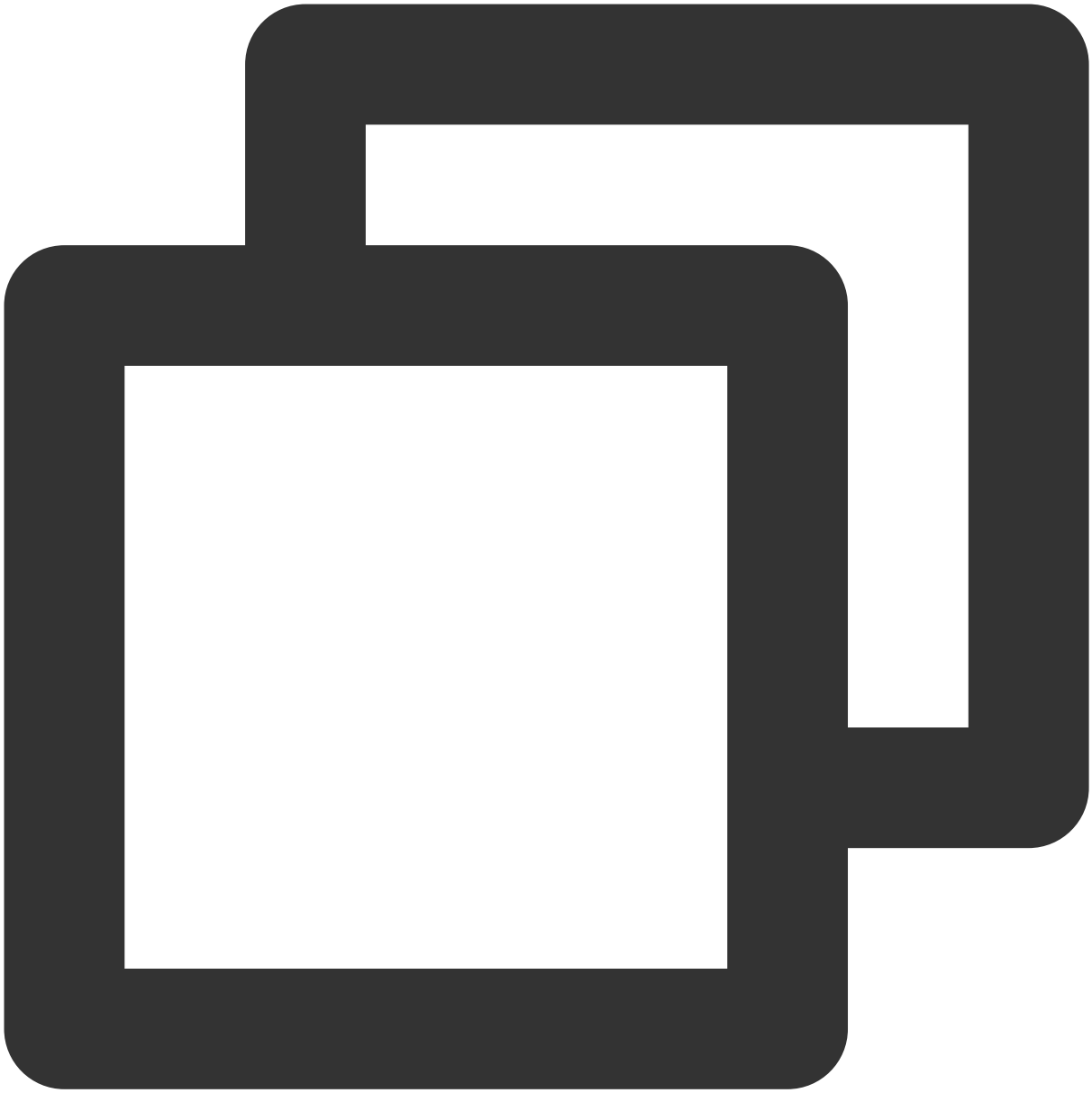
Sample SQL statement:



```
SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

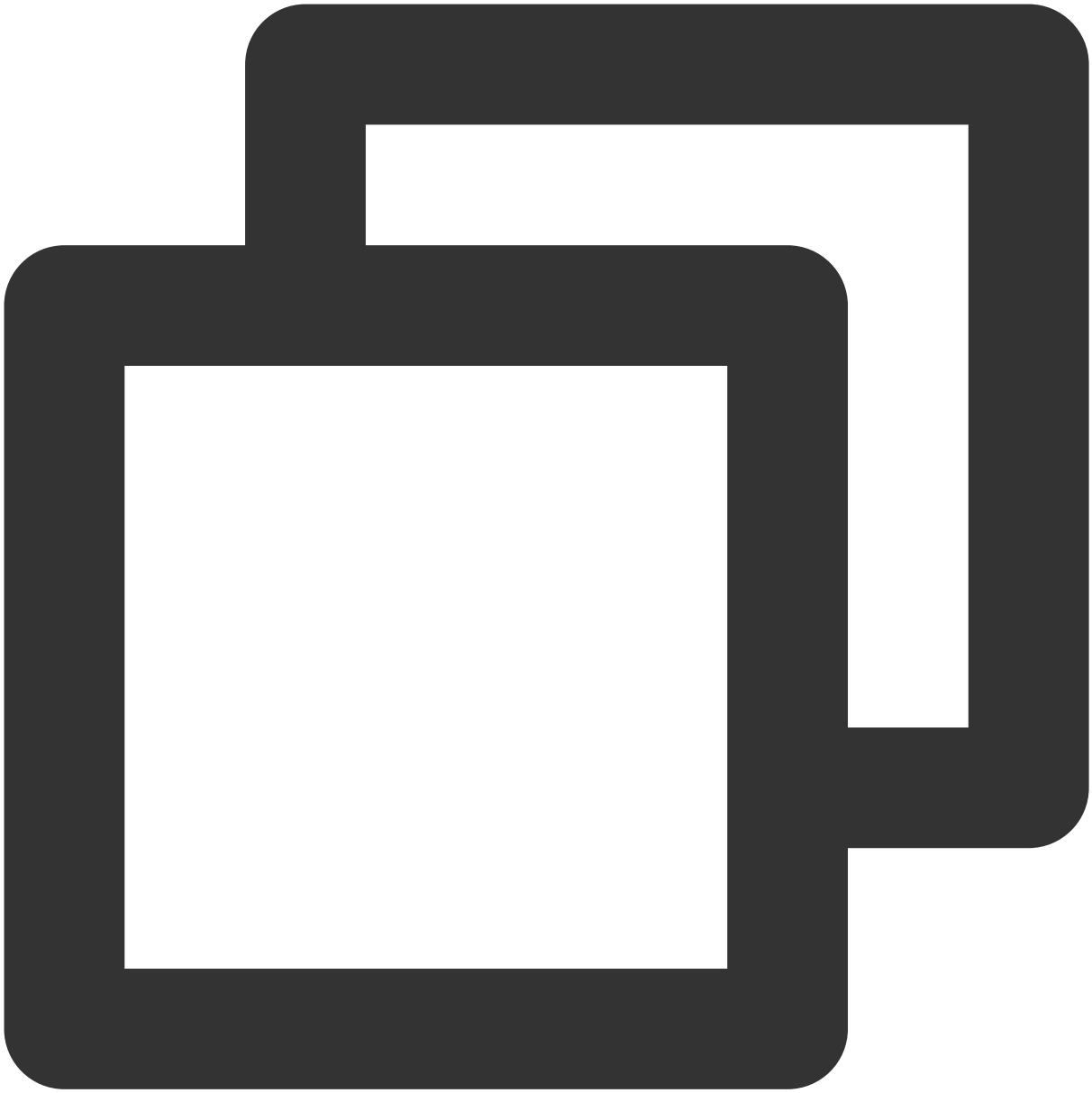
This sample is a simplified version of TPC-H Q1, a typical report operation.

`show processlist` query result:



```
mysql> show processlist;
+--------+-------------+-----------------+-----------+---------+-------+-----------
| Id     | User        | Host            | db        | Command | Time  | State
+--------+-------------+-----------------+-----------+---------+-------+-----------
|      7 | tencentroot | 127.0.0.1:49238 | NULL      | Sleep   |     0 |
|     11 | tencentroot | 127.0.0.1:49262 | NULL      | Sleep   |     0 |
|     13 | tencentroot | 127.0.0.1:49288 | NULL      | Sleep   |     1 |
| 237062 | tencentroot | localhost       | tpch100g  | Query   |    24 | Scheduling
| 237107 | tencentroot | localhost       | NULL      | Query   |     0 | init
+--------+-------------+-----------------+-----------+---------+-------+-----------
6 rows in set (0.00 sec)
```

`show parallel processlist` query result:



```
mysql> show parallel processlist;
+--------+-------------+-----------+----------+---------+------+-------------+-----
| Id     | User        | Host      | db       | Command | Time | State       | Info
+--------+-------------+-----------+----------+---------+------+-------------+-----
| 237062 | tencentroot | localhost | tpch100g | Query   |   18 | Scheduling  | SELE
| 237110 |             |           |          | Task    |   18 | Task runing | conn
| 237111 |             |           |          | Task    |   18 | Task runing | conn
| 237112 |             |           |          | Task    |   18 | Task runing | conn
| 237113 |             |           |          | Task    |   18 | Task runing | conn
```

```
+--------+-------------+-----------+----------+---------+------+-------------+-----
5 rows in set (0.00 sec)
```

As can be seen from the above result:

The parallel query plan assigns queries to four worker threads. There is only one data item in the user thread (ID: 237062). The SQL statement is pushed down to four worker threads. As indicated in `info`, all these four threads are executing `task 1`.

Each thread can be identified and located precisely.

Compared to `show processlist`, `show parallel processlist` can precisely find all running threads of parallel query and will not be affected by other threads.

References

Enabling/Disabling Parallel Query

HINT Statement Control

Parallel Query Metrics

Parallel Query Metrics

Last updated：2024-07-31 10:03:33

This document describes the monitoring metrics offered by TDSQL-C for MySQL to help you monitor the operations related to parallel database query in real time.

Prerequisites

Database version: TDSQL-C for MySQL 8.0 on kernel version 3.1.8 or later.

Parallel Query of Monitoring Metrics

Metric	Parameter	Unit	Description	Data Aggregation Method
Parallelly Queried Threads	txsql_parallel_threads_currently_used	-	The number of threads currently used for parallel queries.	MAX
Failed Parallel Queries	txsql_parallel_stmt_error	-	The number of parallel query statements that report errors.	SUM
Executed Parallel Queries	txsql_parallel_stmt_executed	-	The number of executed parallel query statements.	SUM
Parallel-to-Serial Queries	txsql_parallel_stmt_fallback	-	The number of parallel-to-serial query statements.	SUM

Practical Tutorial for Monitoring Metrics

The following lists some special issues for which you should check the metrics and their solutions.

Example 1:

The value of **Parallelly Queried Threads** is always equal to the value set by `txsql_max_parallel_worker_threads` .

Solution:

This example indicates that the threads are always fully loaded. We recommend you increase the value of the `txsql_max_parallel_worker_threads` parameter when the CPU utilization is low.

Example 2:

The value of **Failed Parallel Queries** increases.

Solution:

Check the CPU utilization and memory utilization. If the load becomes significantly higher, we recommend you reduce the value of `txsql_parallel_degree` to 1/4 of the number of CPU cores.

Example 3:

The value of **Parallel-to-Serial Queries** increases.

Solution:

This example indicates that the currently executed SQL statement does not meet the conditions for parallel query. We recommend you increase the total number of threads (`txsql_max_parallel_worker_threads`) or the maximum memory (`txsql_optimizer_context_max_mem_size`) of the parallel query plan environment that a single statement can apply for, so as to ensure that parallel query can be used for SQL statements.

Performance Test Report

Last updated：2023-01-06 11:35:30

This document describes the performance test report of TDSQL-C for MySQL parallel query.

Standard performance test report

Dataset: TPC-H

Tested data volume: 100 GB

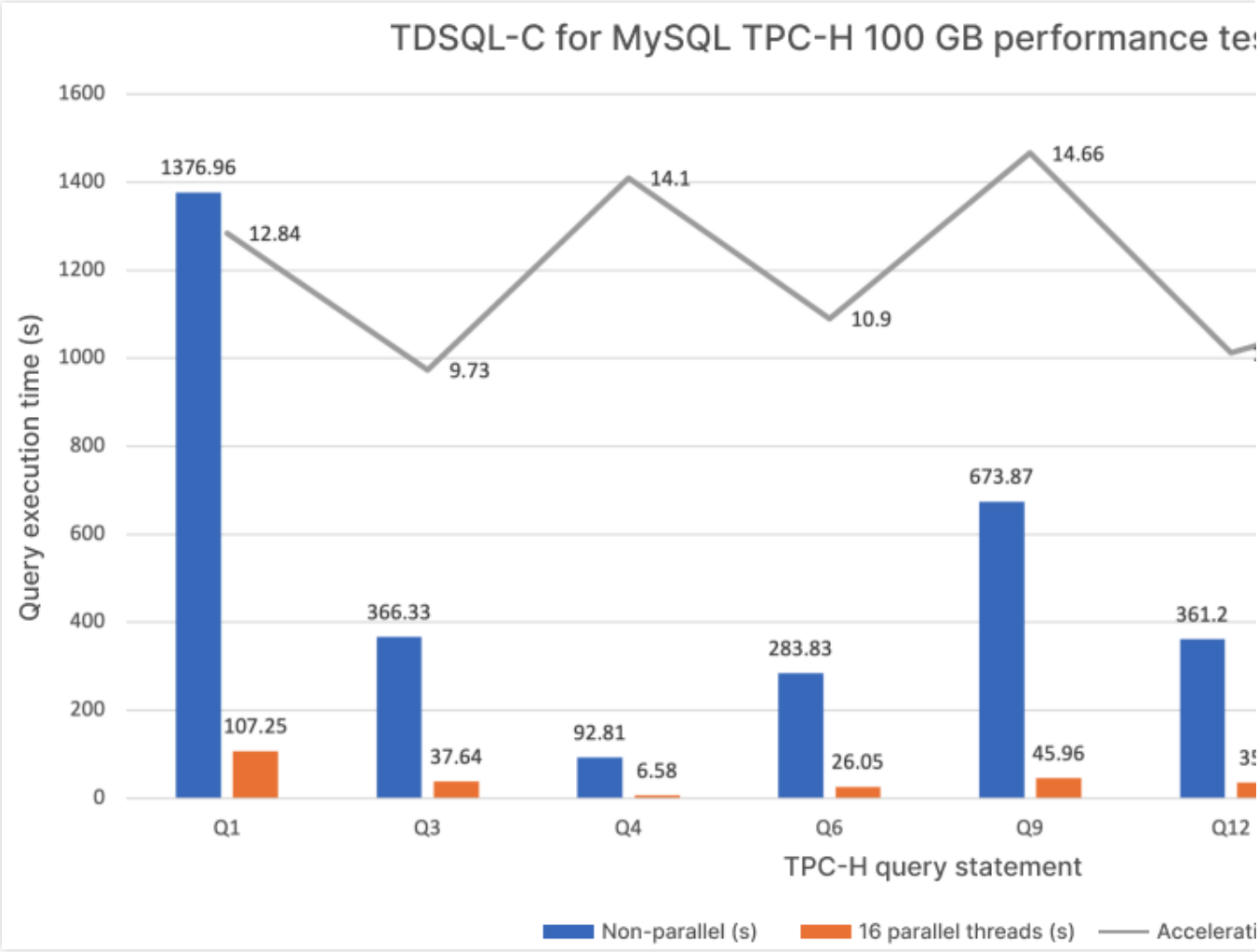
Parallel threads: 16

Query: Q1, Q3, Q4, Q6, Q9, Q12, Q14, and Q19 performance acceleration ratio

Execution timetable

Query	Non-Parallel Execution Time (s)	16-Thread Parallel Query Time (s)	Acceleration Ratio
Q1	1,376.96	197.25	12.84
Q3	366.33	37.64	9.73
Q4	92.81	6.58	14.1
Q6	283.83	26.05	10.9
Q9	673.87	45.96	14.66
Q12	361.2	35.7	10.12
Q14	65.02	5.7	11.41
Q19	20.55	1.87	11.0

Execution time acceleration ratio



Conclusion

The parallel query capability leverages multiple compute cores to greatly shorten the response time of large queries.

Database Security and Encryption
Access and Authorization
Overview

Last updated：2023-11-09 15:05:36

Issues

If you have multiple users managing different Tencent Cloud services such as CVM, VPC, and TencentDB, and they all share your Tencent Cloud account access key, you may face the following problems:
Your key will be easily compromised because it is shared by several users.
You cannot restrict the access from other users and your service will be vulnerable to the security risks caused by their maloperations.

Solutions

In this case, you can use sub-accounts to enable different users to manage different services to avoid such problems. By default, sub-accounts don't have the permissions to use cloud services or related resources. Therefore, you need to create policies to allow sub-accounts to have the permissions they need.
Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access to your Tencent Cloud resources. With CAM, you can create, manage terminate users (groups) and use identity and policy management to control what users can use what Tencent Cloud resources.
You can use CAM to bind a user or user group to a policy which allows or denies them access to specified resources to complete specified tasks. For more information on CAM, see Element Reference and Policies.
You can skip this section if you don't need to manage permissions to TencentDB resources for sub-accounts. Doing so doesn't affect your understanding and use of the rest of the documentation.

Getting started

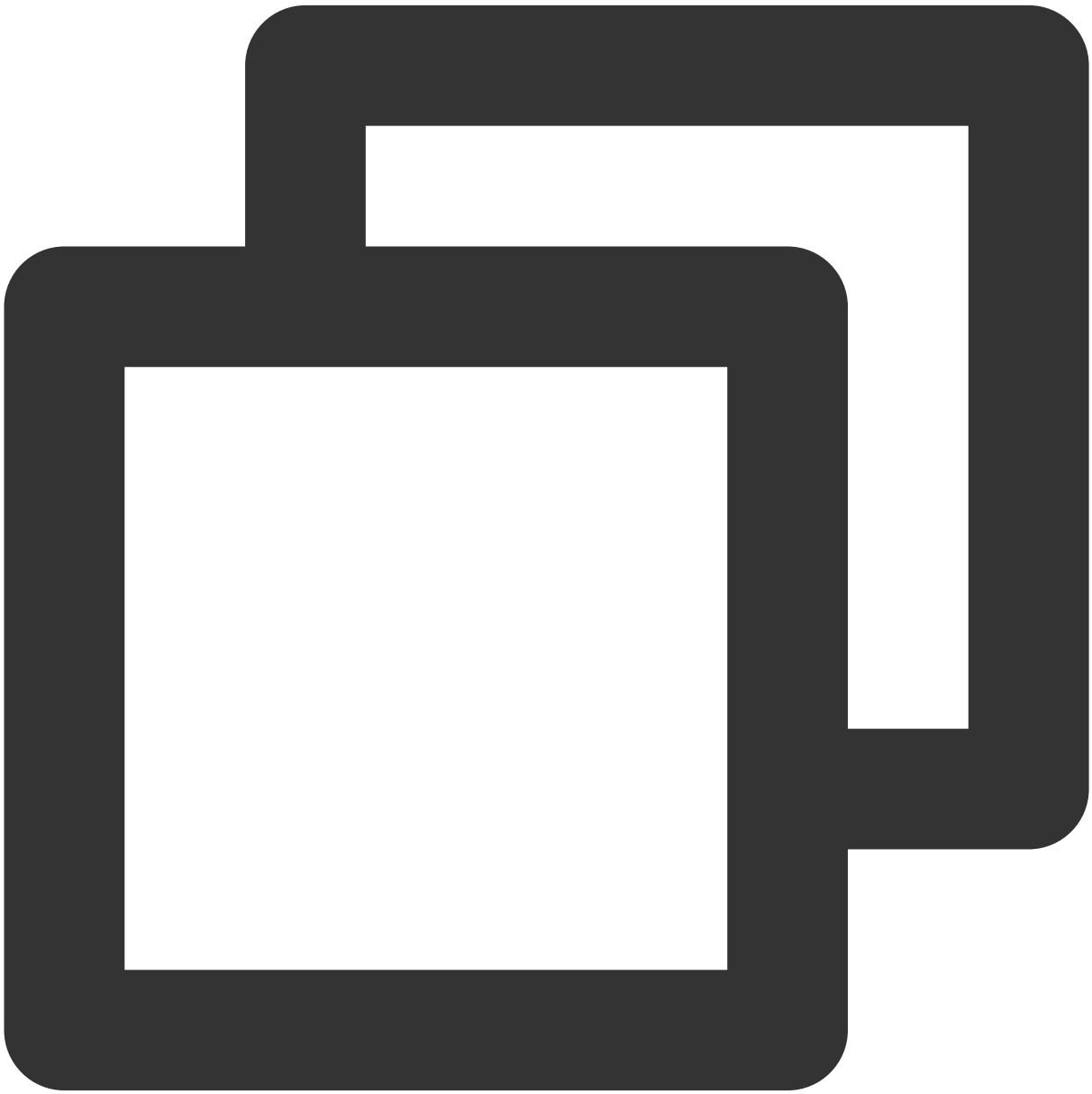
A CAM policy must authorize or deny the use of one or more TDSQL-C for MySQL operations. At the same time, it must specify the resources that can be used for the operations (which can be all resources or partial resources for certain operations). A policy can also include the conditions set for the manipulated resources.
Note:
CAM policies are preferred over projects for managing resources and authorizing operations of TDSQL-C for MySQL, even though the user experience for current users with project-based permissions remains unaffected.
Effectiveness conditions cannot be set for TDSQL-C for MySQL for the time being.

Relevant Information	Link
Basic policy structure	Authorization Policy Syntax > CAM policy syntax
Operation definition in a policy	Authorization Policy Syntax > Action
Resource definition in a policy	Authorization Policy Syntax > Resource path
Resource-level permissions	Authorizable Resource Types

Authorization Policy Syntax

Last updated：2023-11-01 17:22:52

CAM policy syntax



```
{
        "version":"2.0",
```

```
          "statement":
          [
              {
                  "effect":"effect",
                  "action":["action"],
                  "resource":["resource"],
                   "condition": {"key":{"value"}}
              }
          ]
  }
```

version is required. Currently, only the value "2.0" is allowed.

statement describes the details of one or more permissions. This element contains a permission or permission set of other elements such as `effect` , `action` , `resource` , and `condition` . One policy has only one statement.

effect is required. It describes the result of a statement. The result can be "allow" or an "explicit deny".

action is required. It describes the allowed or denied operation. An operation can be an API or a feature set (a set of specific APIs prefixed with "permid").

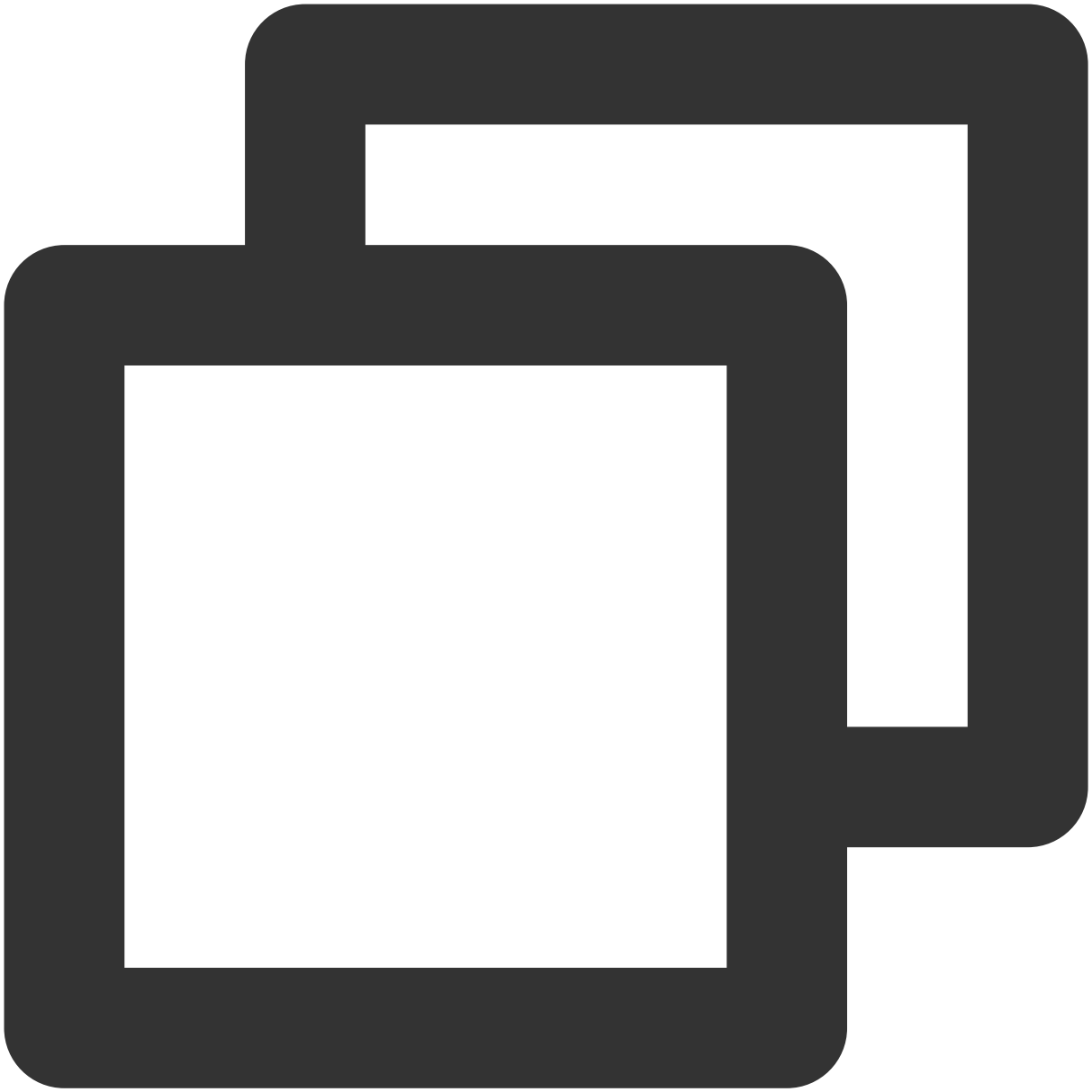
resource is required. It describes the details of authorization. A resource is described in a six-segment format. Detailed resource definitions vary by product.

condition is required. It describes the condition for the policy to take effect. A condition consists of operator, action key, and action value. A condition value may contain information such as time and IP address. Some services allow you to specify additional values in a condition.

Operations

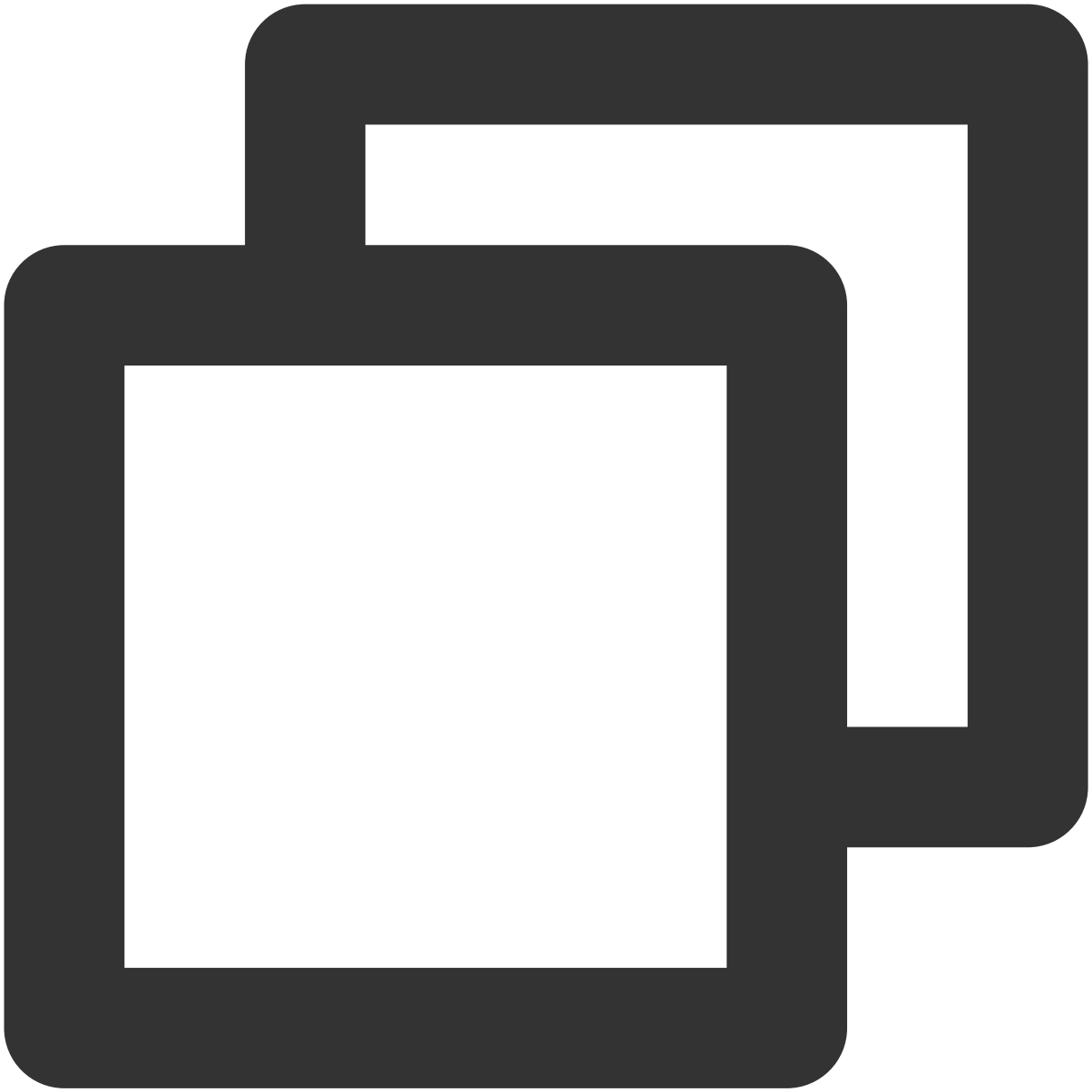
In a CAM policy statement, you can specify any API operation from any service that supports CAM. APIs prefixed with `cynosdb:` are used for TDSQL-C for MySQL, such as `cynosdb:DescribeClusters` or `cynosdb:ResetAccountPassword` .

To specify multiple operations in a single statement, separate them by comma.



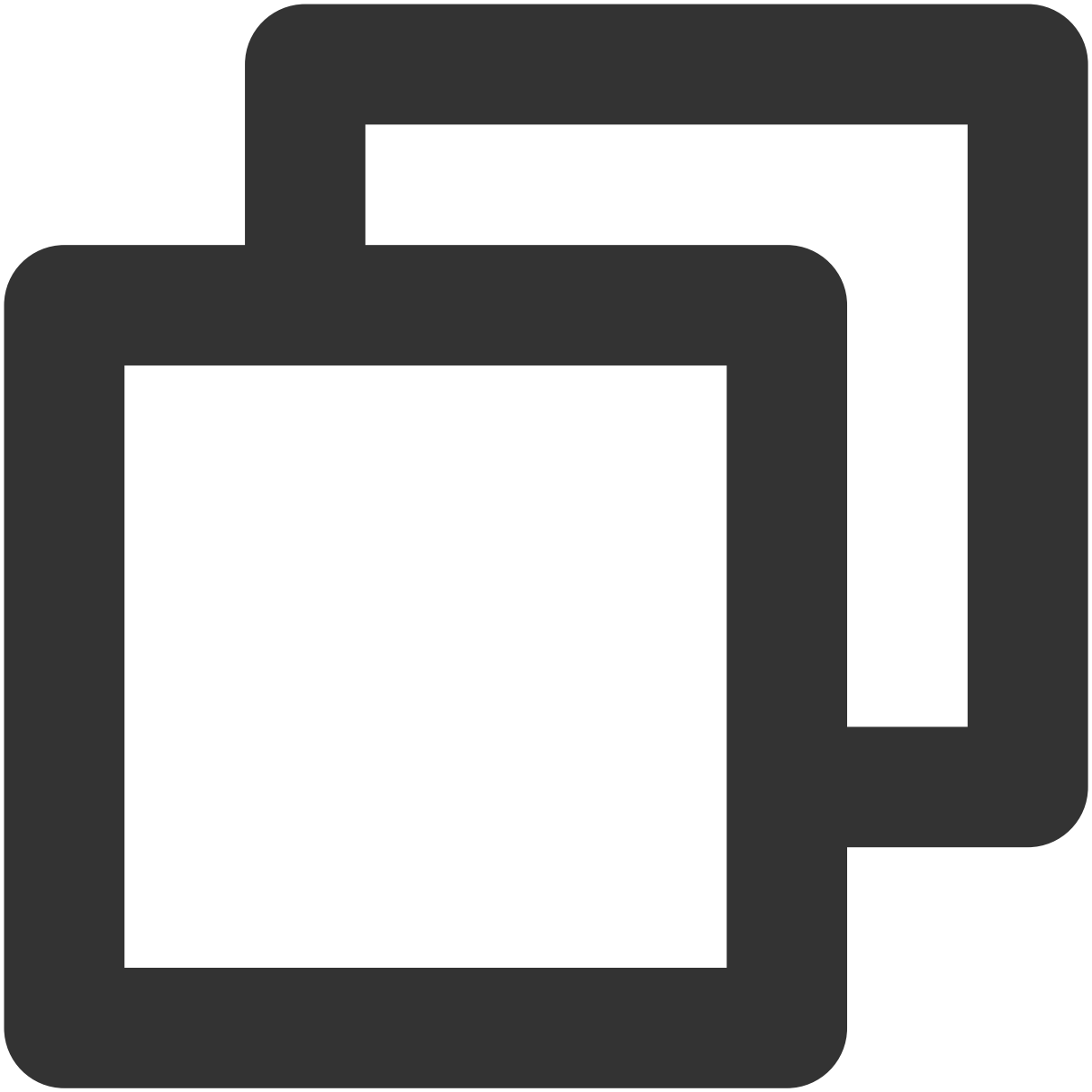
```
"action":["cynosdb:action1","cynosdb:action2"]
```

You can also specify multiple operations by using a wildcard. For example, you can specify all operations beginning with "Describe" in the name.



```
"action":["cynosdb:Describe*"]
```

If you want to specify all operations in TDSQL-C for MySQL, use the `*` wildcard.

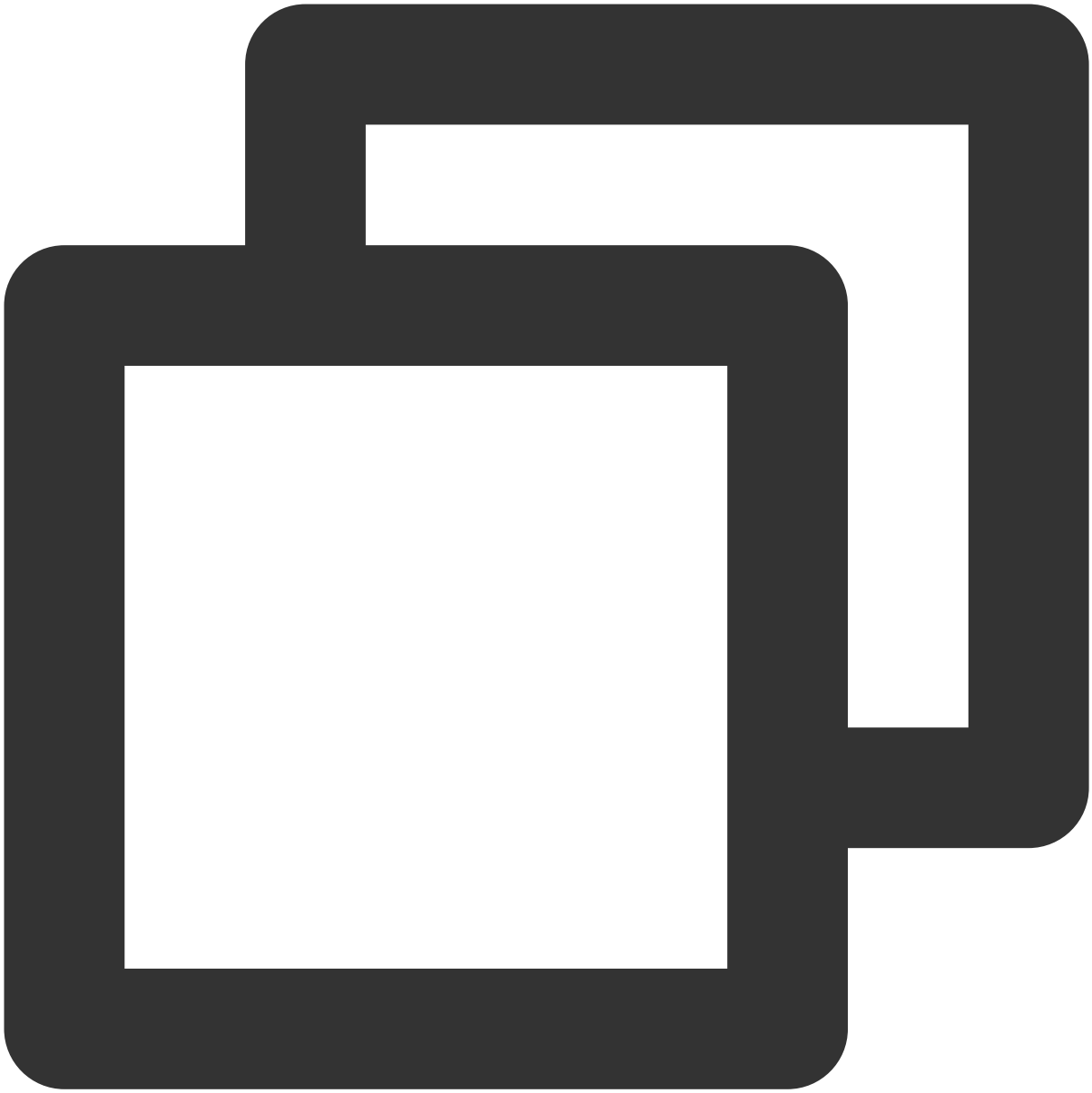


```
"action":["cynosdb:*"]
```

Resource path

Each CAM policy statement has its own applicable resources.

Resource paths are generally in the following format:



```
qcs:project_id:service_type:region:account:resource
```

project_id describes the project information, which is only used to enable compatibility with legacy CAM logic and can be left empty.

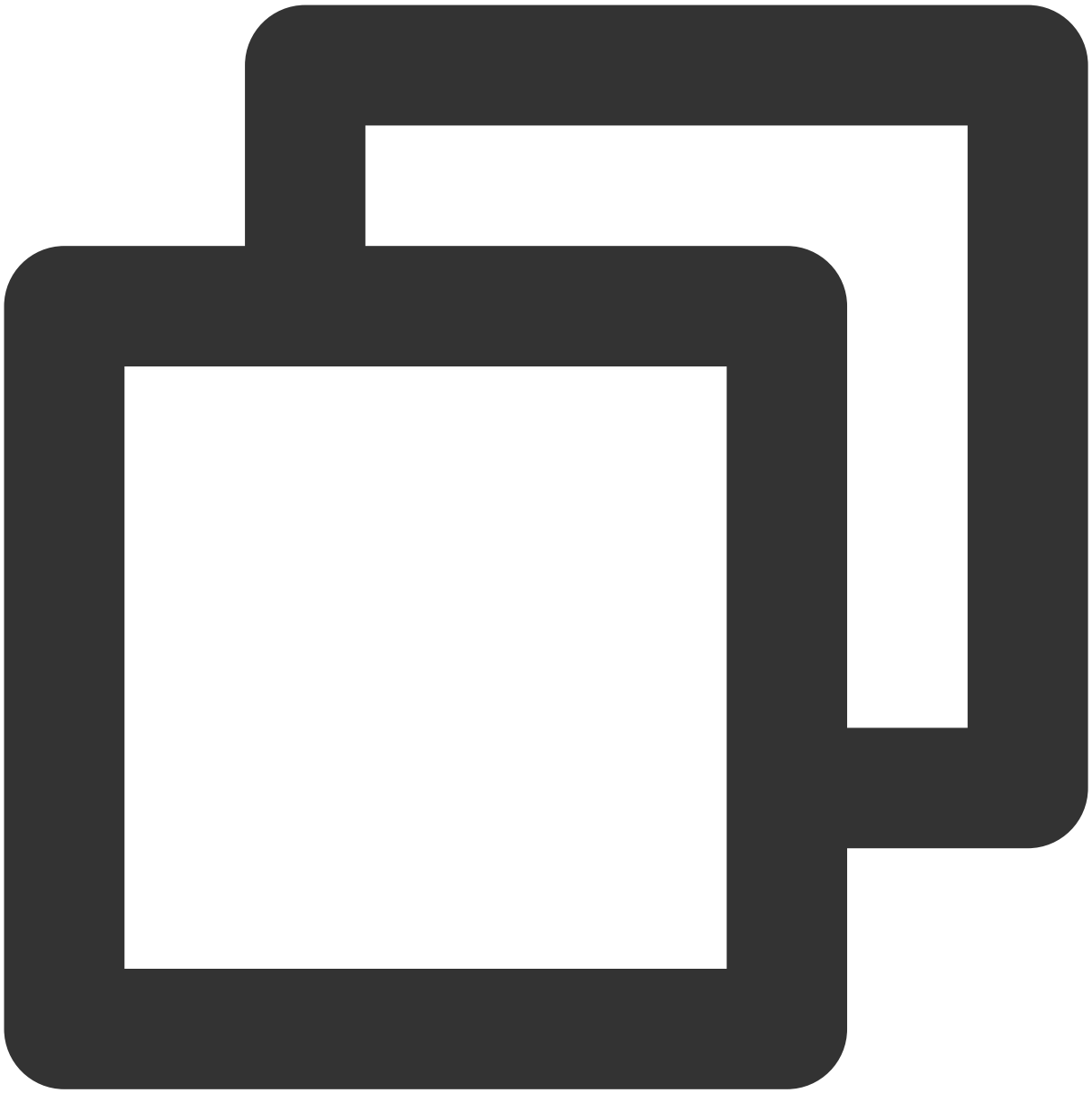
service_type describes the product abbreviation such as `cynosdb`.

region describes the region information, such as `bj`.

account describes the root account of the resource owner, such as `uin/12xxx8`.

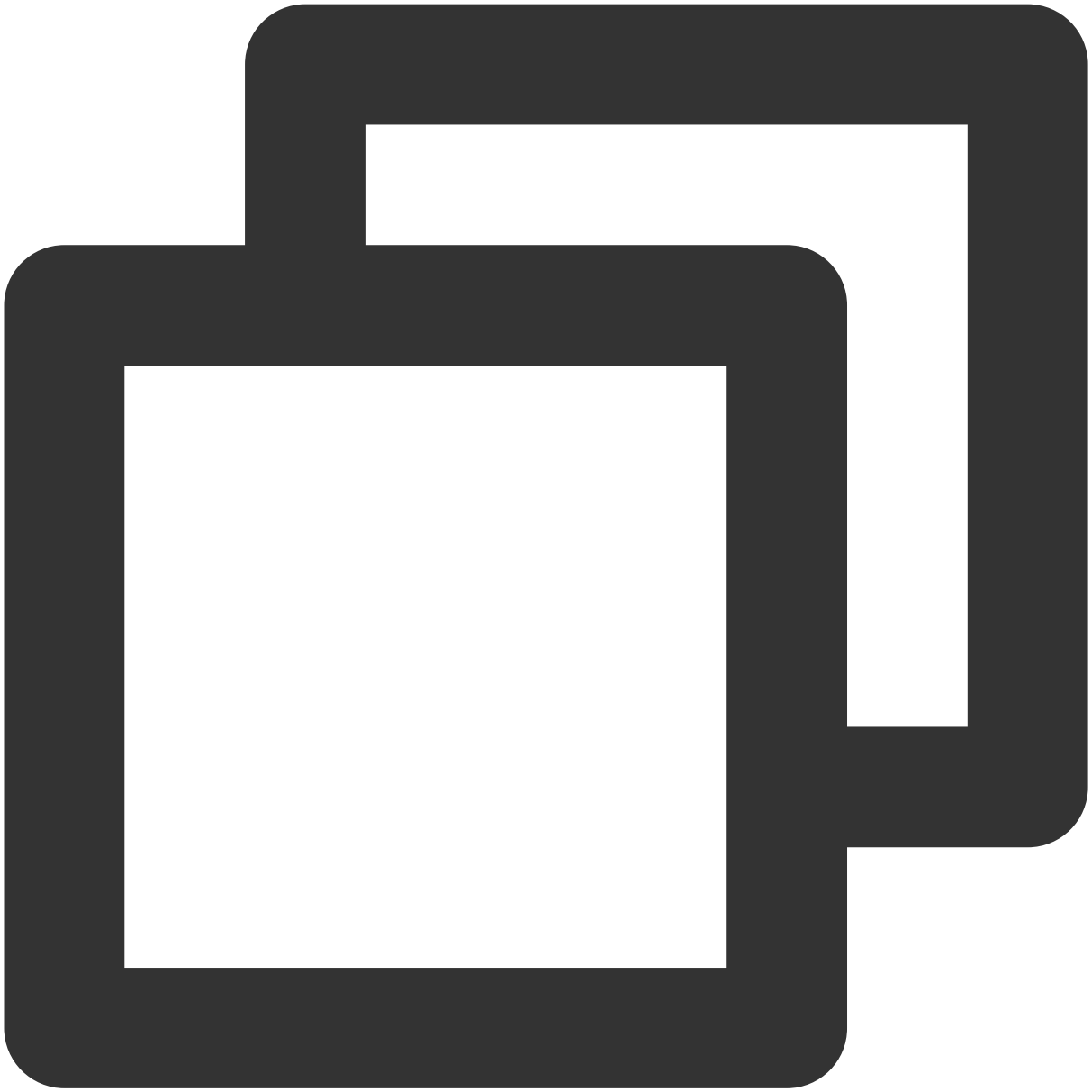
resource describes the detailed resource information of each product, such as `instance/clusterId` or `instance/*`.

For example, you can specify a resource for a specific cluster (cynosdbmysql-123abc) in a statement.



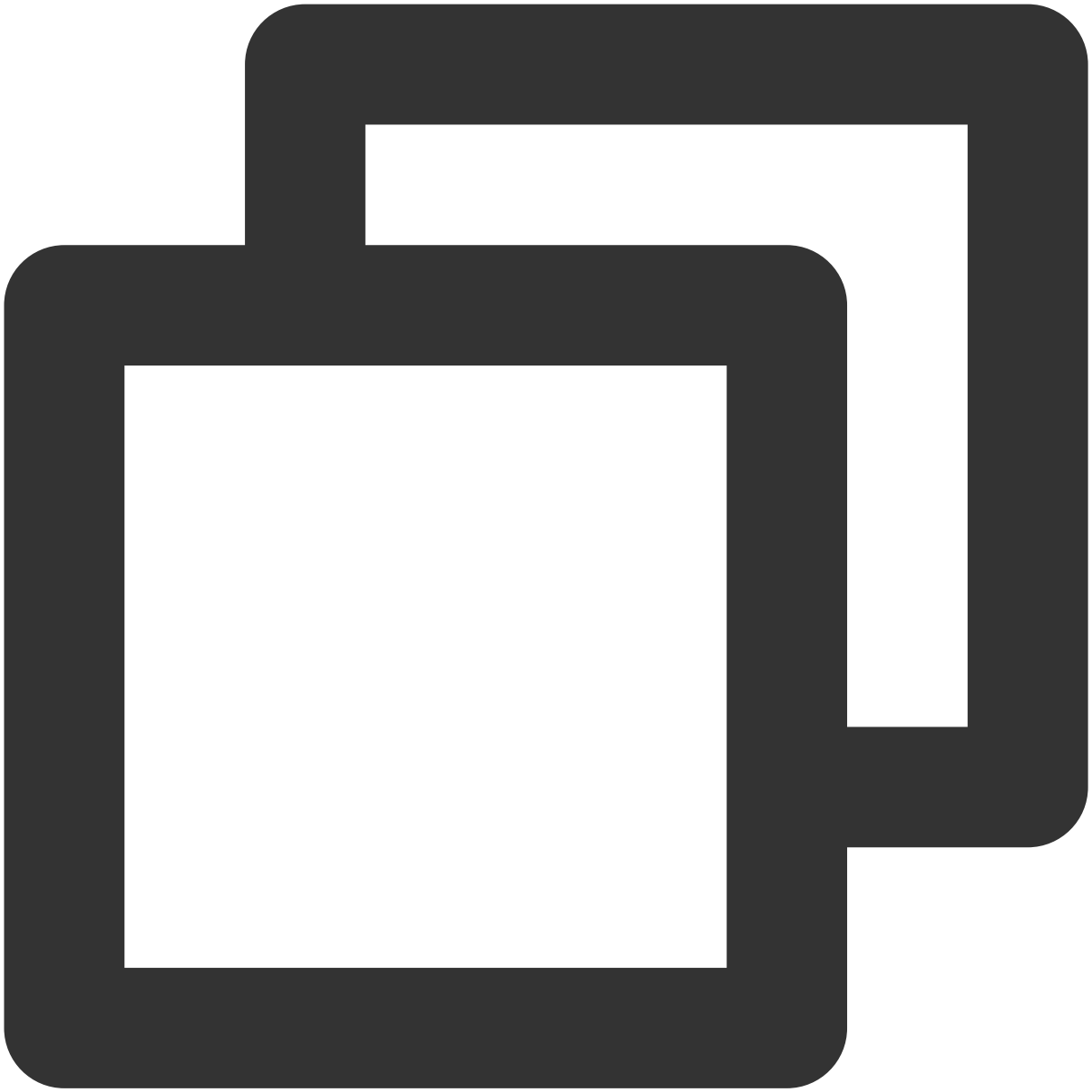
```
"resource":[ "qcs::cynosdb:bj:uin/12xxx8:instance/cynosdbmysql-123abc"]
```

You can also use the `*` wildcard to specify it for all clusters that belong to a specific account.



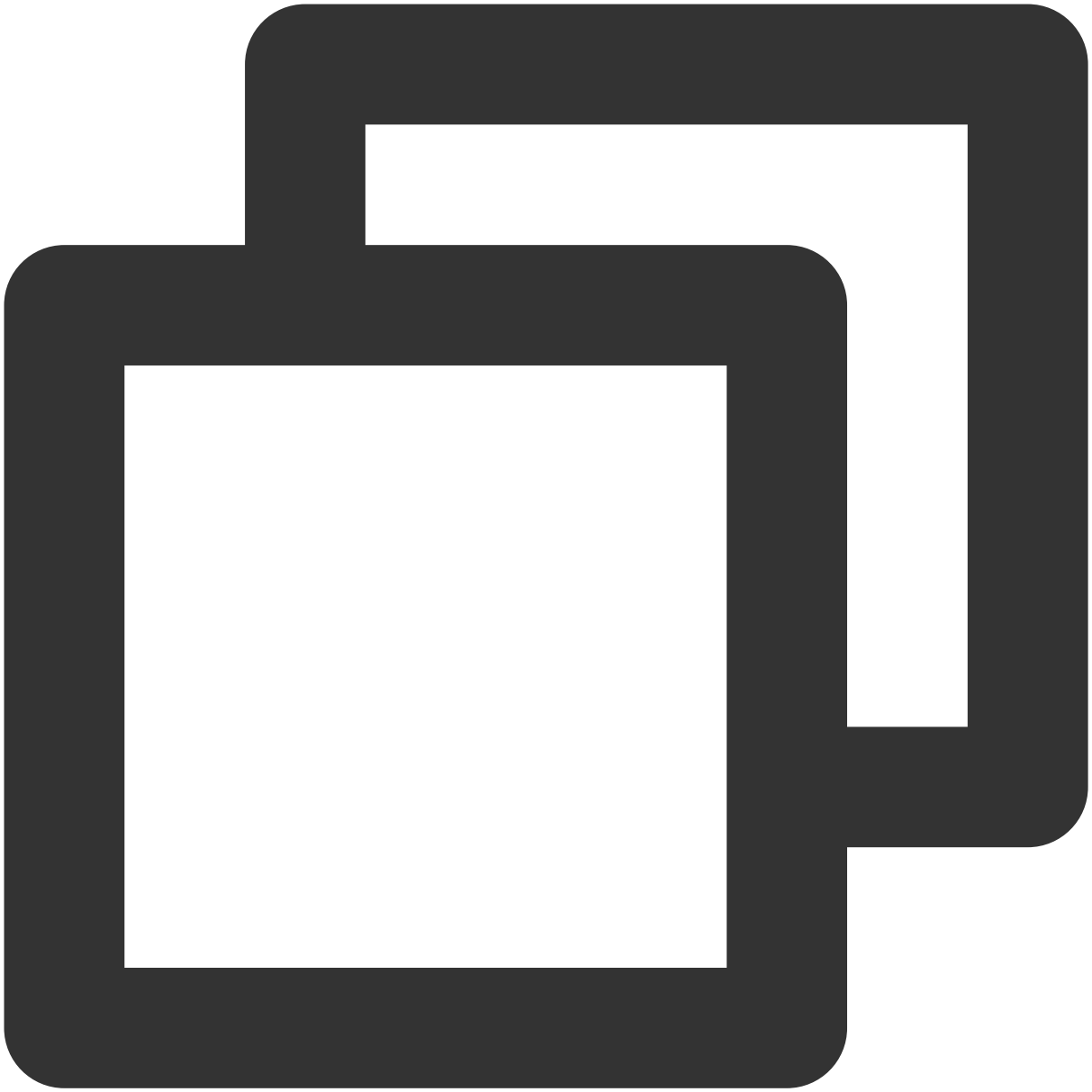
```
"resource":[ "qcs::cynosdb:bj:uin/12xxx8:instance/*"]
```

If you want to specify all resources or if a specific API operation does not support resource-level permission control, you can use the "*" wildcard in the `resource` element.



```
"resource": ["*"]
```

To specify multiple resources in one policy, separate them with commas. In the following example, two resources are specified:



```
"resource":["resource1","resource2"]
```

The table below describes the resources that can be used by TDSQL-C for MySQL and the corresponding resource description methods, where words prefixed with `$` are placeholders, `region` refers to a region, and `account` refers to an account ID.

Resource	Resource Description Method in Authorization Policy
Cluster	`qcs::cynosdb:$region:$account:instance/$clusterId`
VPC	`qcs::vpc:$region:$account:vpc/$vpcId`

Security group	`qcs::cvm:$region:$account:sg/$sgId`

Authorizable Resource Types

Last updated：2023-03-01 14:33:46

Resource-level permission can be used to specify which resources a user can manipulate. TDSQL-C for MySQL supports certain resource-level permissions. This means that for TDSQL-C for MySQL operations with resource-level permissions, you can control the time when a user is allowed to perform operations or to use specified resources. The following table describes the types of resources that can be authorized in CAM.

Resource Type	Resource Description Method in Authorization Policy
Cluster APIs	`qcs::cynosdb:$region::instance/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`

The table below lists the TDSQL-C for MySQL API operations which currently support resource-level permission control as well as the resources supported by each operation. When specifying a resource path, you can use the `*` wildcard in the path.

Note:

Any TencentDB API operation not listed in the table does not support resource-level permission. For such an operation, you can still authorize a user to perform it, but you must specify `*` as the resource element in the policy statement.

Cluster APIs

API Operation	Resource Path
DescribeBackupConfig	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeBackupList	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeRollbackTimeRange	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeRollbackTimeValidity	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
ModifyBackupConfig	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
ActivateCluster	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeClusterDetail	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
IsolateCluster	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
ModifyClusterName	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`

ModifyClusterProject	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
OfflineCluster	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DeleteAccounts	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeAccounts	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ModifyAccountDescription	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ResetAccountPassword	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeClusterInstanceGrps	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ActivateInstance	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeInstanceDetail	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
IsolateInstance	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
UpgradeInstance	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ModifyInstanceName	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
OfflineInstance	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeClusterAddr	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeClusterNetService	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeClusterParams	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeClusterServerInfo	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeErrorLogs	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeMaintainPeriod	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeSlowLogs	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ModifyClusterParam	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ModifyMaintainPeriodConfig	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
DescribeDBSecurityGroups	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
ModifyDBInstanceSecurityGroups	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos
CloseWan	qcs::cynosdb:$region:$account:instanceId/*	qcs::cynos

OpenWan	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeClusters	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeInstances	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`
DescribeIsolatedInstances	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynos`

Security Group Management Introduction

Last updated：2023-01-13 14:44:14

This document describes the concepts and descriptions of security groups.

What is a security group?

A [security group] (https://www.tencentcloud.com/document/product/213/12452) is a stateful virtual firewall capable of filtering. As an important means for network security isolation provided by Tencent Cloud, it can be used to set network access controls for instances of CVM, CLB and TencentDB. Instances with the same network security isolation demands in one region can be put into the same security group, which is a logical group. TencentDB and CVM share the security group list.

Security Group Rules and Limits

For more information on security group rule, see Security Group.
For use limits and quotas of security groups, see Use Limits > Security Group Limits.

Security Group Description

TencentDB-C for MySQL security groups currently only support network access control for VPCs and public networks but not the classic network.
Security groups associated with TencentDB instances in the Frankfurt, Silicon Valley, and Singapore regions currently do not support public network access control.
As TencentDB doesn't have any active outbound traffic, outbound rules don't apply to it.
TDSQL-C for MySQL security groups support read-write and read-only instances.

TDSQL-C for MySQL Security Group Operations

Creating and Managing TencentDB Security Groups
Modifying/Adding Security Group Rules
Configuring Security Groups for TencentDB。

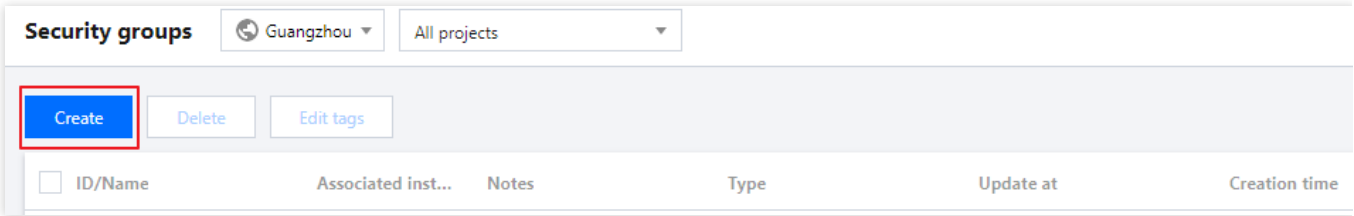
TencentDB Security Group Management Creating and Managing TencentDB Security Groups

Last updated：2023-11-09 16:18:17

This document describes how to create and manage security groups for TDSQL-C for MySQL cluster.

Creating a security group

1. Log in to the VPC console.
2. Select **Security Group** on the left sidebar, select a region at the top, and click **Create**.



3. In the pop-up dialog window, configure the following items, and click **OK**.

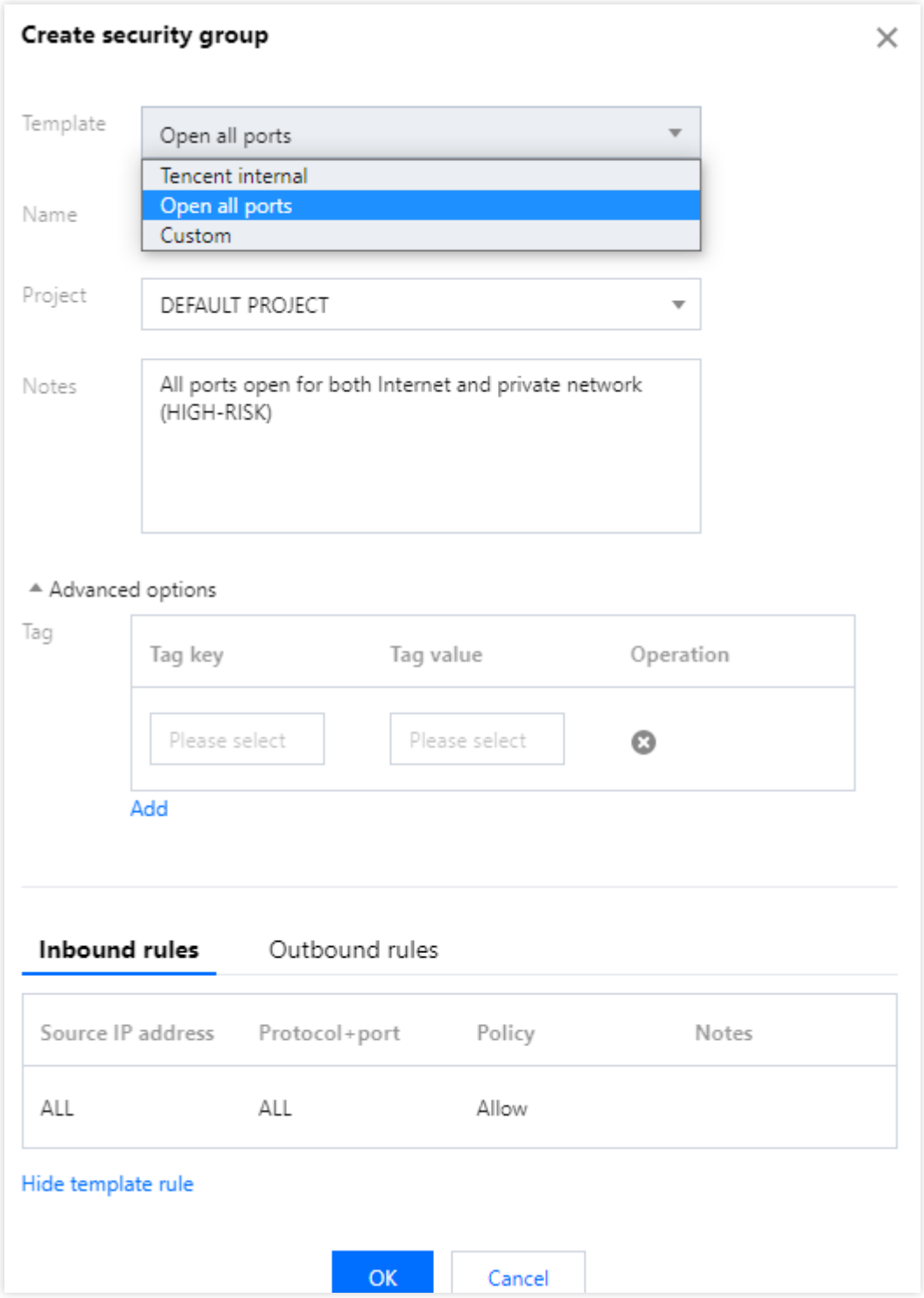


Table 1:

Parameter	Description
Template	Select a template based on the service to be deployed on the TencentDB instance in the security group, which simplifies the security group rule configuration, as shown in the table 2.
Name	Custom name of the security group.

Project	"DEFAULT PROJECT" is selected by default. You can select another project for easier management.
Remark	A short description of the security group.
Advanced options	You can add tags as needed for the security group in **Advanced options**, and no tag is selected by default. For details, see Tag Overview.
Display template rule	If the template is selected, you can click **Display template rule** to view the rule details.

Table 2:

Template	Description	Applicable Scenario
Open all ports	All ports are opened to the public and private networks by default. This may pose security issues.	-
Open ports 22, 80, 443, and 3389 and the ICMP protocol	Ports 22, 80, 443, and 3389 and the ICMP protocol are opened to the public network by default. All ports are opened to the private network.	This template doesn't take effect for TencentDB.
Custom	You can create a security group and then add custom rules. For more information, see Adding Security Group Rules.	-

Managing a security group

After the security group is created, you can perform management operations for security groups. For detailed directions, see the following steps:

Viewing Security Groups

Removing from Security Groups

Cloning Security Groups

Deleting Security Groups

Modifying/Adding Security Group Rules

Last updated：2024-03-07 10:40:57

Security groups are used to manage traffic to and from public and private networks. For the sake of security, most inbound traffic is denied by default, but you can modify and add security group rules as needed. This document describes how to modify and add security group rules in the console.

Prerequisites

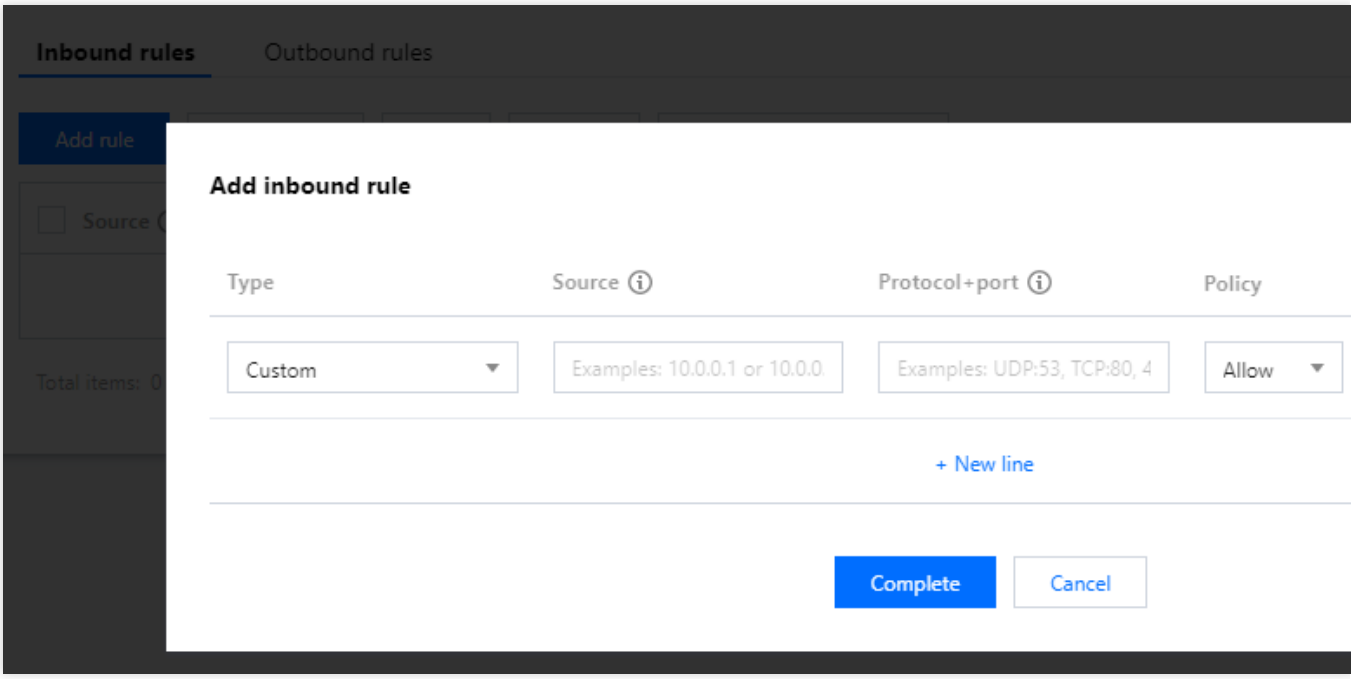
You have created a security group. For more information, see Configuring Security Group .
You have learned about which traffic is allowed or denied for your CVM instance. For more information on security group rules and their use cases, see [Security Group Use Cases]
(https://www.tencentcloud.com/document/product/213/32369).

Modifying Security Group Rules

1. Log in to the CVM console and go to the Security Group page, find the created target security group, and click **Modify Rule** in the **Operation** column.
2. On the security group rule page, select inbound or outbound rules by clicking **Inbound rules** or **Outbound rules**.
3. Locate the desired rule and click **Edit** to modify it.

Adding Security Group Rules

1. Log in to the CVM console and go to the Security Group page, find the created target security group, and click **Modify Rule** in the **Operation** column.
2.
On
the security group rule page, click **Inbound Rules** > **Add Rule**.
3. In the pop-up dialog box, set the rule.



Type: **Custom** is selected by default. You can also choose another system rule template, such as **Windows Login** template, **Linux Login** template, **Ping** template, **HTTP(80)** template, **HTTPS(443)** template, **MySQL(3306)** template, and **SQLServer(1433)** template.

Source or **Destination**: traffic source (inbound rules) or target (outbound rules). You need to specify one of the following options:

Source	Specification
IP address or CIDR IP range	Utilize CIDR notation (IPV4; such as 203.0.113.0, 203.0.113.0/24 or 0.0.0.0/0, where 0.0.0.0/0 signifies a match for all IPv4 addresses. IPV6: such as FF05::B5, FF05:B5::/60, ::/0 or 0::0/0, where ::/0 or 0::0/0 denotes a match for all IPv6 addresses).
Parameter template - IP address	Refer to the IP address object in the Parameter Template.
Parameter template - IP address group	Refer to the IP address group object in the Parameter Template.
Security group	To refer to a security group ID, you may refer to the ID of the following security groups: Current security group: This refers to the security group ID associated with the cloud database, presented in markdown format. Other security groups: This refers to another security group ID within the same project in the same region. **Note：** Referring to the security group ID is an advanced feature that you may opt to use. The rules of the referred security group will not be appended to the current security group.

	When security group rules are configured, if a security group ID is entered in the source/destination, it signifies that only the private IP addresses of the cloud database instances and elastic network interfaces bound to this security group ID are considered as the source/destination, excluding public IP addresses.

Note：

The /**number** following the IP address denotes the subnet mask, of which the **number** signifies the length of the network portion in the subnet mask. For instance, 192.168.0.0/24 represents a network segment. The subnet mask /**24** indicates that the first 24 bits of 192.168.0.0 are the network portion, and the remaining 8 bits are the host portion. Hence, within the 192.168.0.0/24 subnet, the assignable host IP range is: 192.168.0.0 - 192.168.0.255.

Protocol Port:Specify the protocol type and port range. The protocol type supports TCP, UDP, ICMP, ICMPv6, and GRE. You can also refer to the protocol ports or protocol port groups in the Parameter Template.

Note:

To connect to TDSQL-C for MySQL, you need to open its instance port. You can log in to the TDSQL-C for MySQL console, click the cluster ID to enter the details page, and view its port in the **Connection Info** section.

The security group rules displayed on the **Security Group** page in the TDSQL-C MySQL console take effect for private and public (if enabled) network addresses of the TencentDB for MySQL instance.

The supported formats for the protocol port are as follows:

A single port, such as `TCP:80` .

Multiple discrete ports, such as `TCP:80,443` .

Consecutive ports, such as `TCP:3306-20000` .

All the ports, such as `TCP:ALL` .

Policy: **Allow** or **Reject**. **Allow** is selected by default.

Allow: Traffic to this port is allowed.

Reject: Data packets will be discarded without any response.

Notes: A short description of the rule for easier management.

4.

Click **Complete**

5. On the security group rule page, click **Outbound rule** and refer to Steps 2-4 for adding an outbound rule.

Matching Security Group for TencentDB

Last updated：2023-08-24 09:59:53

A security group is a firewall provided by Tencent Cloud for controlling inbound traffic of TencentDB. You can associate a security group with a cluster when purchasing it or later in the console.
This document describes how to configure a security group for TDSQL-C for MySQL in the console.

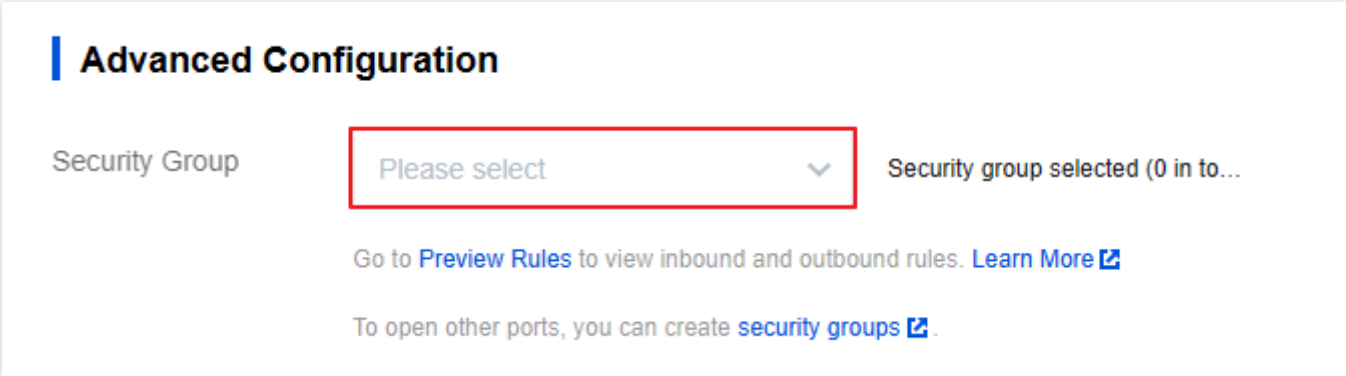
Note:

Currently, security groups can be configured only for TDSQL-C for MySQL instances in VPCs.
TDSQL-C for MySQL allows you to configure different security groups for the read-write and read-only addresses respectively, which don't affect each other.

Configuring a security group

During cluster purchase

1. Log in to the purchase page.
2. Configure the database and basic information, configure the security group in **Advanced Configuration**, and click **Buy Now**.



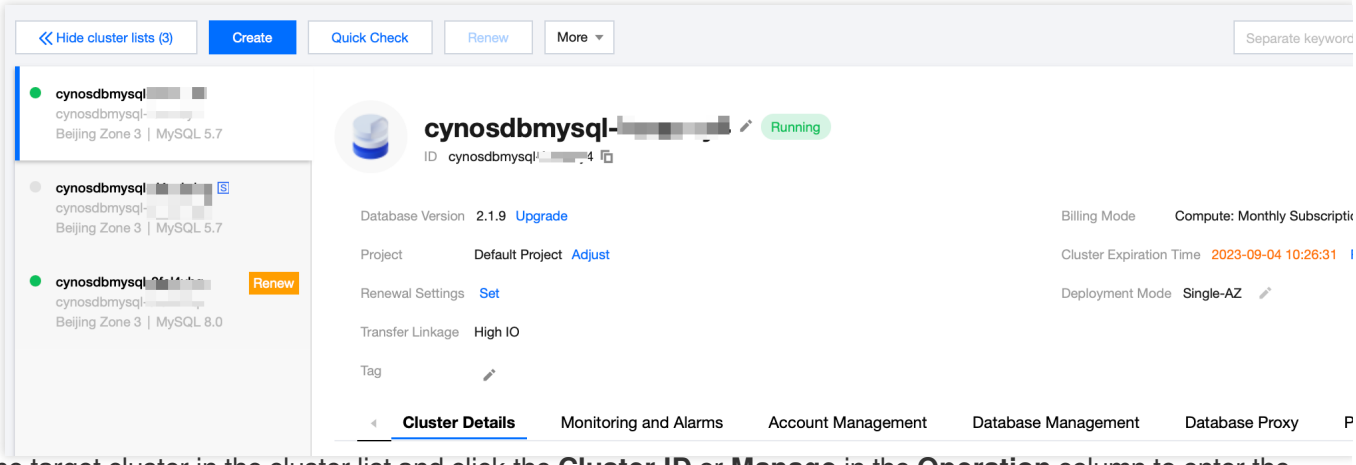
In the console

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode:

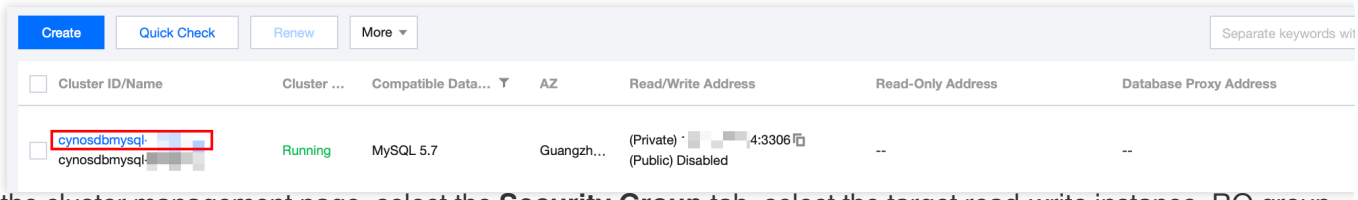
Tab view

List view

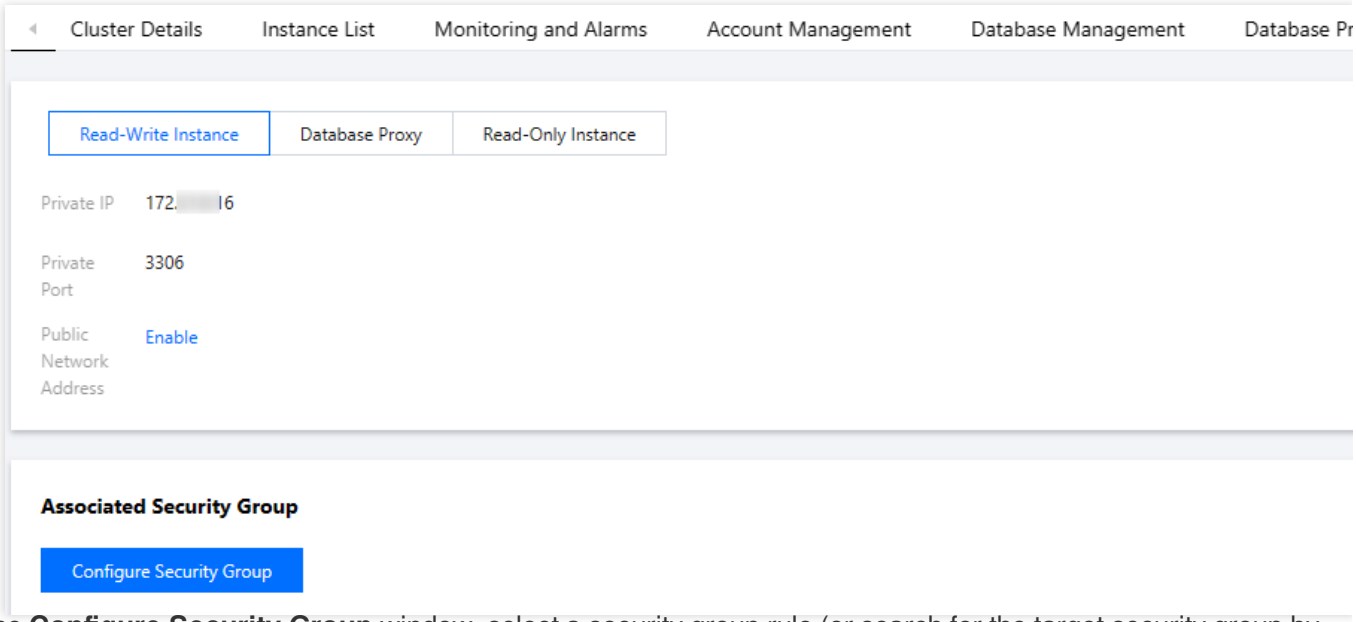
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



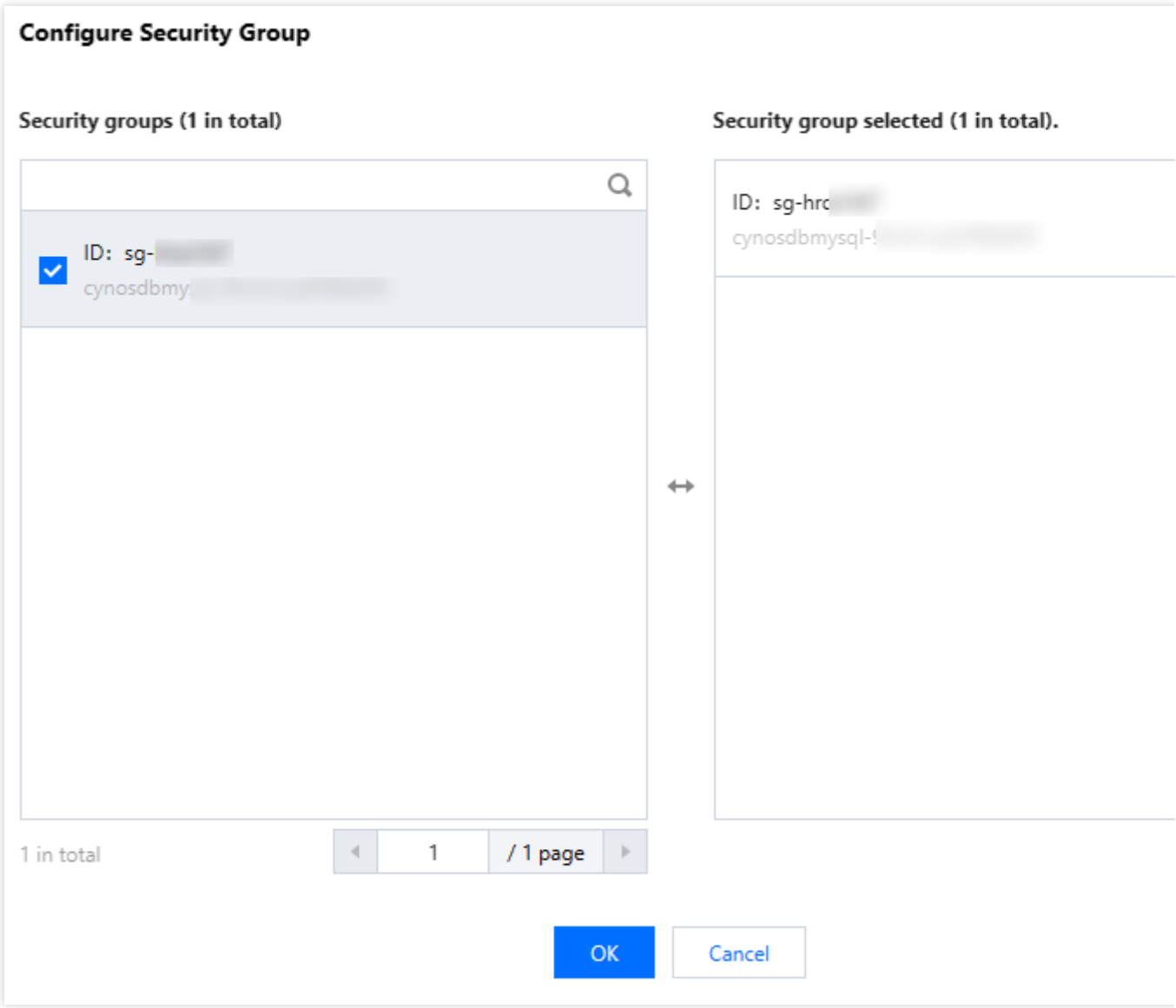
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. On the cluster management page, select the **Security Group** tab, select the target read-write instance, RO group, read-only instance, or database proxy, and click **Configure Security Group**.



4. In the **Configure Security Group** window, select a security group rule (or search for the target security group by ID) and click **OK**.



Note:

You can configure up to five security group rules.

You can configure different access addresses with different security groups, which will control only sources that access the current address.

Adjusting the priority of a security group

If multiple security groups are bound to a TDSQL-C for MySQL instance, they will be executed based on their priorities such as 1 and 2. You can adjust the priorities as follows.

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster in the cluster list, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select the **Security Group** tab, select the target read-write instance, database proxy, or read-only instance, and click **Edit** below **Associated Security Group**.

4. In the **Security Group** column, adjust the priority of the security group by using the moveup/movedown/deletion icons  and click **Save**.

Encryption
Setting SSL Encryption

Last updated：2024-08-19 15:53:55

SSL Encryption Overview

Secure Sockets Layer (SSL) authentication is a process that authenticates the connection from the client to the TencentDB server. After SSL encryption is enabled, you can get a CA certificate and upload it to the server. Then, when the client accesses the database, the SSL protocol will be activated to establish an SSL secure channel between the client and the server to realize encrypted data transfer, prevent data from being intercepted, tampered with, and eavesdropped during the transfer, and ultimately ensure the data security for both the client and the server. The SSL protocol needs to be established based on the OSI Model - Transport Layer (TCP). Its advantage is that it is independent from application layer protocols; therefore, high-level application layer protocols such as HTTP, FTP, and TELNET can be transparently established on it. The SSL protocol completes encryption algorithm processing, communication key negotiation, and server authentication before communication is made through application layer protocols. After that, all data transferred through application layer protocols will be encrypted to ensure communication privacy.

Background

When you connect to a database in an unencrypted manner, all information transferred over the network will be in plaintext and may be eavesdropped, tampered with, and impersonated by illegal users. The SSL protocols are designed to address these risks and can bring the following benefits theoretically:
Information is encrypted and cannot be eavesdropped by a third party.
There is a verification mechanism for immediate tampering detection by both parties in the communication.
Identity certificates will be used to authenticate the identity.
TDSQL-C for MySQL supports enhancing link security by enabling SSL encryption and allows downloading and installing SSL CA certificates to the required application services.

Note:
SSL encryption protects the traffic between the database and the server rather than the data itself. Encrypting the network connection at the transport layer can improve the security and integrity of the communication data, but will increase the response time of the network connection.

Prerequisites

The instance version is MySQL 5.7/8.0.
The instance form can be either pre-configured resources or Serverless.

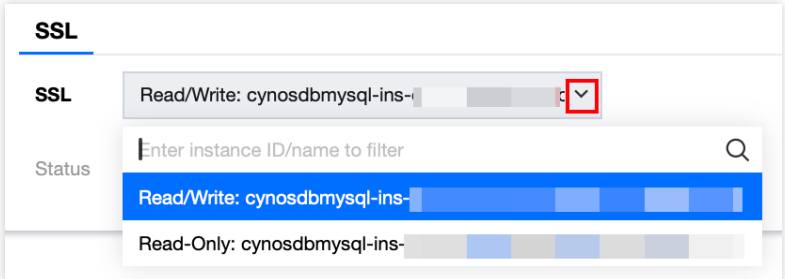
Supported Versions

TDSQL-C for MySQL uses OpenSSL to implement secure connections. TDSQL-C for MySQL supports Transport Layer Security (TLS) versions 1.0, 1.1, 1.2, and 1.3. TLS support depends on the MySQL version. The table below shows the MySQL versions that support TLS.

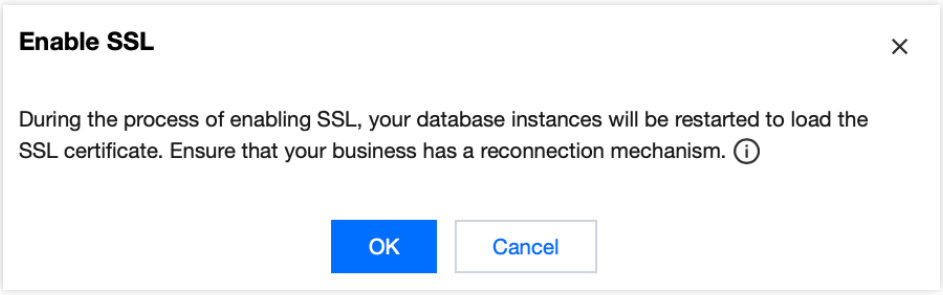
MySQL Version	TLS1.0	TLS1.1	TLS1.2	TLS1.3
MySQL 5.7	Not supported	Not supported	Supported	Supported
MySQL 8.0	Supported	Supported	Supported	Supported

Enabling SSL Encryption

1. Log in to the TDSQL-C for MySQL console, and click the cluster ID in the cluster list to enter the cluster management page.
2. On the cluster management page, select the **Data Security** tab, and under **SSL**, select the read-write or read-only instance that requires enabling SSL encryption.



3. This feature is disabled by default. Toggle the switch to on, and then click **OK** to enable SSL encryption.



Note:

During the process of enabling SSL encryption, your database instance will be restarted to load the SSL certificate. Make sure that your business has a reconnection mechanism.

4. Click **Download** to download the SSL CA certificate, which is valid for 20 years.

The downloaded file is a compressed package (TencentDB-SSL-CA.zip), containing the following three files:

.p7b file: It is used to import the CA certificate into Windows.

.jks file: It is a truststore certificate storage file in Java with a unified password 'tencentdb', used to import the CA certificate chain into the Java program.

.pem file: It is used to import the CA certificate into other systems or applications.

Configuring an SSL CA Certificate

After enabling SSL encryption, you need to configure an SSL CA certificate when using a client to connect to TencentDB. The following takes Navicat as an example to describe how to install an SSL CA certificate. For other applications or clients, see their respective instructions.

Note:

Each time SSL encryption is enabled or disabled, a new certificate will be generated.

1. Open Navicat.

2. Right-click the target database and select **Edit Connection**.

3. Select the SSL tab, and choose the path for the CA certificate in .pem format. After completing the settings, click **OK**.

Note:

If you encounter the 'connection is being used' error, it may be caused by a previous session that has not been disconnected. Close Navicat and try again.

4. Double-click the target database to test whether the connection is normal.

Disabling SSL Encryption

1. Log in to the TDSQL-C for MySQL console, and click the cluster ID in the cluster list to enter the cluster management page.

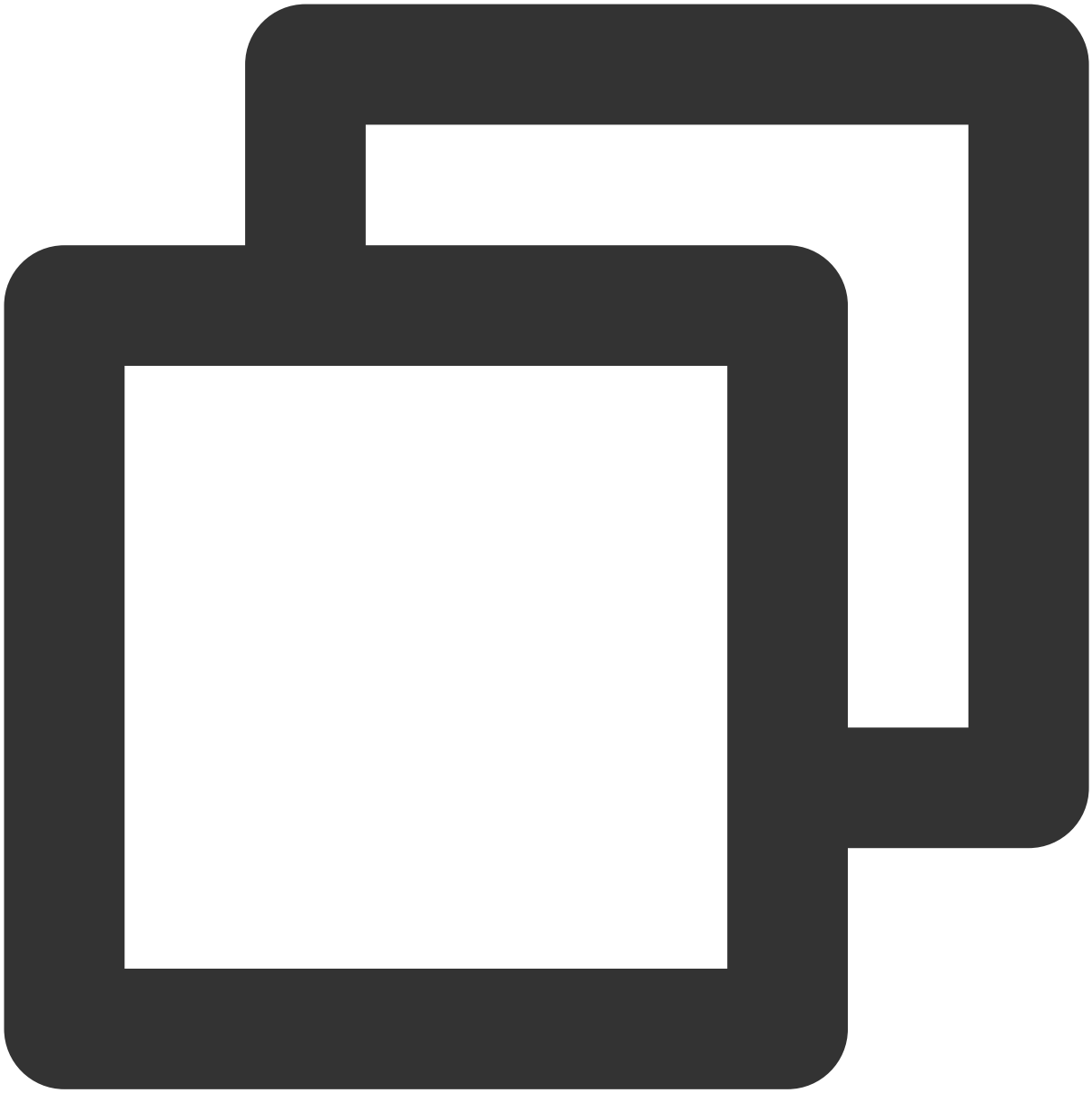
2. On the cluster management page, select the **Data Security** tab, and under **SSL**, select the read-write or read-only instance that requires disabling SSL encryption.
3. Toggle the switch after **Status** to off, and in the pop-up prompt box, click **OK**.

Note:

During the process of disabling SSL encryption, your database instance will be restarted to uninstall the SSL certificate. Make sure that your business has a reconnection mechanism.

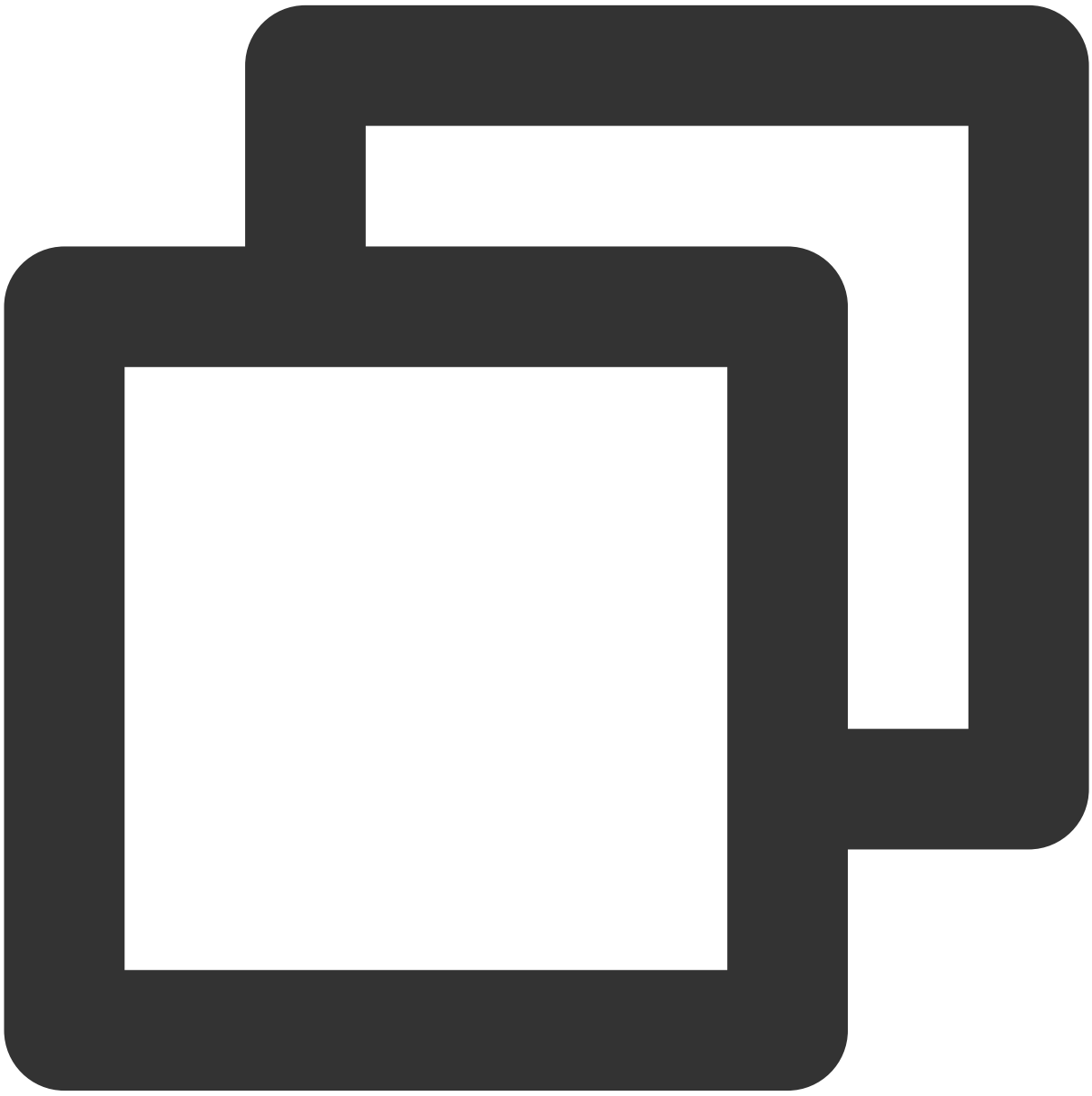
Connecting to an Instance with SSL Encryption Enabled Using the MySQL Command Line Client

If you are using a different version of the database, the connection command parameters of the MySQL client will vary. You can check the version of the database you are using with the following command, and then refer to the subsequent steps to connect to the instance.



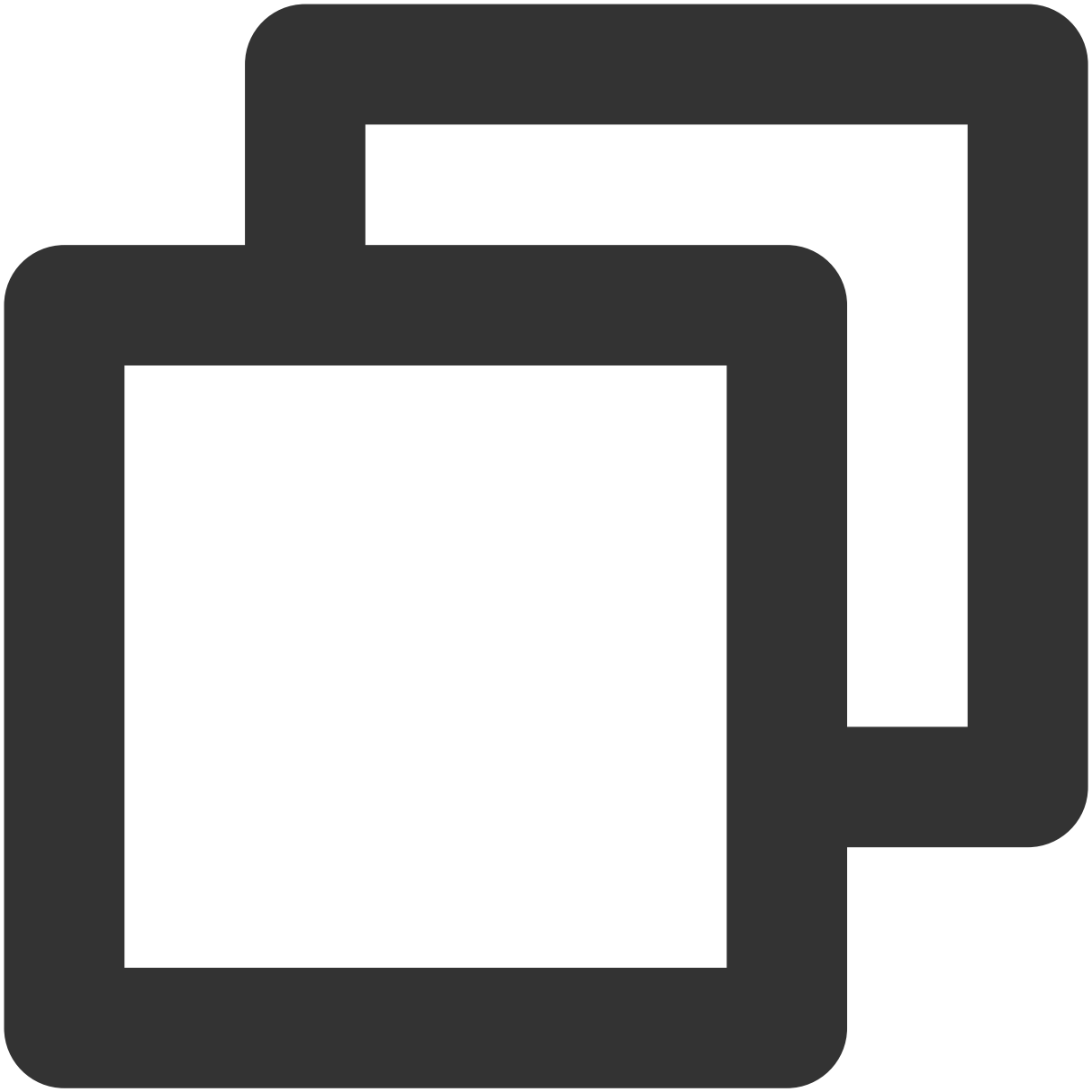
```
SELECT VERSION();

Query result example:
+--------------+
| VERSION()    |
+--------------+
| 8.0.30-txsql |
+--------------+
1 row in set (0.00 sec)
```

1. Download the SSL CA certificate via the TDSQL-C for MySQL console. For the operation, see Enabling SSL Encryption.
2. Use the MySQL Command Line Client to connect to an instance with SSL encryption enabled through command. For commands when the client-side database version is MySQL 5.7/8.0, connect to an instance using the following command.



```
mysql -h <IP address> --ssl-ca=<CA certificate path> --ssl-mode=REQUIRED -P <Port n
```

If you want to use another SSL mode, such as VERIFY_CA or VERIFY_IDENTITY, connect to an instance using the following command.



```
mysql -h <IP address> --ssl-ca=<CA certificate path> --ssl-mode=VERIFY_CA -P <Port
```

Note:

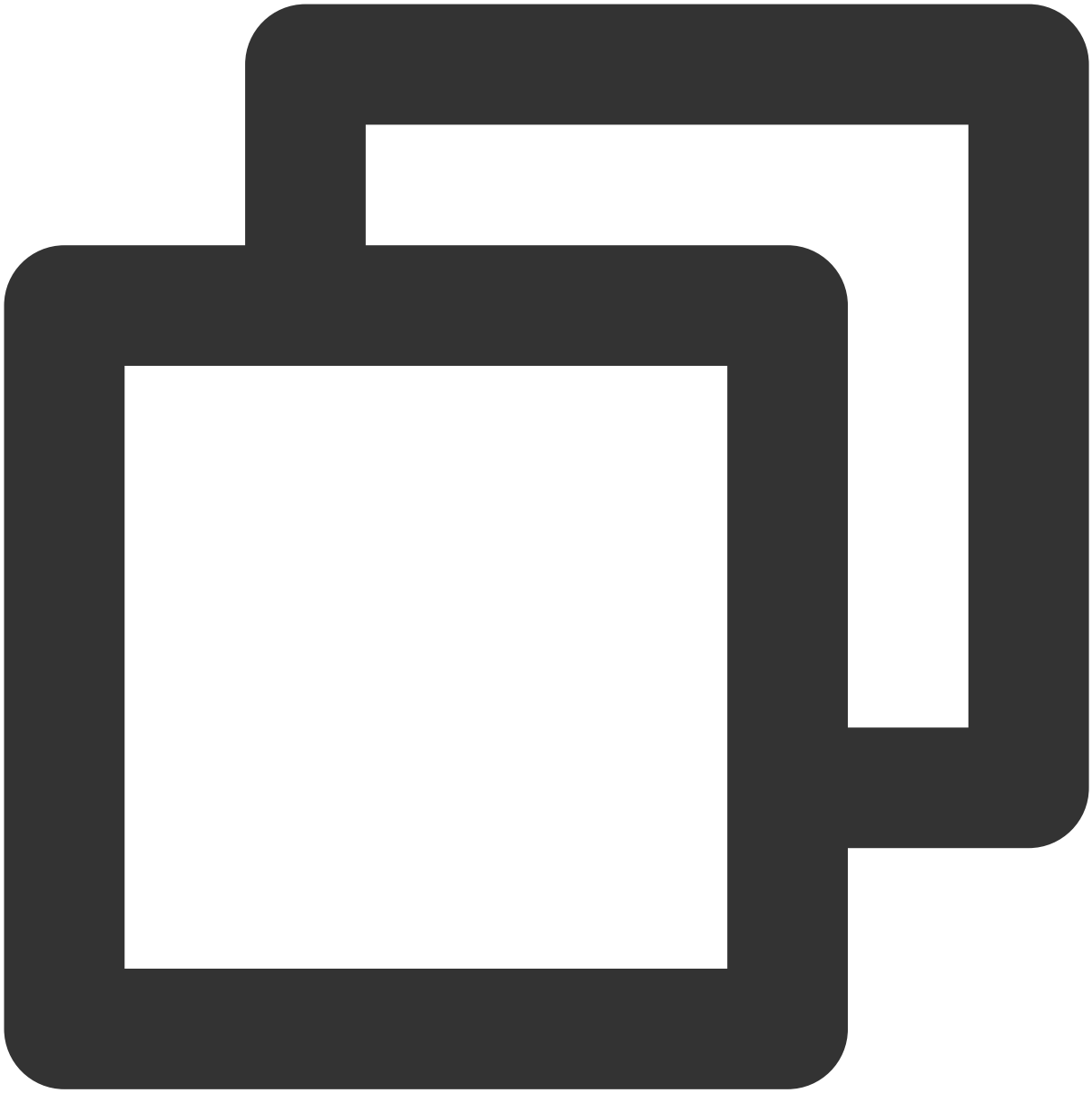
The --ssl-mode parameter indicates the SSL mode. Generally, it is recommended to use the REQUIRED and VERIFY_CA modes, which requires the MySQL client to connect to the MySQL server by using the SSL/TLS protocol and verification of the SSL/TLS certificate of the MySQL server. The VERIFY_IDENTITY mode, in addition to requiring verification of the MySQL server's SSL/TLS certificate, requires that the hostname used by the client should match the identity in the server certificate, otherwise the MySQL client will refuse to connect to the MySQL server.

3. Enter the password of the corresponding username as prompted by the system.

```
[root@VM-32-12-tencentos ~]# mysql -h [redacted]9 --ssl-ca=ca.pem --ssl-mode=VERIFY_CA -P 3306 -
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is [redacted]33
Server version: 8.0.30-txsql 20221220

Copyright (c) 2000, 2023, 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.
```

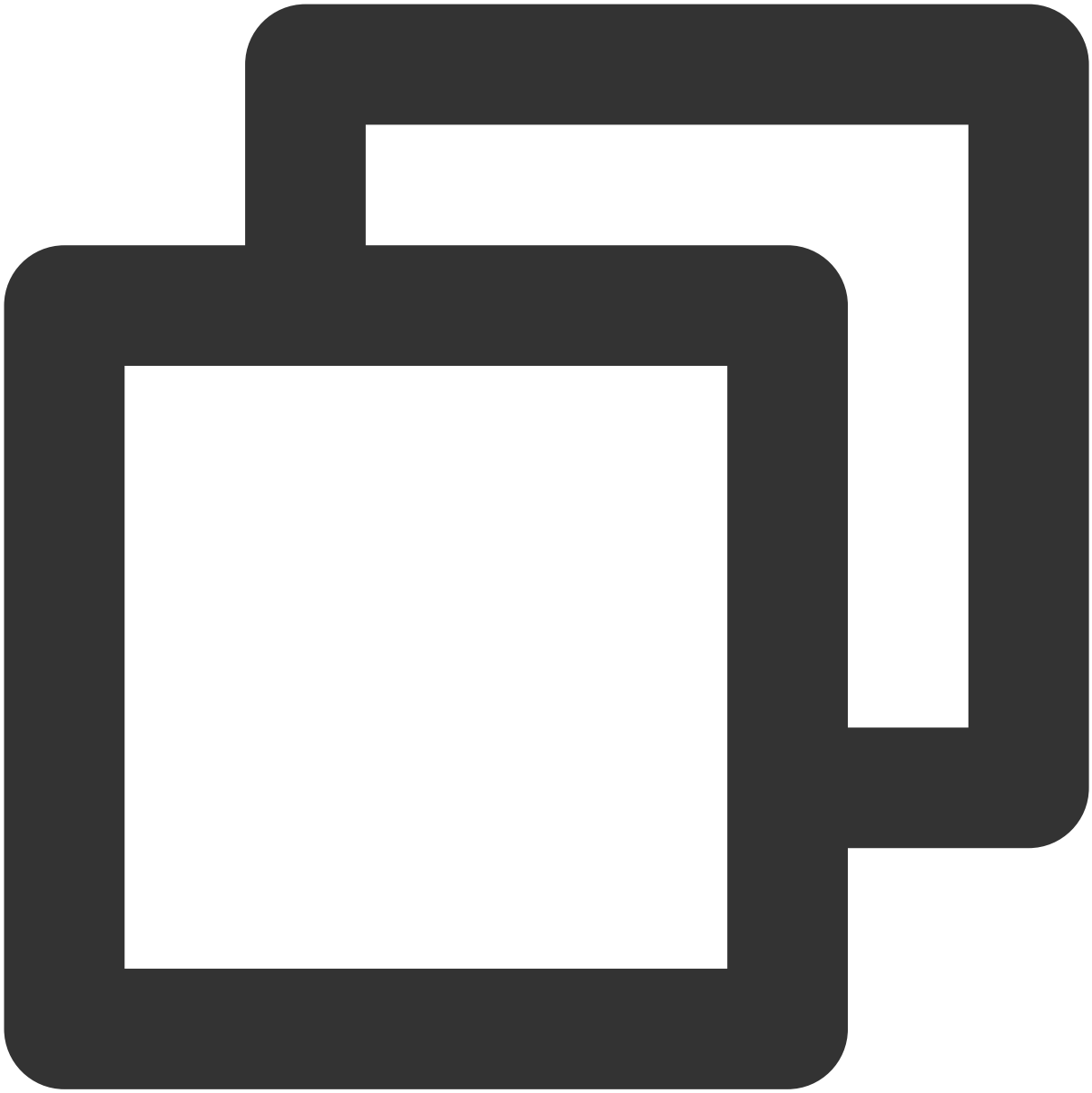
Sample Codes for Connecting to an SSL-enabled Instance for Common Programs

PHP



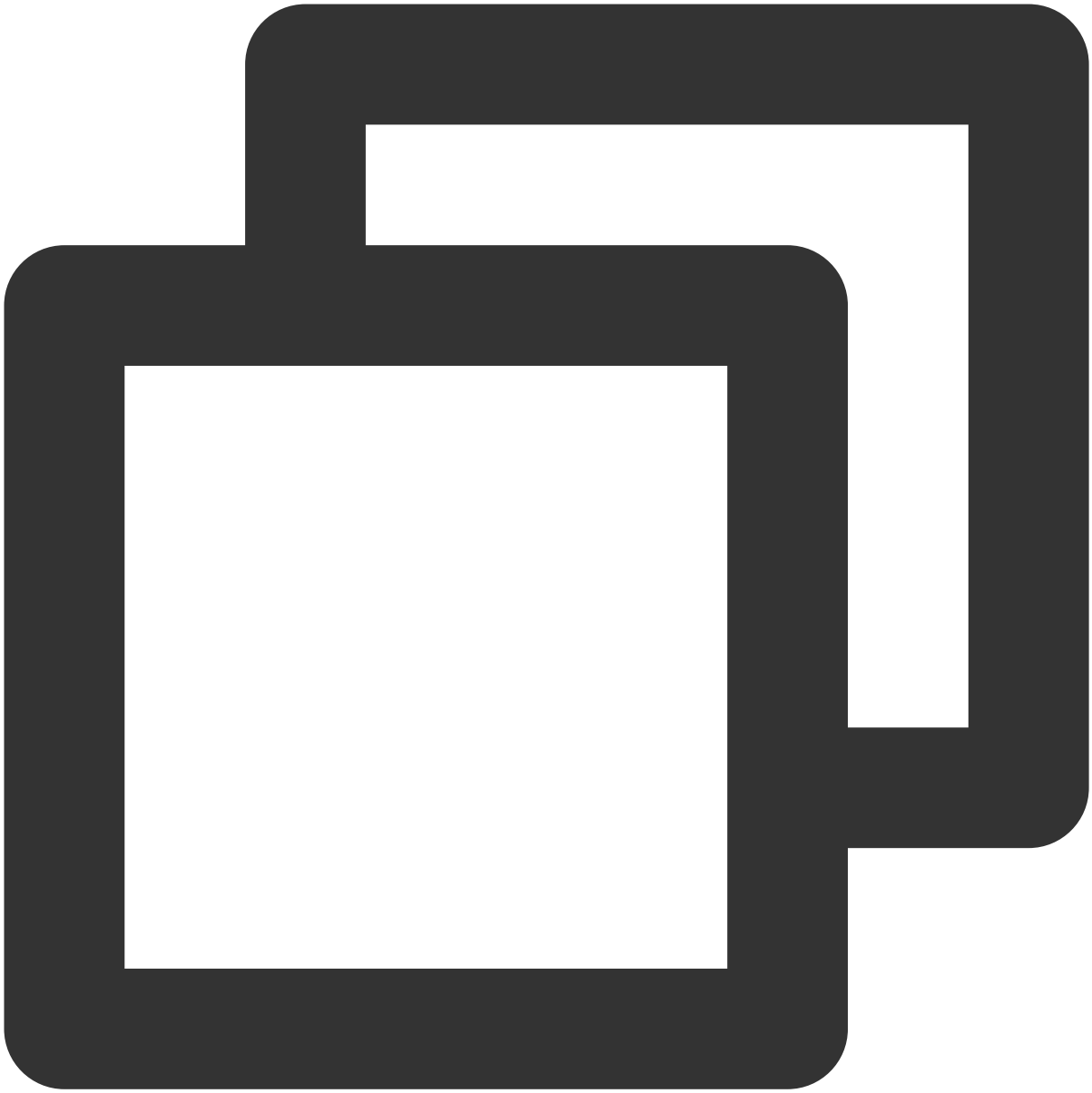
```
$conn = mysqli_init();
mysqli_ssl_set($conn, NULL, NULL, "<Downloaded certificate path>", NULL, NULL);
mysqli_real_connect($conn, '<Database access address>', '<Database access username>
if (mysqli_connect_errno($conn)) {
die('Failed to connect to MySQL: '.mysqli_connect_error());
}
```

PHP (Using PDO)



```
$options = array(
  PDO::MYSQL_ATTR_SSL_CA => '<Downloaded certificate path>'
);
$db = new PDO('mysql:host=<Database access address>;port=<Access port>;dbname=<Spec
```

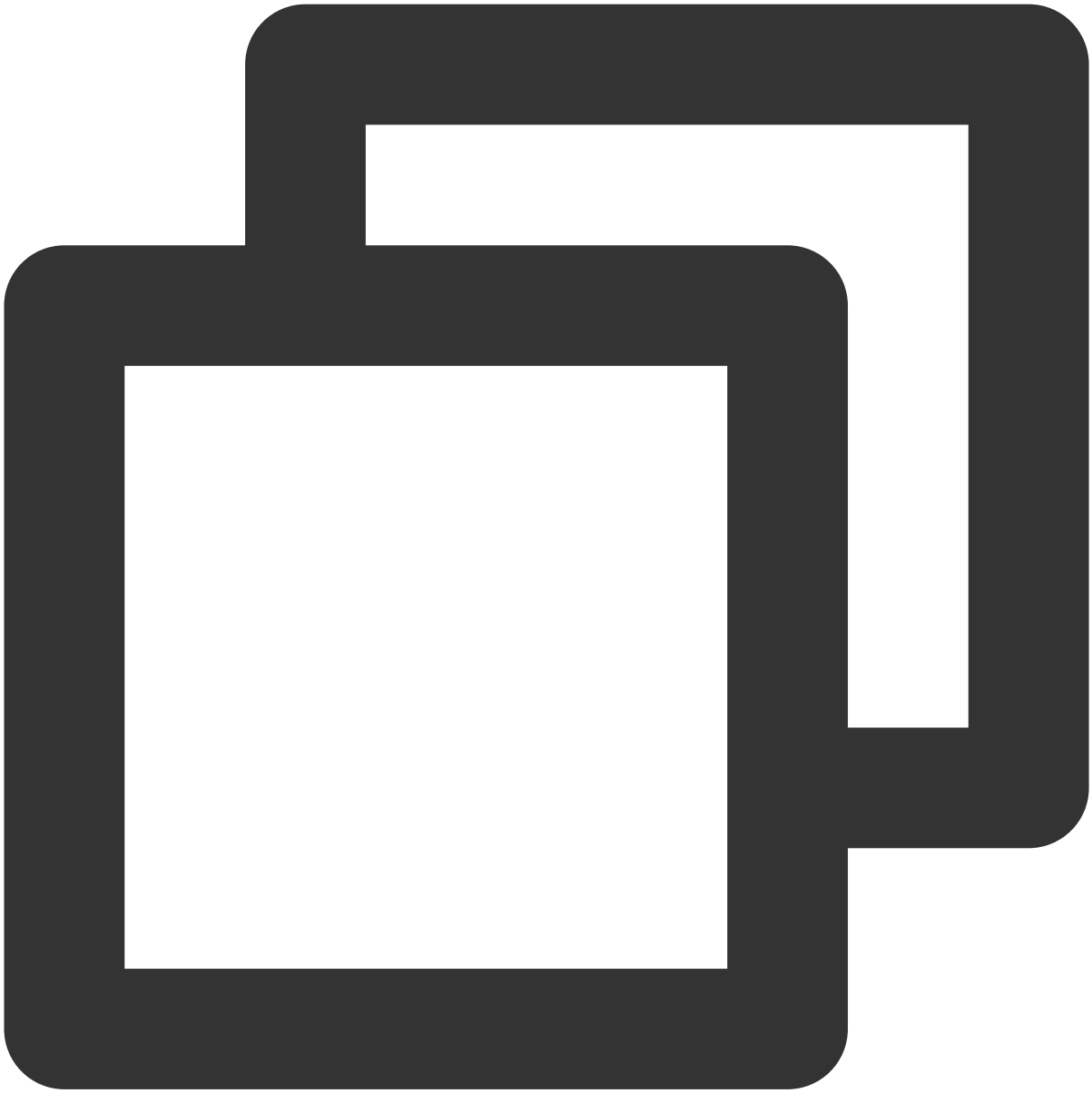
Java (MySQL Connector for Java)



```
# generate truststore and keystore in code

String importCert = " -import "+
  " -alias mysqlServerCACert "+
  " -file " + ssl_ca +
  " -keystore truststore "+
  " -trustcacerts " +
  " -storepass password -noprompt ";
String genKey = " -genkey -keyalg rsa " +
  " -alias mysqlClientCertificate -keystore keystore " +
  " -storepass password123 -keypass password " +
```

```
  " -dname CN=MS ";
sun.security.tools.keytool.Main.main(importCert.trim().split("\\\\s+"));
sun.security.tools.keytool.Main.main(genKey.trim().split("\\\\s+"));

# use the generated keystore and truststore

System.setProperty("javax.net.ssl.keyStore", "<Downloaded certificate path>");
System.setProperty("javax.net.ssl.keyStorePassword","tencentdb");
System.setProperty("javax.net.ssl.trustStore", "<Downloaded certificate path>");
System.setProperty("javax.net.ssl.trustStorePassword","tencentdb");

url = String.format("jdbc:mysql://%s/%s?serverTimezone=UTC&useSSL=true", '<Database
properties.setProperty("user", '<Database access username>');
properties.setProperty("password", '<Database access password>');
conn = DriverManager.getConnection(url, properties);
```

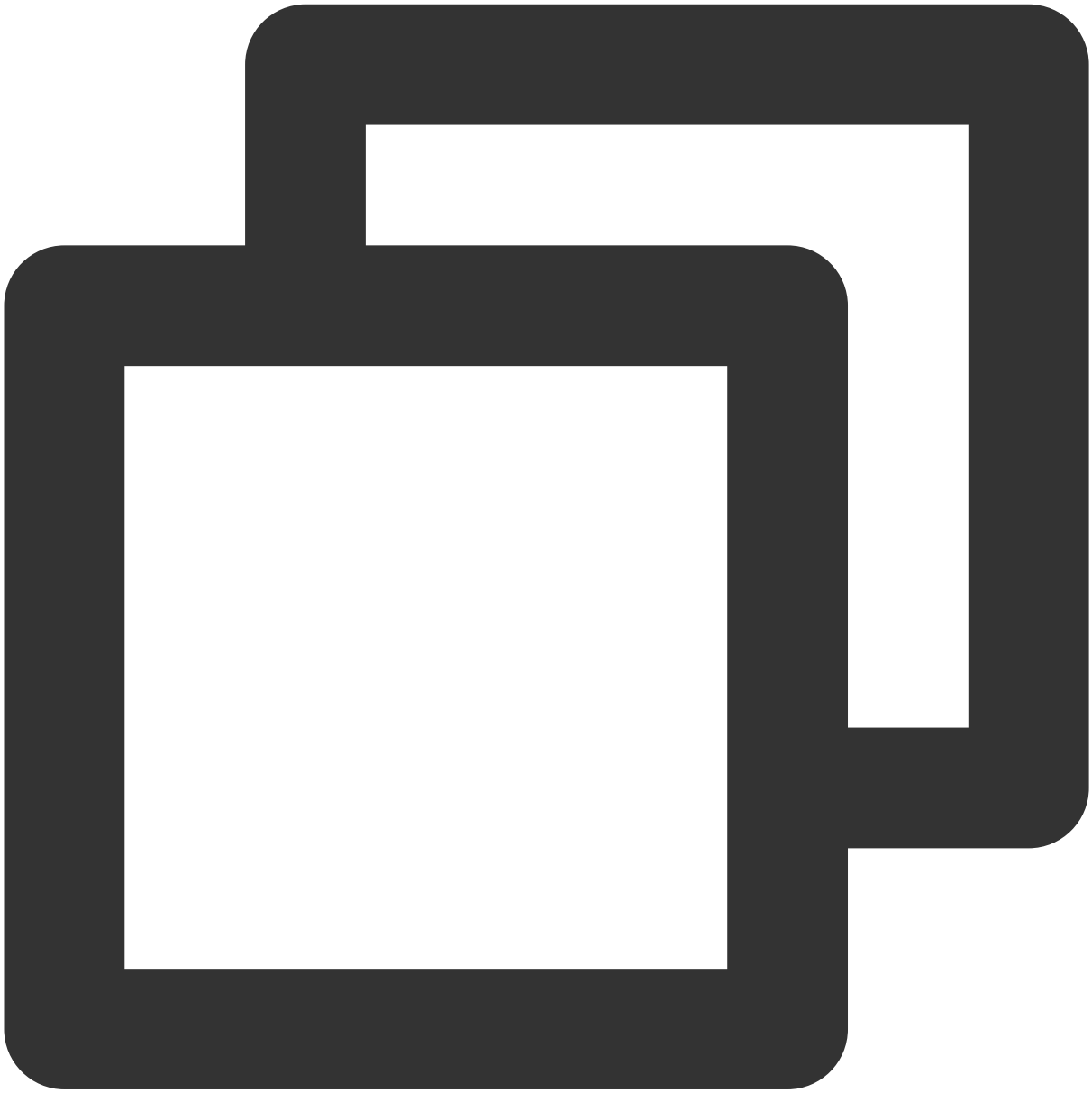
.NET (MySqlConnector)



```
var builder = new MySqlConnectionStringBuilder
{
  Server = "<Database access address>",
  UserID = "<Database access username>",
  Password = "<Database access password>",
  Database = "<Specified database to be accessed>",
  SslMode = MySqlSslMode.VerifyCA,
  SslCa = "<Downloaded certificate>",
};
using (var connection = new MySqlConnection(builder.ConnectionString))
{
```

```
    connection.Open();
}
```

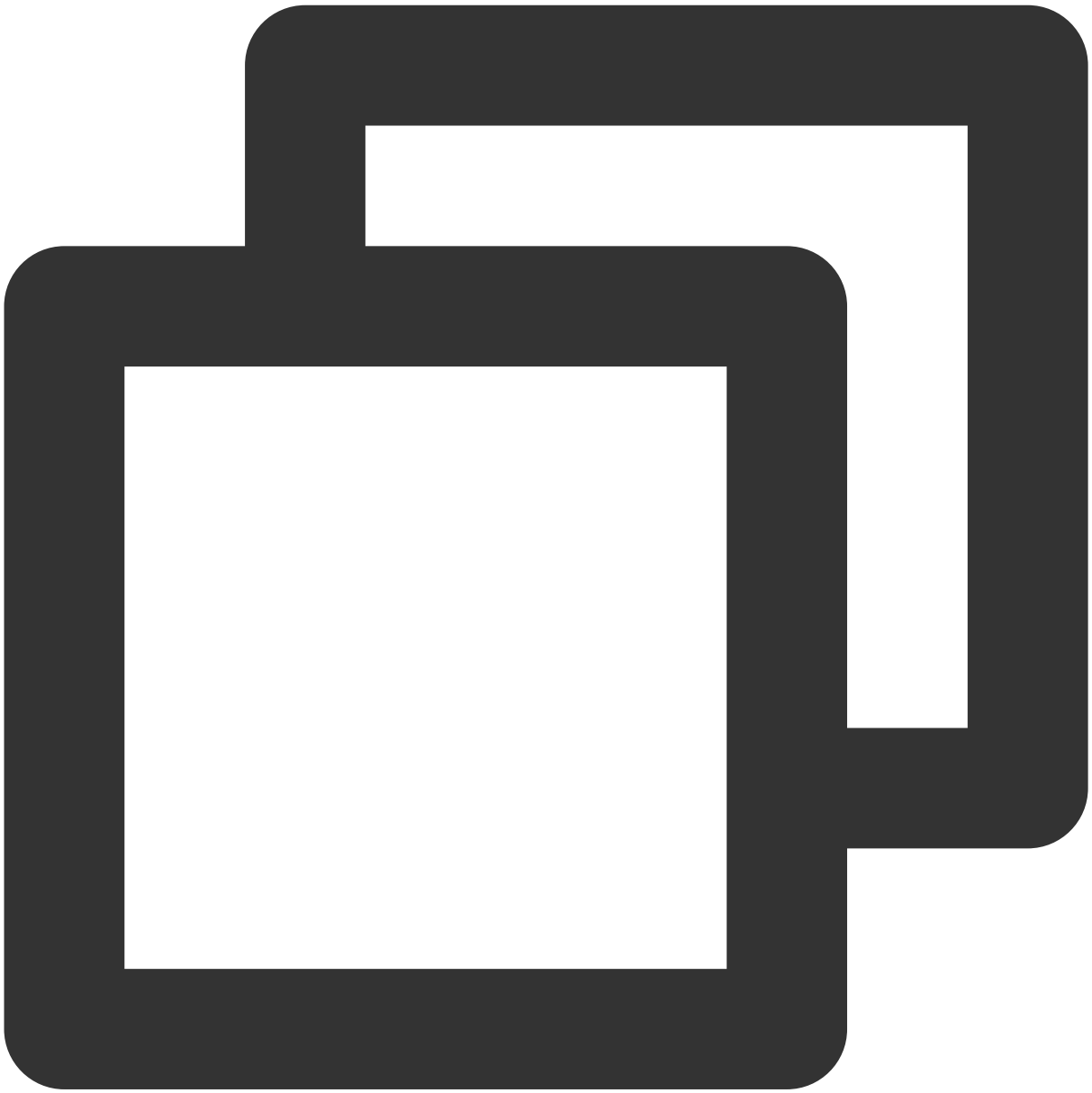
Python (MySQLConnector Python)



```
try:
    conn = mysql.connector.connect(user='<Database access username>',
                                   password='<Database access password>',
                                   database='<Specified database to be accessed>',
                                   host='<Database access address>',
                                   ssl_ca='<Downloaded certificate path>')
except mysql.connector.Error as err:
```

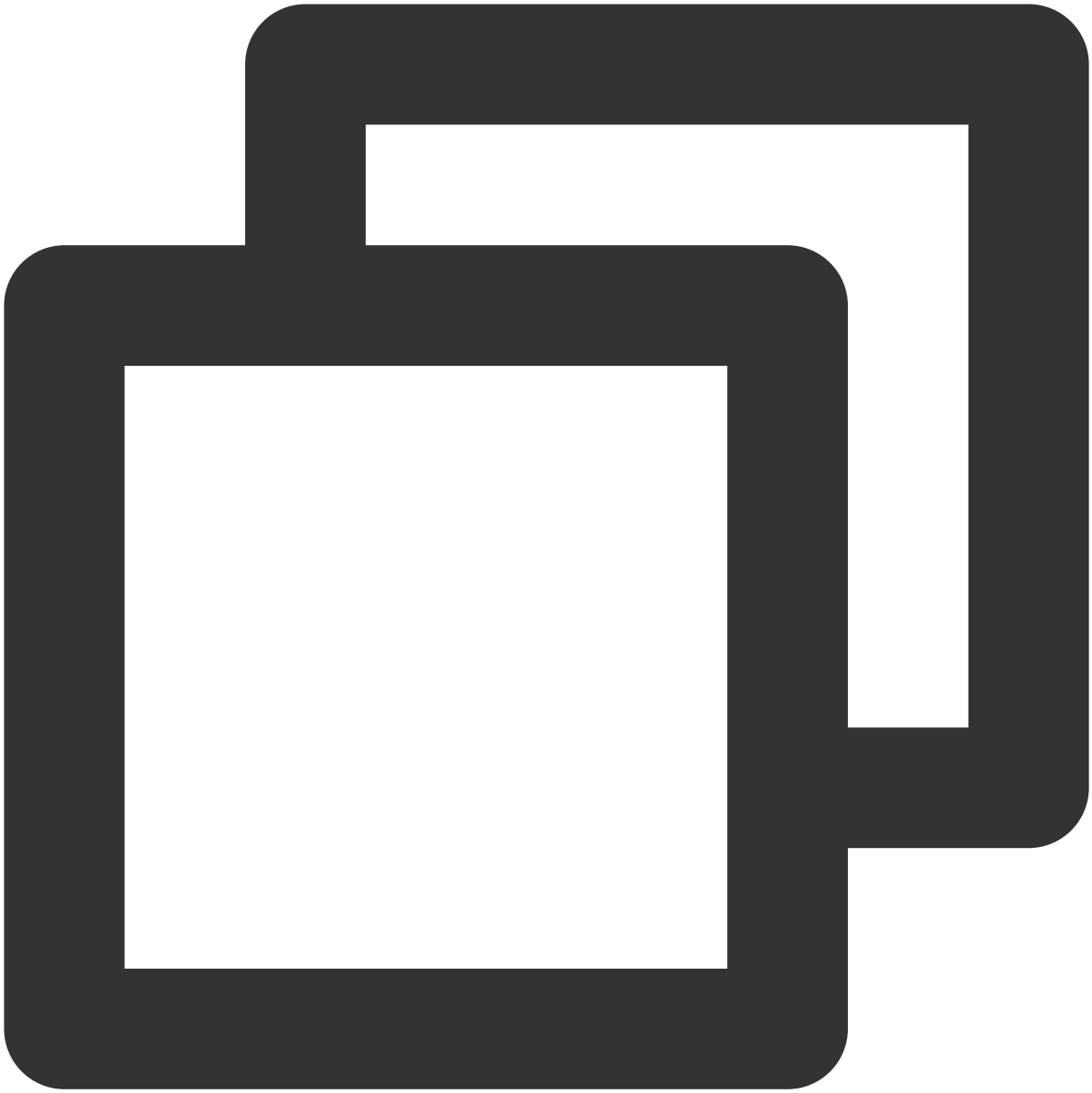
```
    print(err)
```

Python (PyMySQL)



```
conn = pymysql.connect(user='<Database access username>',
                       password='<Database access password>',
                       database='<Specified database to be accessed>',
                       host='<Database access address>',
                       ssl={'ca': '<Downloaded certificate path>'})
```

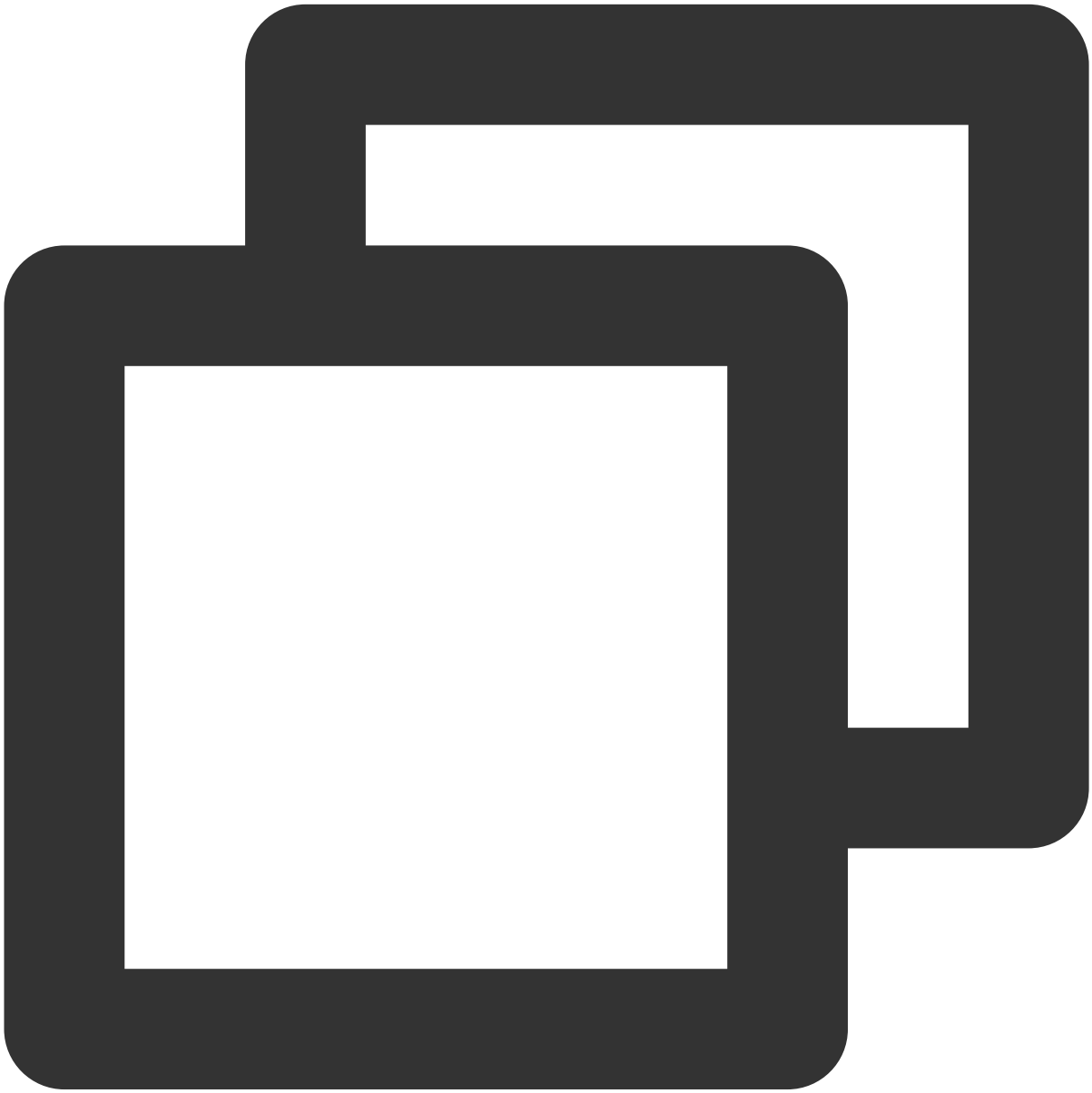
Django (PyMySQL)



```
DATABASES = {
  'default': {
      'ENGINE': 'django.db.backends.mysql',
      'NAME': '<Specified database to be accessed>',
      'USER': '<Database access username>',
      'PASSWORD': '<Database access password>',
      'HOST': '<Database access address>',
      'PORT': '<Access port>',
      'OPTIONS': {
          'ssl': {'ca': '<Downloaded certificate path>'}
      }
```

```
    }
}
```

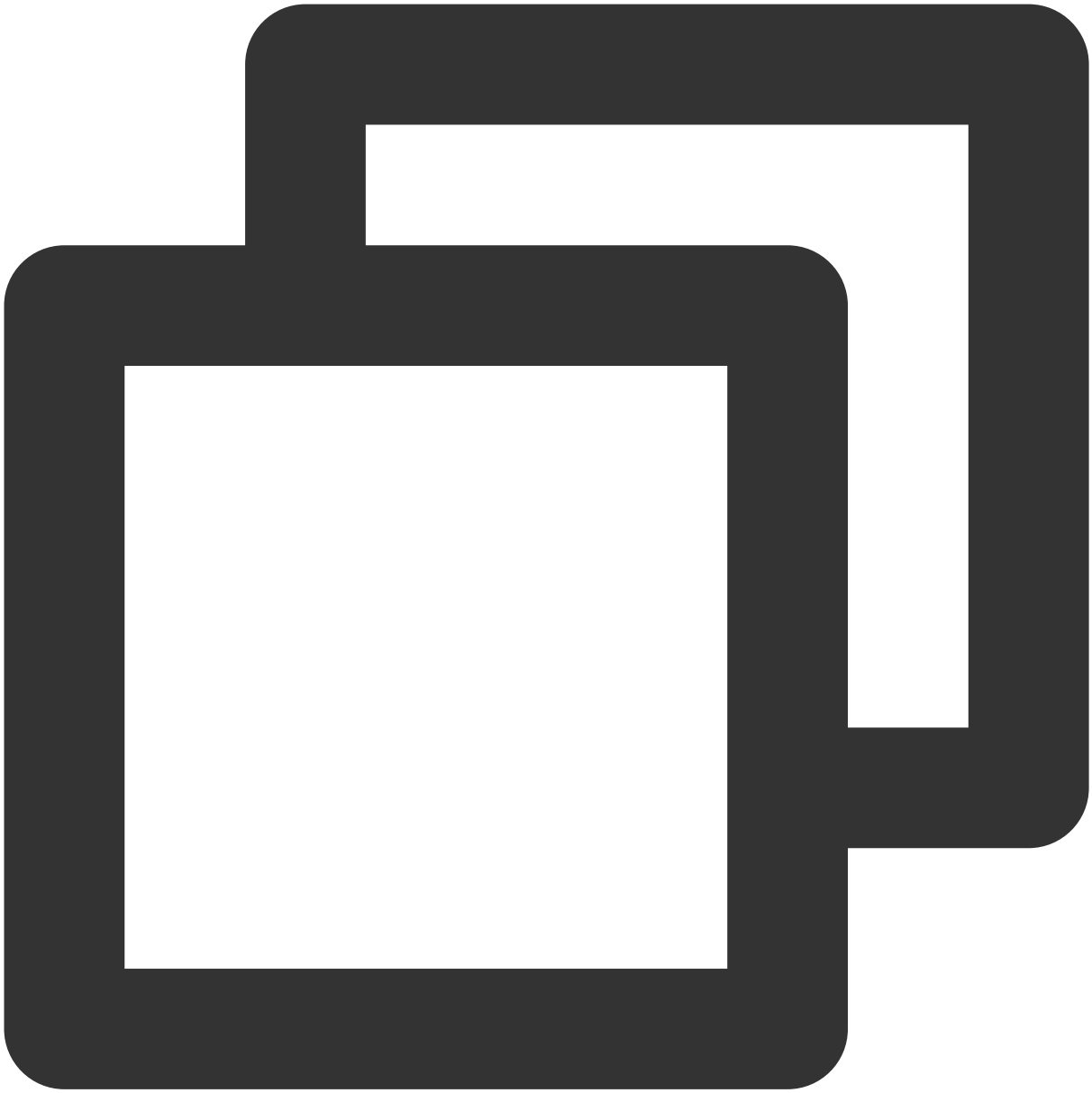
Node.js



```
var fs = require('fs');
var mysql = require('mysql');
const serverCa = [fs.readFileSync("<Downloaded certificate path>", "utf8")];
var conn=mysql.createConnection({
  host:"<Database access address>",
  user:"<Database access username>",
  password:"<Database access password>",
```

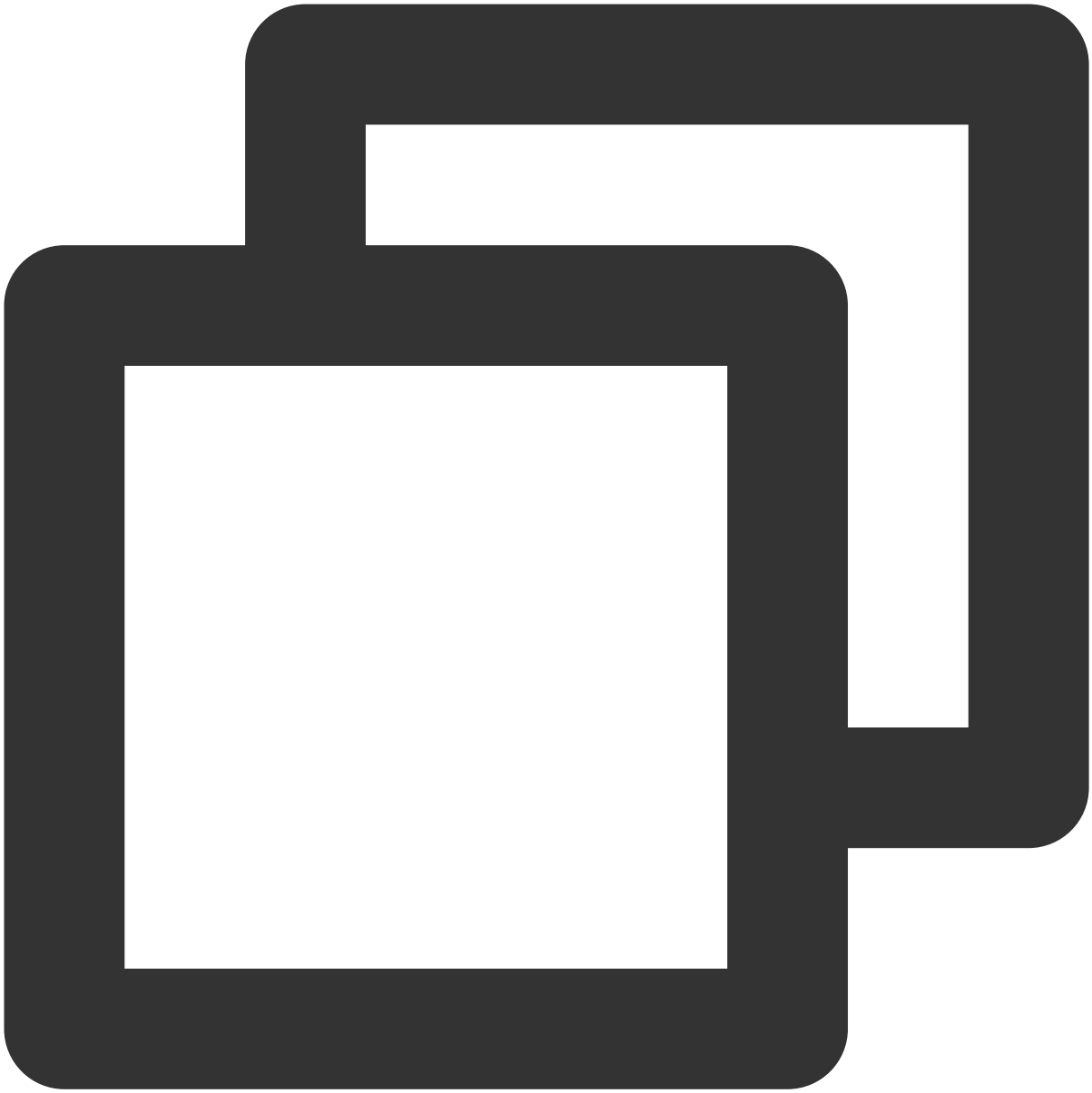
```
    database:"<Specified database to be accessed>",
    port:<Access port>,
    ssl: {
        rejectUnauthorized: true,
        ca: serverCa
    }
});
conn.connect(function(err) {
if (err) throw err;
});
```

Golang



```
rootCertPool := x509.NewCertPool()
pem, _ := ioutil.ReadFile("<Downloaded certificate path>")
if ok := rootCertPool.AppendCertsFromPEM(pem); !ok {
  log.Fatal("Failed to append PEM.")
}
mysql.RegisterTLSConfig("custom", &tls.Config{RootCAs: rootCertPool})
var connectionString string
connectionString = fmt.Sprintf("%s:%s@tcp(%s:<Access port>)/%s?allowNativePasswords
db, _ := sql.Open("mysql", connectionString)
```

Ruby



```
client = Mysql2::Client.new(
      :host     => '<Database access address>',
      :username => '<Database access username>',
      :password => '<Database access password>',
      :database => '<Specified database to be accessed>',
      :sslca => '<Downloaded certificate path>'
  )
```

Monitoring and Alarms
Overview

Last updated：2023-01-03 15:32:48

Overview

Monitoring and alarm are important features to help you maintain your database and provide reliability, availability, and performance support to troubleshooting. Monitoring enables you to stay on top of the resource utilization, performance, and status of TDSQL-C for MySQL, while alarm keeps you informed of database exceptions, so you can handle them promptly. Ultimately, this increases the system stability, improves the Ops efficiency, and reduces Ops costs.
The TDSQL-C for MySQL console has a visual **Monitoring and Alarms** page, which offers a wide variety of performance monitoring metrics and convenient monitoring features. As soon as any business exception occurs, alarm messages will be pushed to you according to the alarm rules you set. In this way, you can stay up to date with your database status, performance, and resource usage comprehensively, with no additional development required.

Concepts

Metric
Metric is a fundamental concept in CM. A metric represents a time-ordered set of data points sent to CM, such as CPU utilization and memory utilization. You can retrieve the statistics of those data points in time series. A metric can be seen as a variable to monitor, and the data points represent the values of the variable over time.
Unit
Unit refers to the measurement unit of the raw metric data; for example, the unit of CPU utilization is %.
Time granularity
Time granularity refers to a statistical period in CM. Timestamp data represents the result of aggregating all the data collected in the specified time granularity.
Alarm
Alarm management is a feature in the Cloud Monitor Alarm service. It triggers alarms for exceptions of Tencent Cloud resources and allows you to view alarm messages, customize alarm thresholds, and subscribe to alarms.

Supported monitoring types

Supported	Description

Monitoring Type	
Instance	Read-write and read-only instances can be monitored.
Database proxy	Database proxy nodes can be monitored. For more information, see Viewing Database Proxy Monitoring Data.

Monitoring granularities

In TDSQL-C for MySQL, the monitoring time granularity is automatically adjusted based on the monitoring time range as follows:

Query Time Range	Time Between Query Start Time and Current Time	Default Time Granularity	Optional Time Granularities
(0h, 1h]	(0d, 1d]	5s	5s/1min/5min
(0h, 1h]	(1d, 15d]	1min	1min/5min
(0h, 1h]	(15d, 31d]	5min	5min
(1h, 24h]	(0d, 15d]	1min	1min/5min/1h
(1h, 24h]	(15d, 31d]	5min	5min/1h
(24h, 7d]	(0d, 31d]	5min	5min/1h/1d
(7d, 31d]	(0d, 31d]	1h	1h/1d

Note:

Currently, you can view monitoring data of TDSQL-C for MySQL in the past 31 days.

Monitoring metrics

TDSQL-C for MySQL offers a rich set of monitoring metrics to help you better track your database resource utilization, performance, and status. For more information, see Supported Monitoring Metrics.

Monitoring
Supported Monitoring Metrics

Last updated：2024-06-17 16:15:40

TDSQL-C for MySQL provides monitoring metrics from both the instance dimension and the database proxy node dimension.

Monitored metrics

Note:

You can view the following monitoring metrics in the console or through TencentCloud API as instructed in Viewing Monitoring Data.

`N` in the monitoring metric unit indicates the unit of the current query time granularity. For example, if the granularity is 5 seconds, `N` will be second (sec); if it is 1 minute, `N` will be minute (min).

Parallel query monitoring metrics are supported only for TDSQL-C for MySQL 8.0 (kernel version 3.1.8) or later.

The monitoring metrics related to columnar indexes are supported by TDSQL-C for MySQL 8.0 kernel version 3.1.14.

Monitoring Metrics at the Instance Dimension

Category	Monitoring Metrics (Chinese)	Monitoring Metrics (English)	Unit	Data Aggregation Method
Resource Monitoring	CPU Utilization	cpu_use_rate	%	MAX
	Memory Utilization	memory_use_rate	%	MAX
	Memory Usage	memory_use	MB	MAX
	Storage Utilization Rate	storage_use_rate	%	MAX
	Storage Usage	storage_use	GB	MAX
	Data Table Space Usage	data_use	GB	MAX
	Temporary	tmp_use	GB	MAX

		Table Space Usage			
		Undo Table Space Usage	undo_use	GB	MA
		CCU	CCU	Individual	MA
		Total Client Traffic Sent Per Second	bytes_sent	MB/s	MA
		Total Client Traffic Received Per Second	bytes_received	MB/s	MA
		Read Request IOPS	read_iops	per second	MA
		Write Request IOPS	write_iops	per second	MA
		Total IOPS	iops	per second	MA
		Read I/O Throughput	read_bandwidth	MB/s	MA
		Write I/O Throughput	write_bandwidth	MB/s	MA
		Total I/O Throughput	io_bandwidth	MB/s	MA
		Storage Space Occupied by Log	log_capacity	GB	MA
Engine Monitoring	Connection	Operations Executed per Second	qps	per second	MA
		Transactions Executed per Second	tps	per second	MA
		Connection Utilization Rate	connection_use_rate	%	MA

		Maximum Number of Connections	max_connections	Individual	MA
		Currently Open Connections	threads_connected	Individual	MA
		Number of Threads Created	threads_created	Individual	SUM
		Number of Threads Running	threads_running	Individual	MA
	Access	Slow Queries	slow_queries	Individual	SUM
		Full-Table Scans	select_scan	Individual	SUM
		Number of Queries	com_select	Individual	SUM
		Number of Updates	com_update	Individual	SUM
		Number of Deletions	com_delete	Individual	SUM
		Number of Insertions	com_insert	Individual	SUM
		Number of Overwrite	com_replace	Individual	SUM
		Total Number of Requests	queries	Individual	SUM
		Number of Submissions	com_commit	Individual	SUM
		Number of Rollbacks	com_rollback	Individual	SUM
		Number of Full-Table Scan Composite Queries	select_full_join	Individual	SUM

		Number of Range Scan Composite Queries	select_full_range_join	Individual	SUM
		Number of Sort Merge Passes	sort_merge_passes	Individual	SUM
		Qcache Hit Rate	qcache_hit_rate	%	MIN
		Qcache Utilization Rate	qcache_use_rate	%	MIN
		Number of Multi-Table Updates	com_update_multi	Individual	SUM
		Number of Multi-Table Deletions	com_delete_multi	Individual	SUM
	Table	Number of Temporary Tables	created_tmp_tables	Individual	SUM
		Number of Times Waiting for Table Lock	table_locks_waited	Individual	SUM
		Number of Tables Already Opened	opened_tables	Individual	MAX
		Number of Table Locks Released Immediately	table_locks_immediate	Individual	SUM
		Number of Table Open Cache Hits	table_open_cache_hits	Individual	SUM
		Number of Table Open Cache Misses	table_open_cache_misses	Individual	SUM

	InnoDB	InnoDB Engine Cache Hit Rate	innodb_cache_hit_rate	%	MIN
		InnoDB Engine Cache Utilization	innodb_cache_use_rate	%	MIN
		Number of Disk Reads	innodb_os_file_reads	Individual	MAX
		Number of Disk Writes	innodb_os_file_writes	Individual	MAX
		Number of InnoDB_fsyncs	innodb_os_fsyncs	Individual	MAX
		Number of InnoDB Open Tables	innodb_num_open_files	Individual	MAX
		Number of InnoDB Reads	innodb_data_read	Byte	SUM
		Total Number of InnoDB Reads	innodb_data_reads	Individual	SUM
		Total Number of InnoDB Writes	innodb_data_writes	Individual	SUM
		Number of InnoDB Writes	innodb_data_written	Byte	SUM
		Number of InnoDB Rows Deleted	innodb_rows_deleted	Individual	SUM
		Number of InnoDB Rows Inserted	innodb_rows_inserted	Individual	SUM
		Number of InnoDB Rows Updated	innodb_rows_updated	Individual	SUM
		Number of InnoDB Rows	innodb_rows_read	Individual	SUM

		Read			
		Average InnoDB Row Lock Acquiring Time	innodb_row_lock_time_avg	Milliseconds	MA
		Number of InnoDB Row Lock Waits	innodb_row_lock_waits	Individual	SUM
		Number of InnoDB Dirty Pages	innodb_buffer_pool_pages_dirty	Individual	MA
		Number of InnoDB Pending Writes	innodb_data_pending_writes	Individual	MA
		Number of InnoDB Pending Reads	innodb_data_pending_reads	Individual	MA
		Number of InnoDB Log Wait Writes	innodb_log_waits	Individual	SUM
		Number of InnoDB Log Physical Writes	innodb_log_writes	Individual	SUM
		Number of InnoDB Log Physical Write Requests	innodb_log_write_requests	Individual	SUM
	Tmp	Number of Temporary Tables	created_tmp_disk_tables	Individual	SUM
		Number of Temporary Files	created_tmp_files	Individual	SUM
	Handler	Number of Requests to Read Next Line	handler_read_rnd_next	Individual	SUM

		Number of Internal Rollbacks	handler_rollback	Individual	SUM
		Number of Internal Submissions	handler_commit	Individual	SUM
	Buffer	Number of InnoDB Empty Pages	innodb_buffer_pool_pages_free	Individual	MAX
		Total Number of InnoDB Pages	innodb_buffer_pool_pages_total	Individual	MAX
		Number of InnoDB Logical Reads	innodb_buffer_pool_read_requests	Individual	SUM
		Number of InnoDB Physical Reads	innodb_buffer_pool_reads	Individual	SUM
		Number of InnoDB Buffer Pool Writes	innodb_buffer_pool_write_request	Individual	SUM
	Parallel Query	Number of Parallelly Queried Threads	txsql_parallel_threads_currently_used	Individual	MAX
		Number of Failed Parallel Queries	txsql_parallel_stmt_error	Individual	SUM
		Number of Executed Parallel Queries	txsql_parallel_stmt_executed	Individual	SUM
		Number of Rollback Serial Queries	txsql_parallel_stmt_fallback	Individual	SUM
	Column Store Index	Number of Column Store	innodb_csi_select_pushdown	Individual	SUM

		Index Execution Statements			
		Number of Column Store Index Rollback Statements	innodb_csi_select_fallback	Individual	SUM
		Column Store Index Synchronization Delay	innodb_csi_lag_seconds	Second	MAX
		Column Index Storage Usage	innodb_csi_disk_usage	GB	MAX
		Number of CSI Creation Failures	innodb_csi_create_failed	Individual	SUM
	Other	Total Number of Open Files	open_files	Individual	MAX
Deployment Monitoring		Replication Status	replication_status	0-Yes,1-No	If th take
		Replication Delay	replication_delay	Milliseconds	MAX
		Replication Lag in lsn Distance	replication_delay_distance	Bytes	MAX

Monitoring Metrics at Database Proxy Node Dimension

To obtain monitoring metrics at the database proxy node dimension, you need to first enable the database proxy. For detailed metric descriptions, refer to Viewing Database Proxy Monitoring Data.

Performance monitoring data analysis

You can set alarm policies to monitor and analyze performance metrics. Below are some examples:
CPU utilization: It describes how the CPU in the database server is used while running. If the CPU load is high, there will be problems such as slow response of the database system and query delay. Therefore, you can set a threshold

alarm for this monitoring metric as needed. Generally, if the CPU utilization exceeds 90%, you need to check the request volume and query performance, and make timely adjustments. If the CPU utilization meets the application or database requirements such as throughput and concurrency and is expected, then it is appropriate.

Used storage space: If the used storage space is always greater than or equal to 85% of the total disk space, you need to check how the storage space is used and whether the data can be deleted from the instance or archived to another system to free up space. If the amount of stored data is too large, the disk is about to be full, and you can solve the problem of insufficient disk space through capacity expansion.

Traffic inbound to/outbound from client: You can check the fluctuations of these metrics in **Resource Monitoring** to determine whether the problem is caused by the business peak or database, and then optimize the business or adjust the database configuration.

Max connections: `max_connections` is one of the parameters in TDSQL-C for MySQL, which is used to specify the maximum number of connections allowed by the database. When the database process opens a new connection, the number of currently connected clients will be queried. If the number has reached the set value of `max_connections`, a new connection cannot be established. If there is a large number of connections, the instance performance drops, and the response time increases, you need to consider restricting the maximum number of connections to the database. If the value of `max_connections` is set too high, it may cause problems such as insufficient memory and thread crash in the system. Therefore, you should carefully evaluate your application needs and set an appropriate value. The value generally falls between 200 and 500. You can also adjust the maximum number of connections based on the connection conditions of applications and databases.

When a performance metric exceeds the configured alarm threshold, you may need to modify relevant parameters to optimize the database to improve its performance and availability.

Viewing Monitoring Data

Last updated：2023-11-09 15:26:18

TDSQL-C for MySQL provides a rich set of performance monitoring metrics to help you stay up to date with the detailed status and performance of your database. Generally, you can view monitoring metrics and data in the TDSQL-C for MySQL console by creating a dashboard in Tencent Cloud Observability Platform (TCOP) or through TencentCloud API. We recommend that you choose the console, where you can quickly obtain fine-grained monitoring data for the target time range and locate Ops problems accordingly.
This document describes how to view the monitoring data and perform visual operations on the monitoring page.

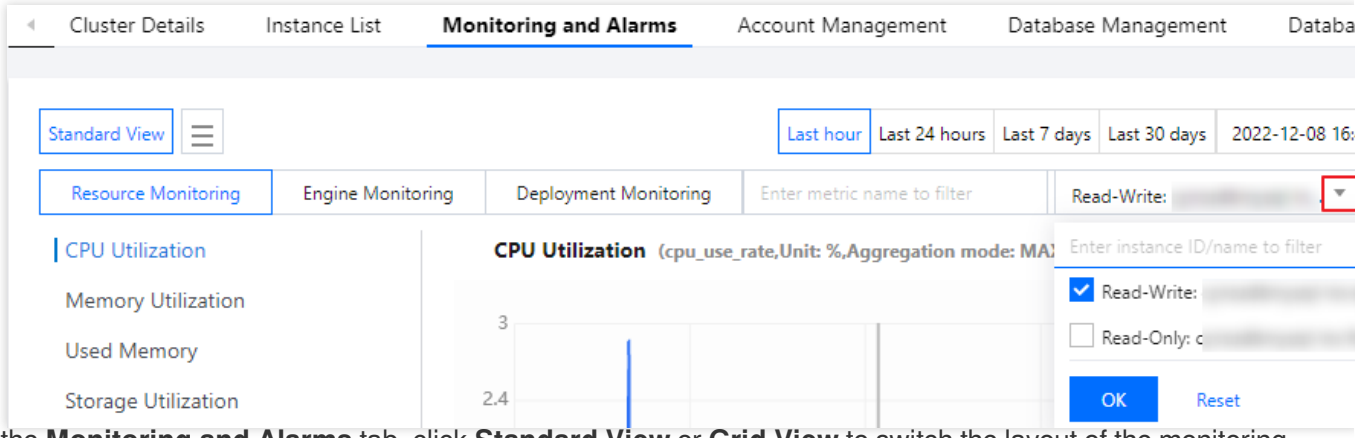
Viewing Monitoring Data

Option	Strengths	Operation
Viewing data in the TDSQL-C for MySQL console	The console is simple and visual, where you can quickly locate problems in a familiar way.	Viewing Monitoring Data in the Console
Creating a TCOP dashboard to view data	You can visually create customized monitoring metric groups.	Creating Dashboard to View Monitoring Data
Pulling data through TencentCloud API	You can flexibly pull monitoring metric data to analyze or connect it to other platforms.	Pulling Monitoring Data via TencentCloud API

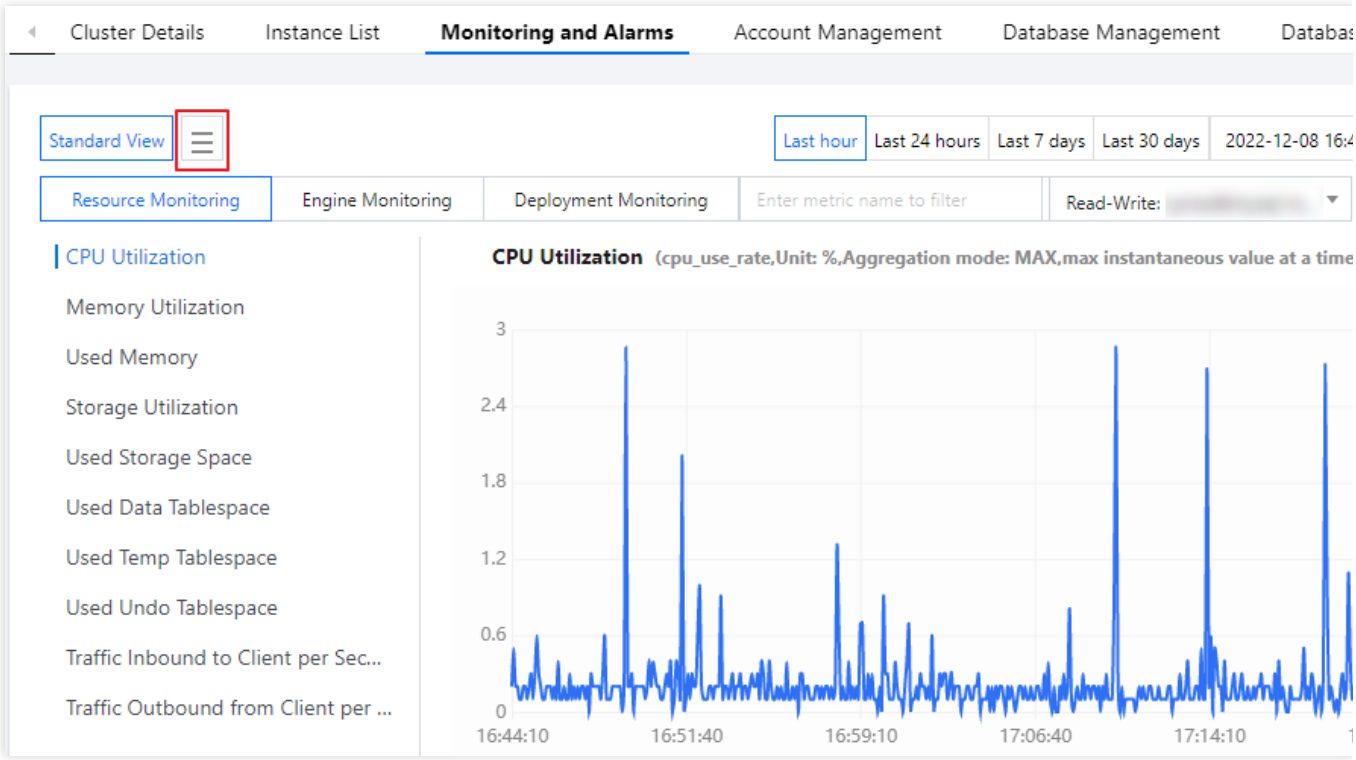
Visual Operations on the Monitoring Page

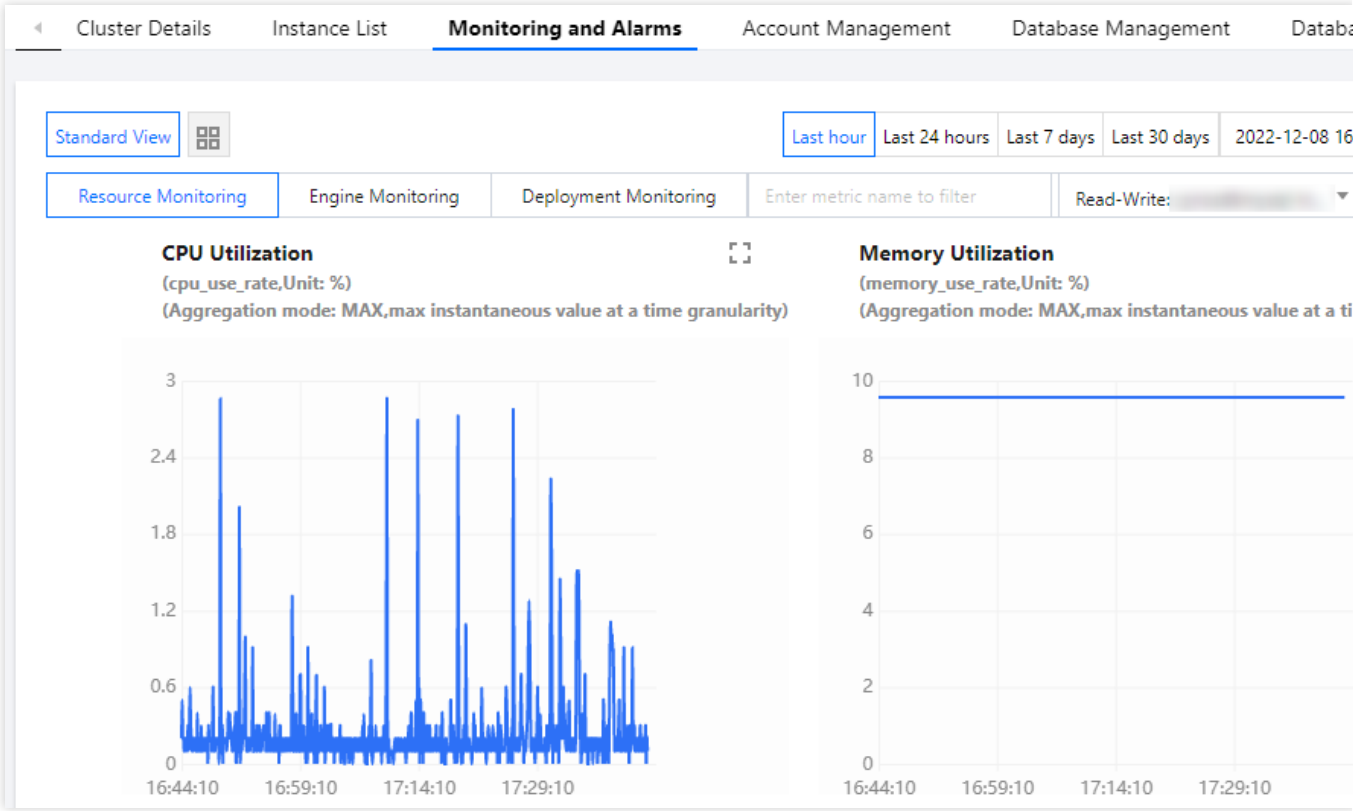
Selecting the standard monitoring view

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select **Monitoring and Alarms**, select one or multiple target instances in the drop-down list, and click **OK**.



4. On the **Monitoring and Alarms** tab, click **Standard View** or **Grid View** to switch the layout of the monitoring view.





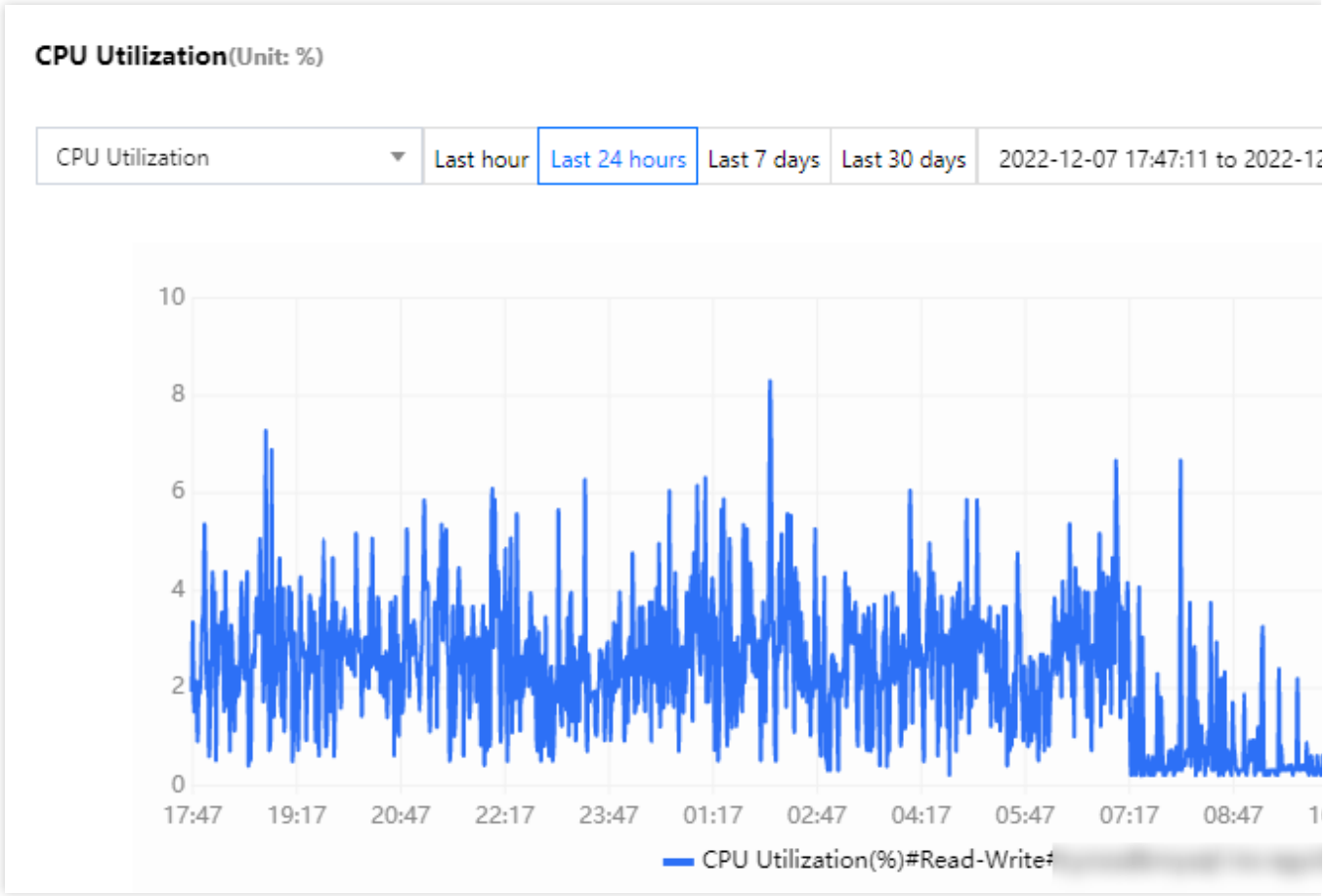
Displaying a chart in full screen

You can display a single metric in full screen for a clearer preview of metric data.

1. On the **Monitoring and Alarms** tab, click

on the right of the corresponding metric to display the metric in full screen.

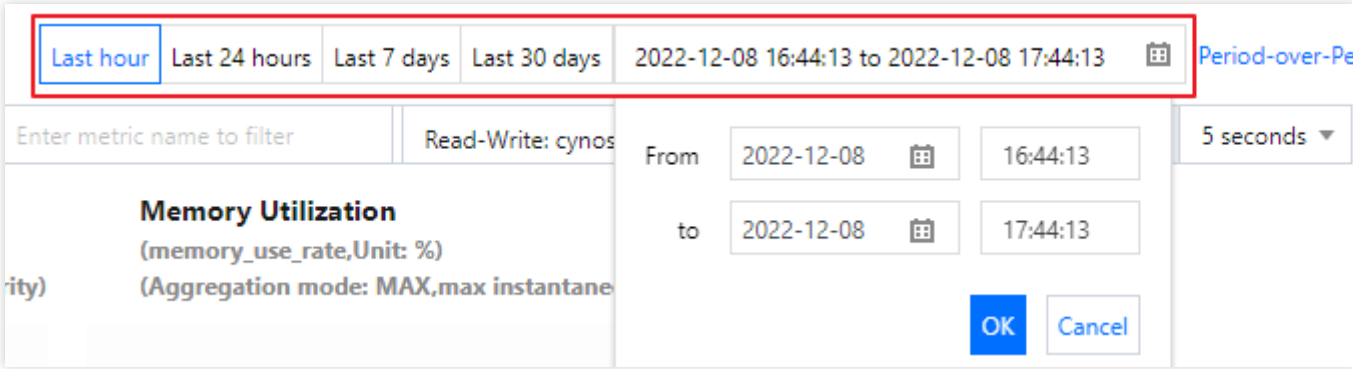
2. After the data is displayed in full screen, you can filter the metric and select a time range and time granularity to view the metric data. You can click **X** in the top-right corner to close the full screen window.



Selecting a monitoring time range

You can select or customize a time range to query the monitoring data over this time period.

On the **Monitoring and Alarms** tab, you can select **Last hour**, **Last 24 hours**, **Last 7 days**, or **Last 30 days**. You can also click the time picker to specify the start time and end time for monitoring data query.



Adding period-over-period comparison

You can compare monitoring data from multiple time ranges by adding period-over-period comparison.

Note:

The time range in the first time picker determines the time ranges in new time pickers. For example, if it is 3 days in the first time picker, then you can select only the start time in a new time picker, and the end time will be 3 days later by default.

Use limits

If you select multiple instances to view monitoring data, the period-over-period comparison feature is not supported.

You can compare the data of an instance monitoring metric within up to three time ranges.

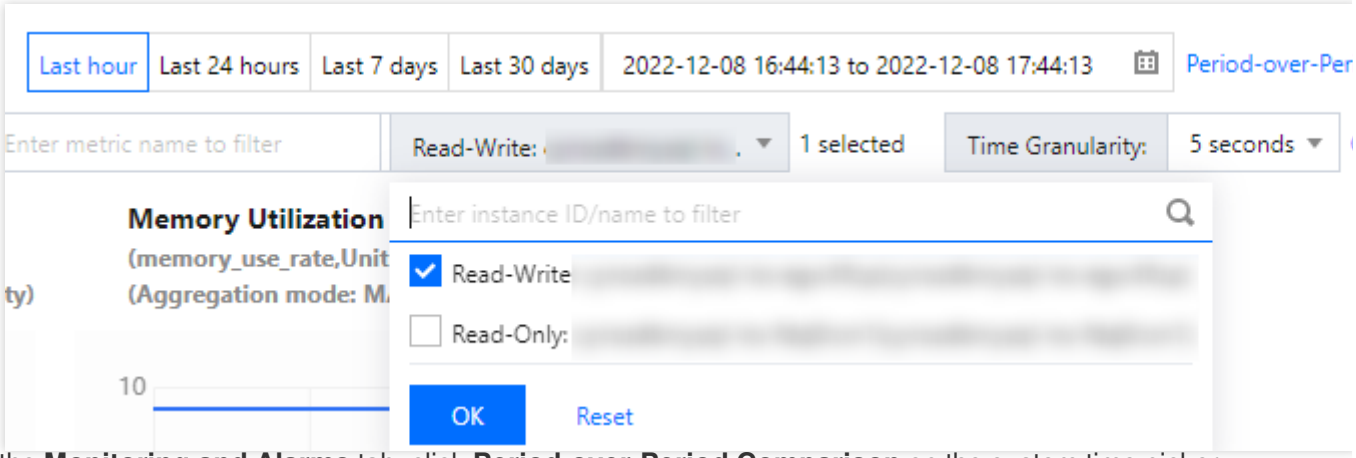
Directions

1. Log in to the TDSQL-C for MySQL console.

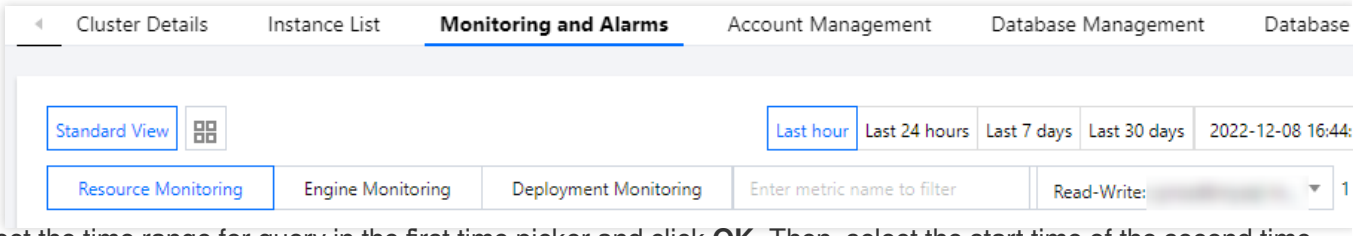
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.

3. On the cluster management page, select the **Monitoring and Alarms** tab.

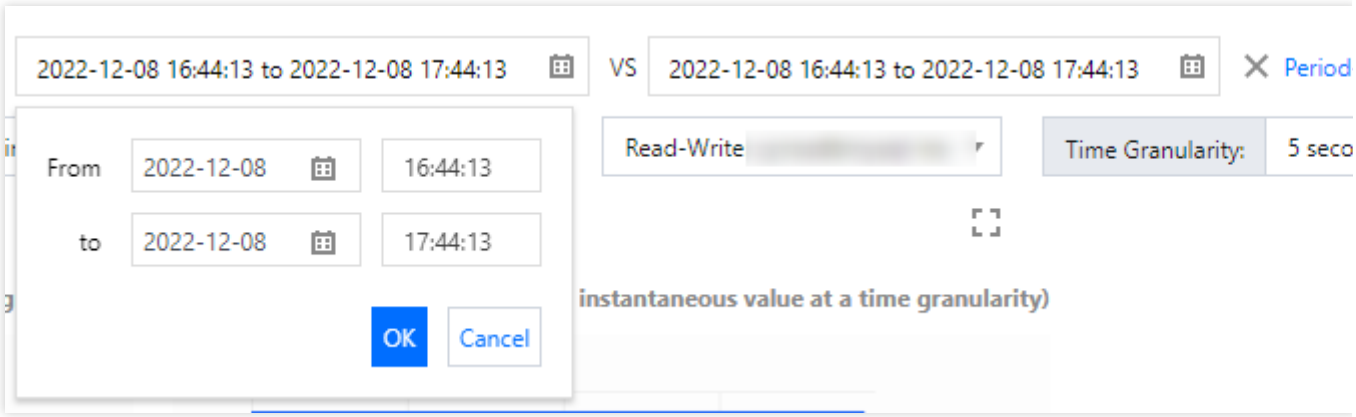
4. Select a target instance in the drop-down box and click **OK**.



5. On the **Monitoring and Alarms** tab, click **Period-over-Period Comparison** on the custom time picker



6. Select the time range for query in the first time picker and click **OK**. Then, select the start time of the second time range in the second time picker and click **OK**.



Note:

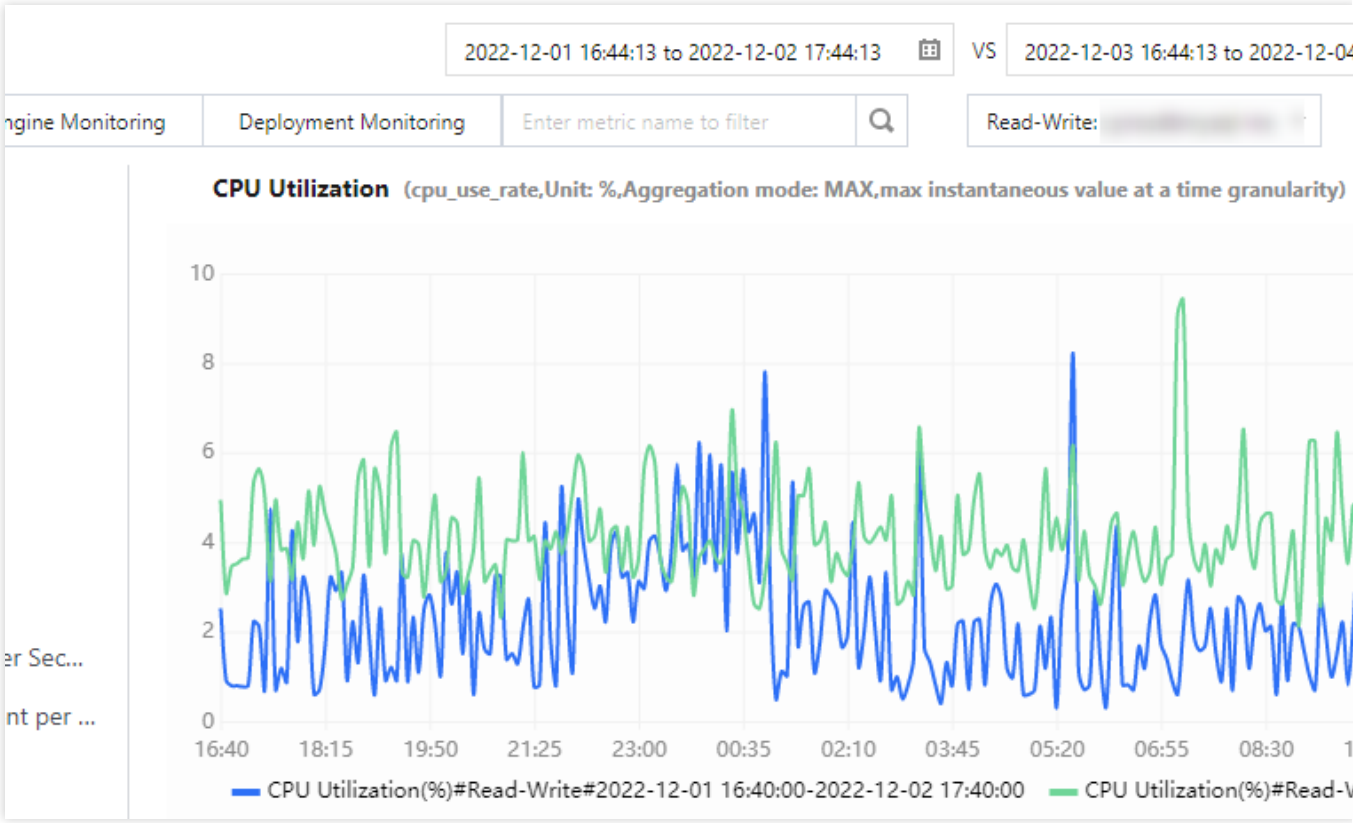
In new time pickers, you only need to select the start time, and the time range will be automatically aligned with that in the first time picker. For example, if the time range in the first time picker is 3 days, then all time ranges added subsequently will also be 3 days.

7. After you select the time ranges for comparison, you can query the monitoring metric details of the instance within the selected time ranges at the specified time granularity.

Setting a monitoring time granularity

You can select different time granularities for monitoring data query within the specified time range.

On the **Monitoring and Alarms** tab, select a time granularity in the **Time Granularity** drop-down list.



Time ranges and corresponding time granularities

Query time range	Range between query start time and current time	Default time granularity	Optional time granularities
(0h, 1h]	(0d, 1d]	5s	5s/1min/5min
(0h, 1h]	(1d, 15d]	1min	1min/5min
(0h, 1h]	(15d, 31d]	5min	5min
(1h, 24h]	(0d, 15d]	1min	1min/5min/1h
(1h, 24h]	(15d, 31d]	5min	5min/1h

(24h, 7d]	(0d, 31d]	5min	5min/1h/1d
(7d, 31d]	(0d, 31d]	1h	1h/1d

Example:

Suppose the current time is 15:00, July 10, 2022, and the query time range is 15:00–16:00, June 29, 2022 (1 hour), then the query time range is 1 hour. As the time between the query start time and the current time is longer than 1 day but shorter than 15 days, the default time granularity is 1 minute. You can also select the option of 5 minutes.

Note:

Currently, you can view monitoring data of TDSQL-C for MySQL in the past 31 days.

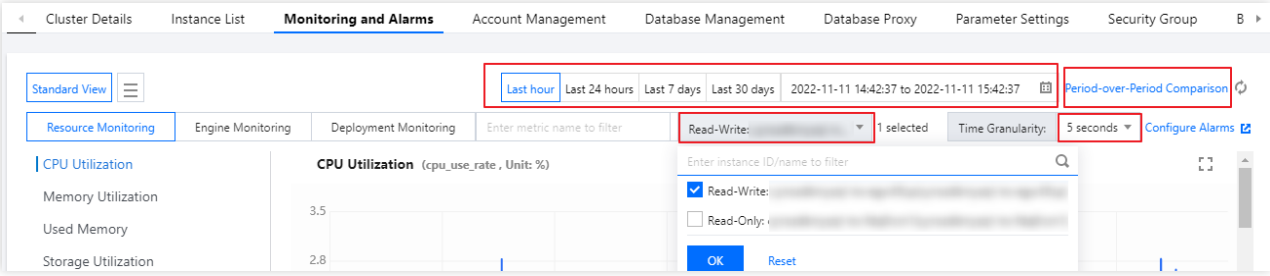
Viewing Monitoring Data in the Console

Last updated：2023-01-03 15:32:48

This document describes how to view monitoring data in the console.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select the **Monitoring and Alarms** tab.
4. Select one or multiple target instances and the query time range to view instance monitoring metrics. You can also add period-over-period comparison and select the time granularity.



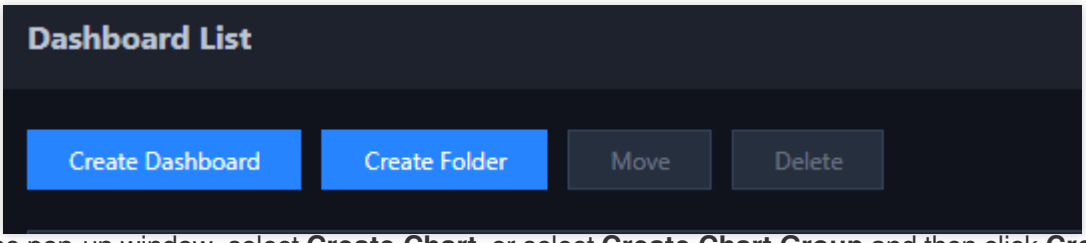
Creating Dashboard to View Monitoring Data

Last updated：2023-08-24 10:06:18

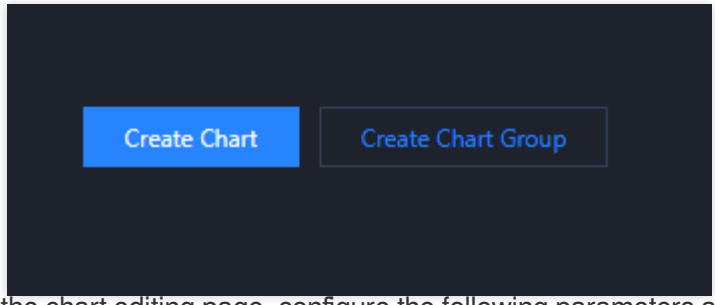
This document describes how to create a Tencent Cloud Observability Platform (TCOP) dashboard to view monitoring data.

Directions

1. Log in to the TCOP console.
2. On the left sidebar, select **Dashboard** > **Dashboard List**.
3. Below the dashboard list, click **Create Dashboard**.



4. In the pop-up window, select **Create Chart**, or select **Create Chart Group** and then click **Create Chart** in the created chart group.

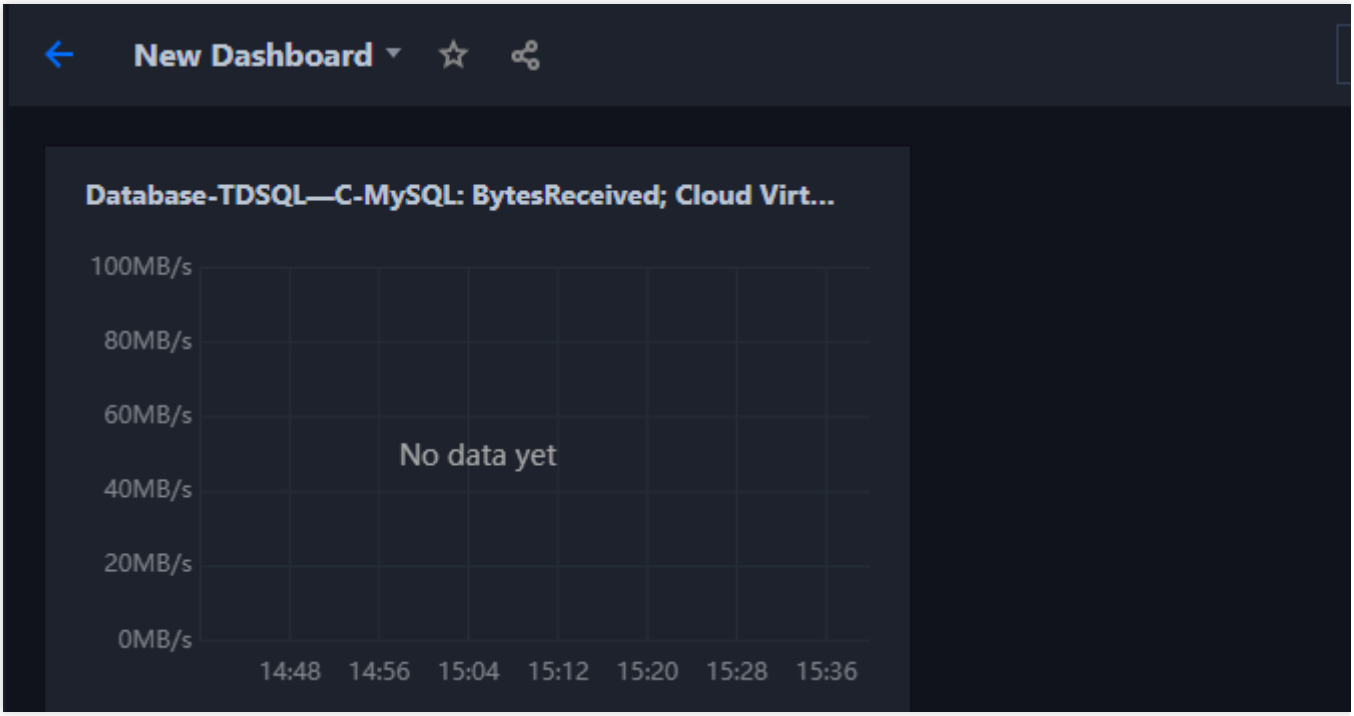


5. On the chart editing page, configure the following parameters and click **Save** in the top-right corner.

Parameter	Description
Select Product	Select **Tencent Cloud services**.
Metric	Select **TencentDB** > **TDSQL-C** > **MySQL** and select the target monitoring metric.
Filter	Select a filter to extract the data that meets the conditions for display on the chart. You can add multiple filters, and the chart displays only the data meeting all of them. Data can be filtered by instance, tag, and template variable.
Group by	Select a tag to aggregate and display data by group (optional).

Comparison	You can select day-over-day (compared with the same period yesterday), week-over-week (compared with the same period last week), and custom dates for comparison.
Alias	Enter the legend alias.
Enable Sorting	After it is enabled, you can set a sorting rule.
Sorting Rule	You can select the maximum, minimum, average, or sum value and specify whether to display items in ascending or descending order.
Displayed Quantity	Set the number of instances to be displayed.

6. After the configuration is saved successfully, you can view the monitoring data of the metric selected during chart creation.



Note:

This method is used to create a chart for a single monitoring metric. To create charts for multiple metrics, simply repeat the above steps.

Pulling Monitoring Data via TencentCloud API

Last updated：2023-09-12 14:59:09

This document describes how to pull the monitoring metric data of TDSQL-C for MySQL through TencentCloud API. For detailed directions and more examples, see GetMonitorData.

API description

This API is used to get the monitoring data of TDSQL-C for MySQL by passing in its namespace, object dimension, and monitoring metric.
API call rate limit: 20 calls/sec (1,200 calls/minute). A single request can get the data of up to 10 instances for up to 1,440 data points.
This API may fail due to the rate limit if you need to call many metrics and objects. We recommend you distribute call requests across a period of time.
The API request rate is limited to 20 requests/sec by default.

Input parameters

The list below only contains API request parameters and some common parameters. For a complete common parameter list, see Common Params.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: `GetMonitorData` .
Version	Yes	String	Common parameter. The value used for this API: `2018-07-24` .
Region	Yes	String	Common parameter. For more information, see the list of regions supported by the product.
Namespace	Yes	String	Namespace, such as `QCE/cynosdb_mysql` . For more information, see TDSQL-C for MySQL Monitoring Metrics.
MetricName	Yes	String	Metric name, such as `CPUUsagerate` (CPU utilization). You can pull the data of one single metric at a time.
Instances.N	Yes	Array of	Dimension combination of instance object in the format of

		Instance	`key-value` pair. Different types of instances have completely different fields. For the dimension combination of TDSQL-C for MySQL, see TDSQL-C for MySQL Monitoring Metrics.
Period	No	Integer	Statistical period for monitoring data in seconds, such as 60. Default value: 300.
StartTime	No	Timestamp ISO8601	Start time, such as `2021-07-15T19:51:23+08:00`.
EndTime	No	Timestamp ISO8601	End time, which is the current time by default, such as `2021-07-15T20:51:23+08:00`. `EndTime` cannot be earlier than `StartTime`.

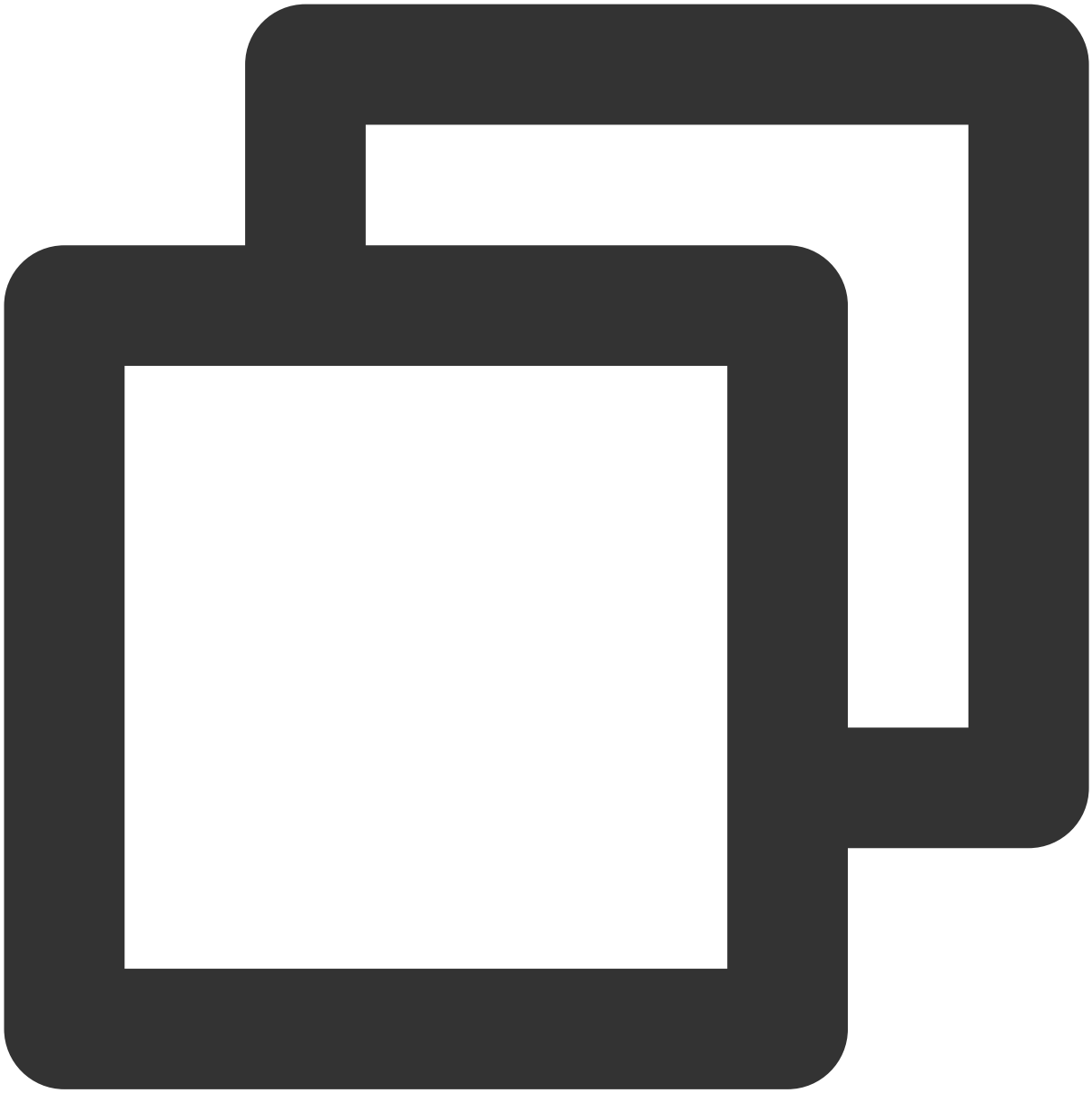
Output parameters

Parameter Name	Type	Description
Period	Integer	Statistical period
MetricName	String	Metric name
DataPoints	Array of DataPoint	Array of data points
StartTime	Timestamp ISO8601	Start time
EndTime	Timestamp ISO8601	End time
RequestId	String	The unique request ID, which is returned for each request and is required for troubleshooting.

Examples

This example shows you how to get the 5-minute CPU utilization monitoring data of a TDSQL-C for MySQL instance in a specific period of time.

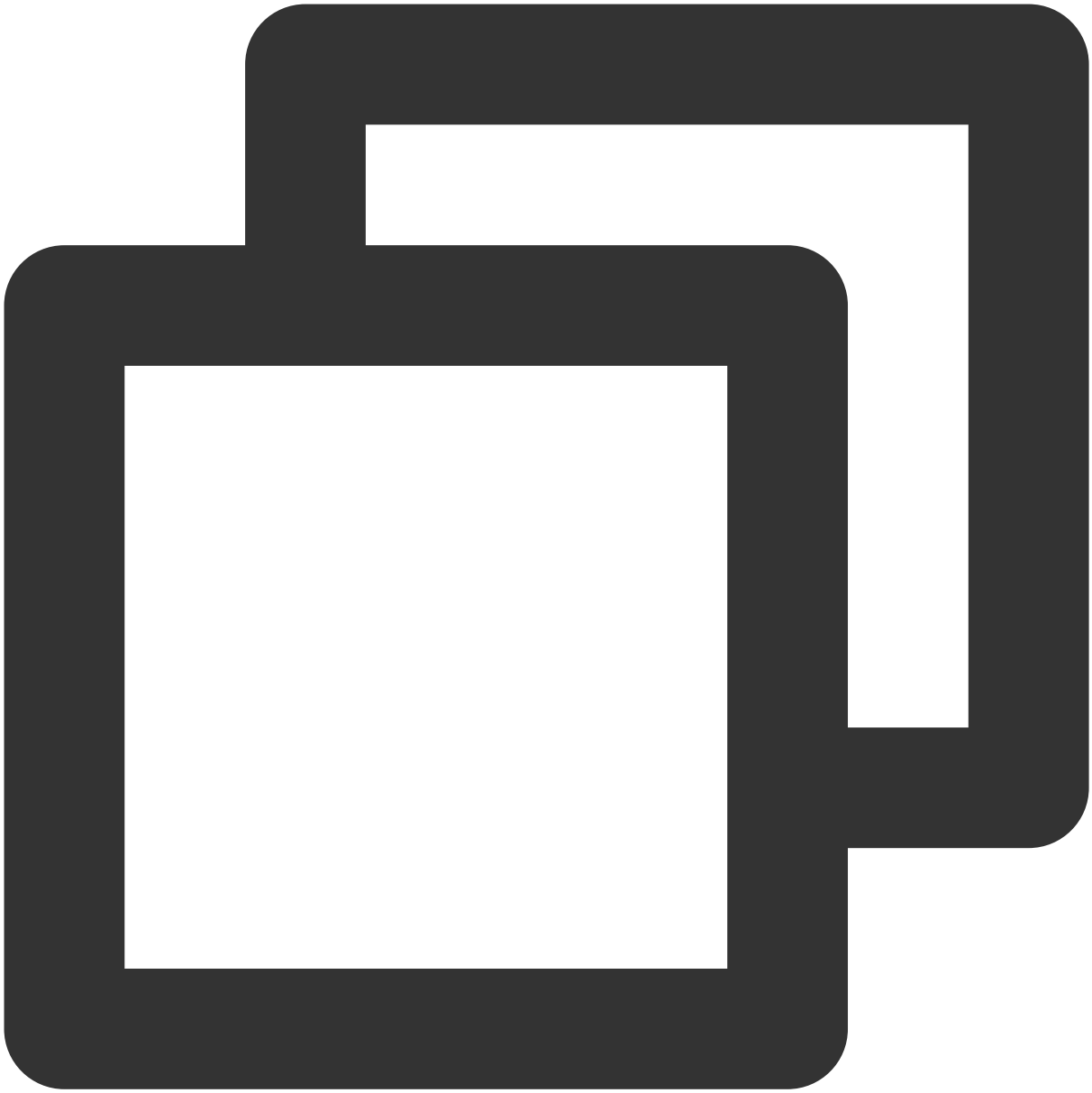
Sample input



```
POST / HTTP/1.1
Host: monitor.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: GetMonitorData
<Common request parameters>

{
    "Namespace": "QCE/cynosdb_mysql",
    "MetricName": "CPuUsageRate",
    "Period": 3600,
    "Instances": [
```

```
        {
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "cynosdbmysql-ins-edpn3t6b"
                }
            ]
        }
    ],
    "StartTime": "2022-07-15T10:00:00",
    "EndTime": "2022-07-15T15:00:00"
}
```

Sample output




```
{
  "Response": {
    "DataPoints": [
      {
        "Dimensions": [
          {
            "Name": "InstanceId",
            "Value": "cynosdbmysql-ins-edpn3t6b"
          }
        ],
        "Timestamps": [
```

```
            1657850400,
            1657854000,
            1657857600,
            1657861200,
            1657864800
          ],
          "Values": [
            0.26,
            0.24,
            0.23,
            0.26,
            0.24
          ]
        }
      ],
          "EndTime": "7/15/2022 3:00:00 PM",
          "MetricName": "CPuUsageRate",
          "Period": 3600,
      "RequestId": "71c72744-bec5-49d0-b42c-433609ab4166"
           "StartTime": "7/15/2022 10:00:00 AM"
    }
}
```

Alarms
Monitoring Metric Alarm
Alarm Configuration Overview

Last updated：2023-08-22 15:50:48

TDSQL-C for MySQL allows you to set alarms for real-time performance metrics, so that you can monitor the resource usage and running status of databases. You can configure alarm notifications to receive alarms on a timely basis and view alarm records to review alarm information in order to optimize the overall performance of your database.

Setting an alarm

You can get alarm notifications and locate problems in the following steps. This helps you handle database exceptions promptly, increase the system stability, improve the Ops efficiency, and reduce Ops costs.

Steps	Description	Operation
Step 1. Set an alarm policy	Set an alarm policy to trigger alarms and send messages when the TDSQL-C for MySQL cluster status changes.	Setting Alarm Policy
Step 2. Set an alarm notifications	Add recipients and set the time period and channels for alarm notifications.	Setting Alarm Notification
Step 3. View alarm records	View alarm notifications generated in the past half year.	Viewing Alarm Records

Setting Alarm Policy

Last updated：2023-08-24 10:03:40

This document describes how to create and manage an alarm policy in the console.

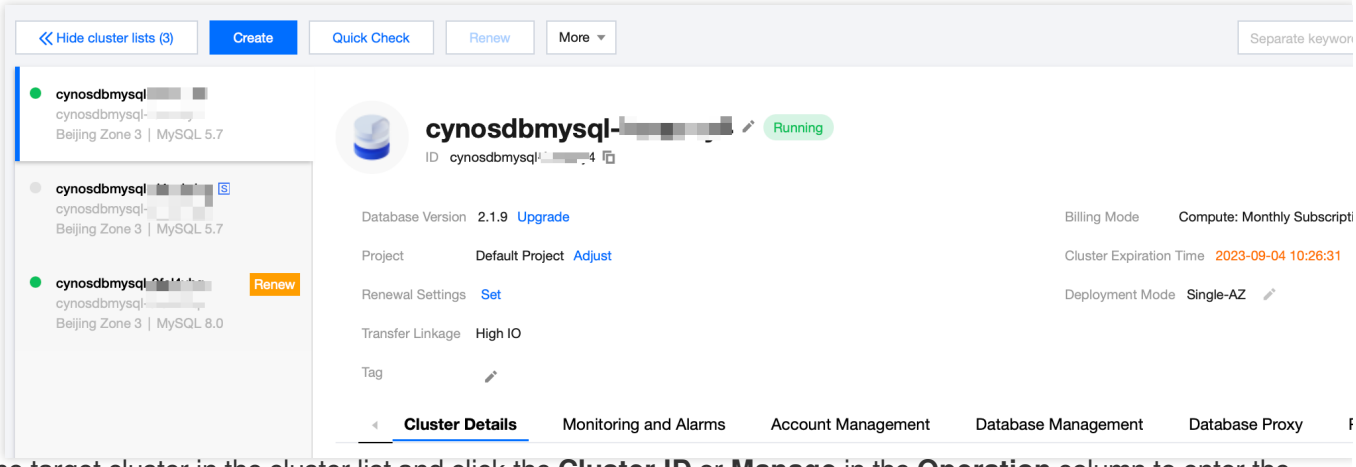
Overview

You can create alarm policies to trigger alarms and send alarm notifications when the TDSQL-C for MySQL status changes. The created alarm policies can determine whether an alarm needs to be triggered according to the difference between the monitoring metric value and the given threshold at intervals.
When the alarm is triggered by a status change, you can promptly take the necessary preventative or corrective action. Alarm policies can therefore, if appropriately created, help you improve the robustness and reliability of your database. For more information on alarms, see Creating Alarm Policy.
To send an alarm for a specific status of TDSQL-C for MySQL, you need to create an alarm policy at first. An alarm policy is composed of three compulsory components, that is, the name, type and alarm triggering conditions. Each alarm policy is a set of alarm triggering conditions with the logical relationship "OR", that is, as long as one of the conditions is met, an alarm will be triggered. The alarm will be sent to all users associated with the alarm policy. Upon receiving the alarm, the user can view the alarm and take appropriate actions in time. After creating an alarm policy, you can change its name, remarks, or triggering conditions and edit its alarm objects.

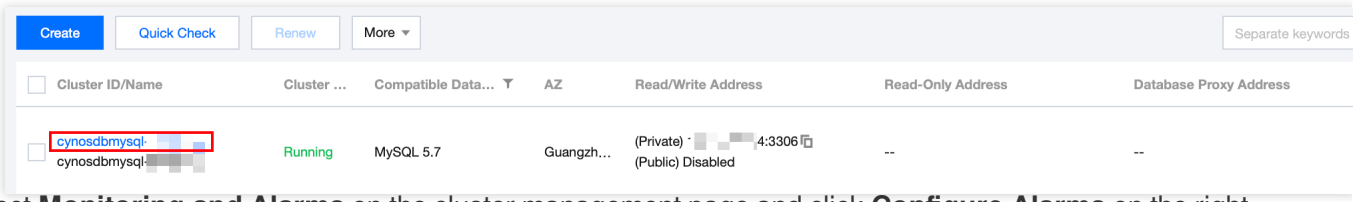
Directions

Setting an alarm policy

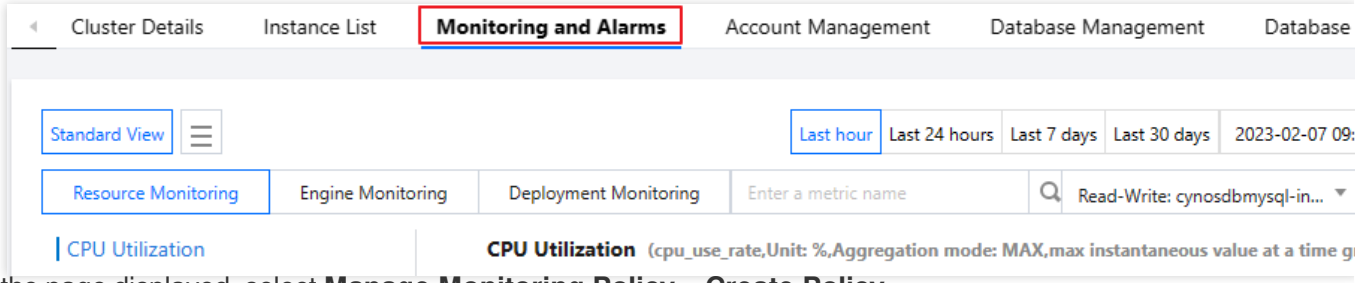
1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top and proceed according to the actually used view mode:
Tab view
List View
Click **Target Cluster** in the cluster list on the left to enter the cluster management page.



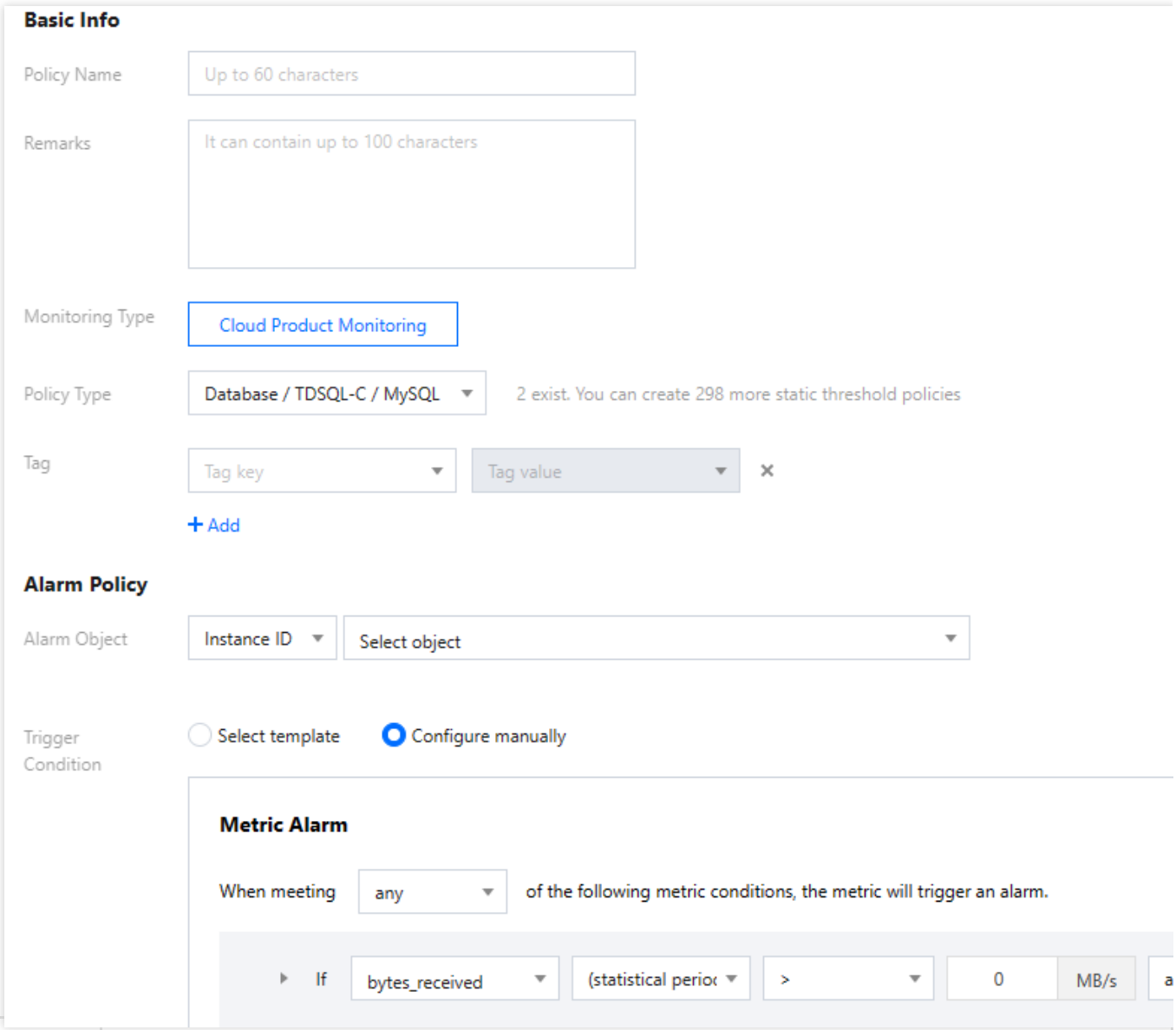
Find the target cluster in the cluster list and click the **Cluster ID** or **Manage** in the **Operation** column to enter the cluster management page.



3. Select **Monitoring and Alarms** on the cluster management page and click **Configure Alarms** on the right.



4. On the page displayed, select **Manage Monitoring Policy** > **Create Policy**.
5. On the policy creation page, configure the following parameters and click **Complete**.

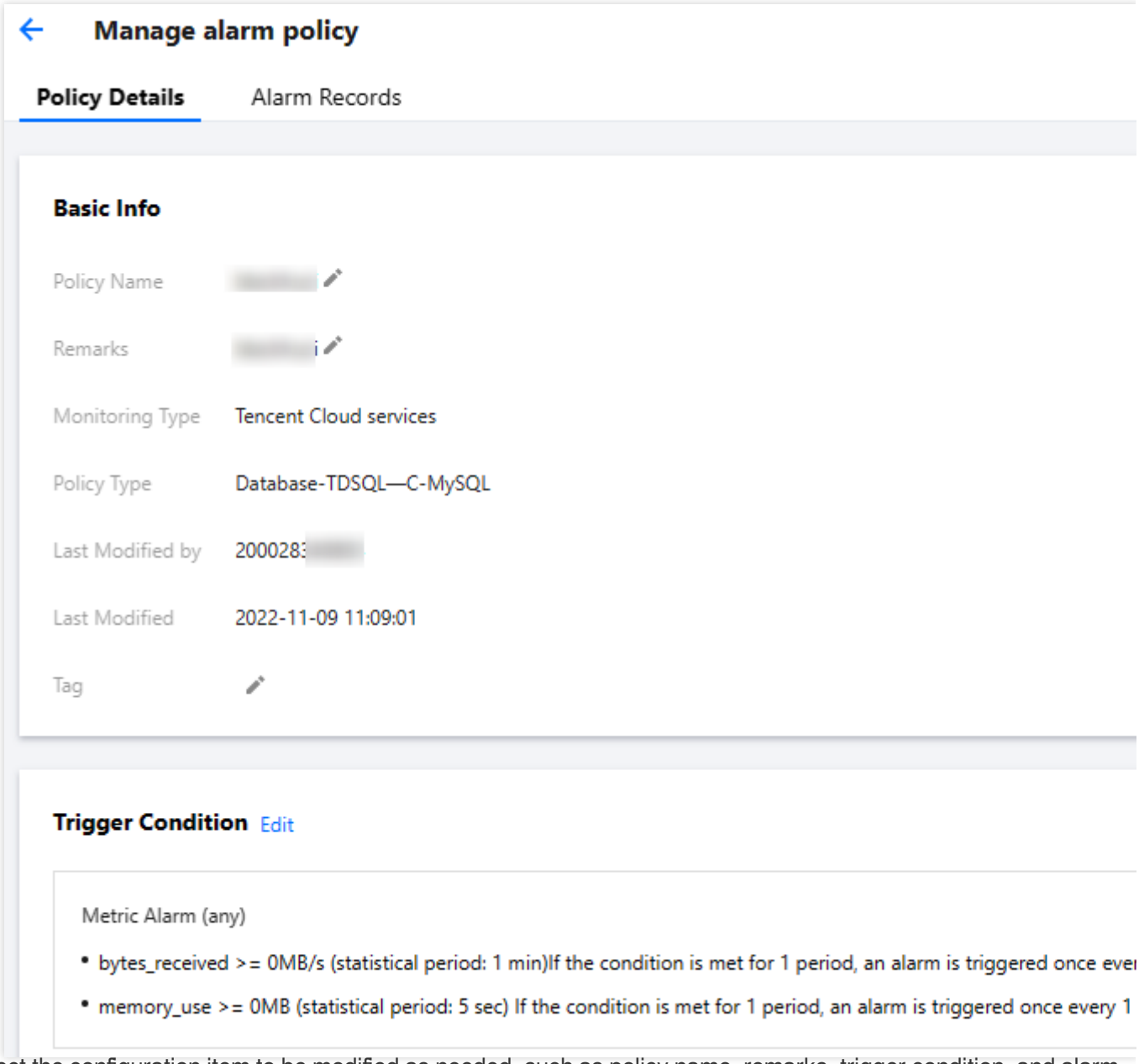


Parameter	Description
Policy Name	Enter the policy name, which can contain up to 60 characters.
Remarks	Enter remarks for the policy, which can contain up to 100 characters.
Monitoring Type	Select **Tencent Cloud services**.
Policy Type	Select **TencentDB** >**TDSQL-C** > **MySQL**.
Alarm Object	Find the object instance to be associated with by selecting the region where the object is located or searching for the instance ID of the object.
Trigger	Select a predefined template or manually configure the alarm trigger condition, which is a

Condition	semantic condition composed of metric, comparison, threshold, statistical period, and duration. For example, if the metric is disk utilization, the comparison is >, the threshold is 80%, the statistical period is 5 minutes, and the duration is two statistical periods, then the data on disk utilization of a database will be collected once every five minutes, and an alarm will be triggered if the disk utilization exceeds 80% for two consecutive times.
Configure Alarm Notification	Select a preset or custom notification template. Each alarm policy can be bound to three notification templates at most. For more information, see Creating Notification Template.

Managing an alarm policy

1. In the alarm policy list, click the name of the target alarm policy to enter the alarm policy management page.
2. Select **Policy Details** on the alarm policy management page.



3. Select the configuration item to be modified as needed, such as policy name, remarks, trigger condition, and alarm object.

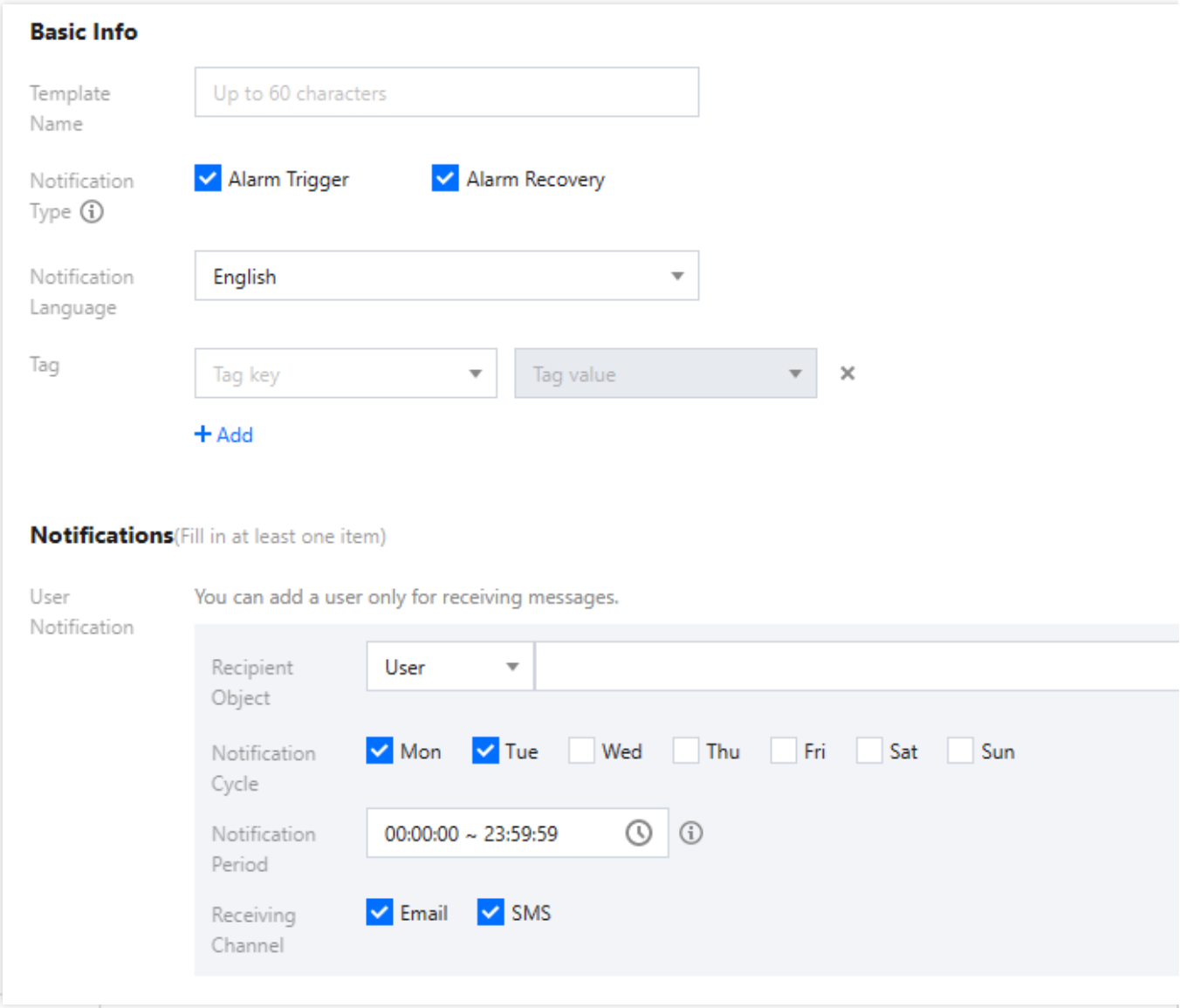
Setting Alarm Notification

Last updated：2023-08-22 15:22:00

After setting an alarm policy, you need to add alarm recipients, which can be done during alarm notification configuration.
This document describes how to set an alarm notification template in the console. When creating an alarm policy on the alarm policy configuration page, you can select an existing template or create a new one for easy configuration. You can also quickly apply the same template for multiple policies.
For more information on relevant limits and other descriptions, see Creating Notification Template.

Setting an alarm notification template

1. Log in to the TCOP console and select **Notification Template** on the left sidebar.
2. Click **Create Notification Template**, configure the following items on the page displayed, and click **Complete**.



Parameter	Description
Template Name	Enter the notification template name, which can contain up to 30 characters.
Notification Type	Select one or multiple notification types. Options include **Alarm Trigger** and **Alarm Recovery**.
Notification Language	Select the notification language, which can be Chinese or English.
Recipient Object	Select users or user groups to receive notifications.
Notification Cycle	Select the notification cycle, which can be any number of days in a week.
Notification	Set the notification period. Alarm notifications will be sent at the start time in the notification cycle

Period	and will be stopped at the end time.
Receiving Channel	Select email, SMS, etc.
API URL	Enter a URL (domain name or IP[:port][/path]), WeCom group chatbotwebhook, DingTalk group chatbot webhook, or Slack group application webhook accessible over the public network, and TCOP will push alarm messages to the corresponding URL, WeCom group, DingTalk group, or Slack group (if the recipient is a user, there is no need to add an API callback).

Note:

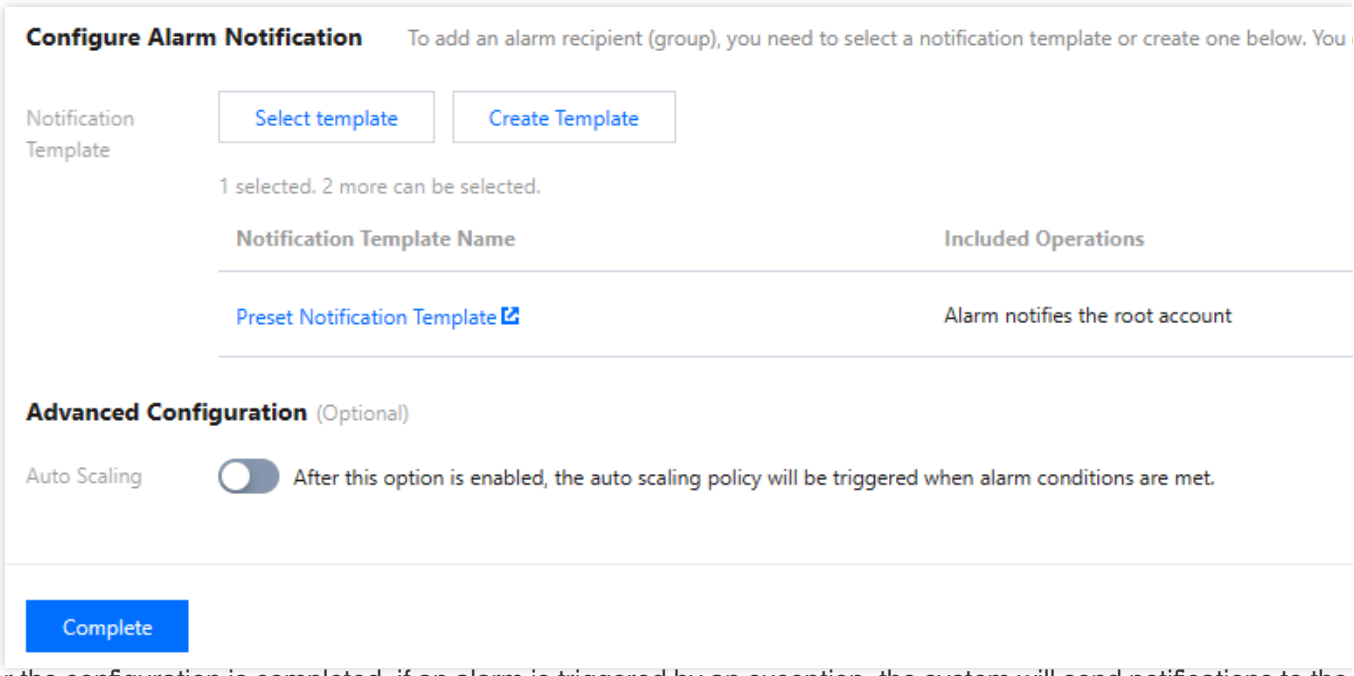
You need to enter the API callback only when pushing notifications to WeCom group chatbots, DingTalk group chatbots, or Slack group applications.

After the API callback is added, if the initial delivery of an alarm notification fails, CM attempts up to three retries with a delay between failed attempts set at 5 seconds.

After the API callback is added, each WeCom or DingTalk group chatbot can receive up to 20 notifications per minute.If this quota is exceeded, the excess notifications will be discarded for WeCom, and the traffic will be throttled for 10 minutes for DingTalk.

Configuring an alarm notification

1. Log in to the TCOP console and select **Policy Management** on the left sidebar.
2. Click **Create Policy** and select up to three alarm notification templates in **Configure Alarm Notification** in the **Configure Alarm Policy** window.
3. Click **Complete** after configuring the alarm policy and notification.



4. After the configuration is completed, if an alarm is triggered by an exception, the system will send notifications to the recipients via the specified channels (email, SMS, WeCom, and phone call).

Viewing Alarm Records

Last updated：2023-02-07 16:52:30

You can view alarms triggered in the past half year in the console to quickly locate specific problems for further troubleshooting. You can also analyze the alarm records to stay informed of and improve the database status.
This document describes how to view alarm records in the console. For more information on feature description, see Viewing Alarm Records.

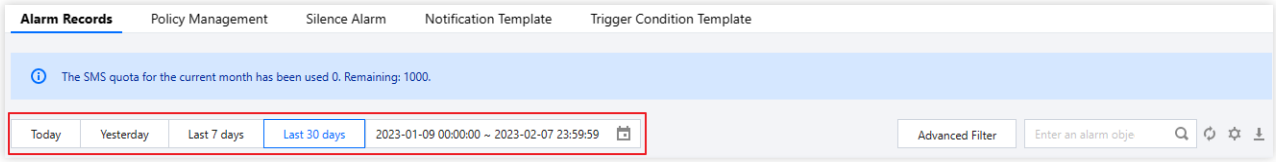
Directions

1. Log in to the CM console, and select **Alarm List** on the left sidebar.
2. You can view alarm records in the alarm list.

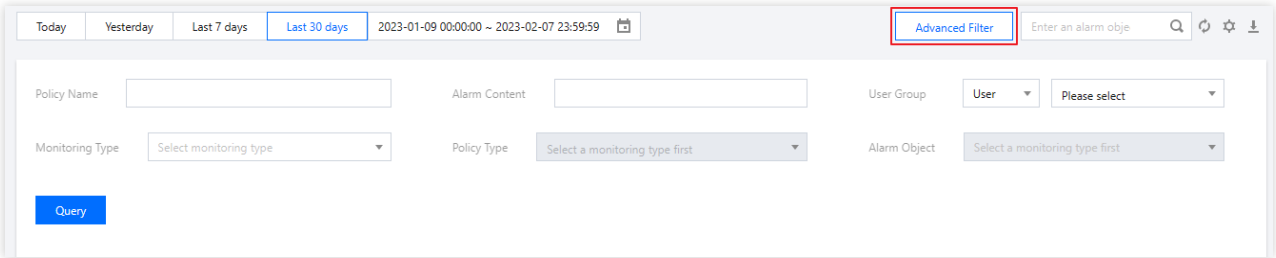
Note:

You can click the

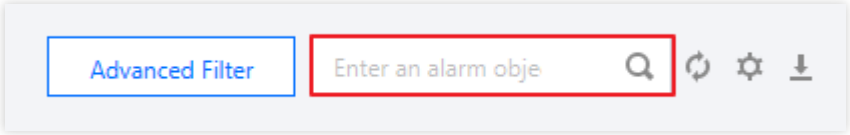
icon to customize the displayed fields of alarm records in the list, including occurrence time, monitoring type, policy type, alarm object, alarm content, duration, alarm status, policy name, end time, alarm type, alarm reception, alarm channel, instance group, project, and network.
Select a time or a custom time range to view the target alarm records.



Click **Advanced Filter** to view the target alarm records.



Enter alarm object in search box to view the target alarm records.



Event Alarm
Overview

Last updated：2023-02-15 15:12:09

Overview

EventBridge is a secure, stable, and efficient serverless event management platform. As a pipeline for automatic collection, processing, and distribution of flow data and events, it enables you to monitor exceptions in TDSQL-C for MySQL by visually configuring event rules. This helps you stay up to date with application status for timely optimization and Ops.

Features

Event collection: EventBridge provides a standard event delivery interface for standardized connection of different event sources, such as Tencent Cloud, SaaS, and custom services.
Event management: EventBridge provides event management capabilities such as event format match, content filtering, format conversion, tracking, archiving, and replay to better support users in event-driven architecture (EDA).
Event delivery: EventBridge supports connections to different types of targets with high scalability and can provide different solutions based on actual business scenarios.

Concepts

See Basic Concepts.

Restrictions

See Limits.

Supported Event Types

Last updated：2023-02-23 10:55:13

This document describes the event types in TDSQL-C for MySQL for which alarms can be reported.

Event	Event Name	Description
Memory OOM	GuestOom	The system memory was overloaded.
Instance read-only (excessive storage)	DiskOverQuota	The used storage space exceeded the limit, so data could only be read but not written.
Database agent mount node removal	ProxyNodeFaultEliminated	A database proxy mount node was removed due to a failure.
Database agent exception	ProxyUnHealthy	An exception occurred in the database proxy.
HA (instance high availability)	HA	An HA switch occurred in the instance.
Version upgrade	UpgradeVersion	The minor version was upgraded in the cluster.
Cross-AZ latency	CrossAzReplicationLag	The cross-AZ source-replica latency was too high.
TencentCloud API operation (CloudAudit)	ApiCall	A TencentCloud API was called.
Console operation (CloudAudit)	ConsoleCall	An operation was performed in the console.

Setting Event Alarms

Last updated：2023-09-12 14:59:48

TDSQL-C for MySQL supports the event alarming feature. After you set an event rule, target, and recipient, if this feature detects an abnormal event, it will send alarm notifications to the recipient through various channels. This document describes how to set an event alarm.

Step 1. Activate EventBridge

EventBridge uses CAM to manage permissions. CAM is a permission and access management service that helps you securely manage the access permissions of resources under your Tencent Cloud account. With CAM, you can create, manage, and terminate users and user groups and use identity and policy management to control user access to Tencent Cloud resources. To use EventBridge with a root account or sub-account, you need to activate it first as instructed in Activating EventBridge.

Step 2. Configure event alarm push for TDSQL-C for MySQL

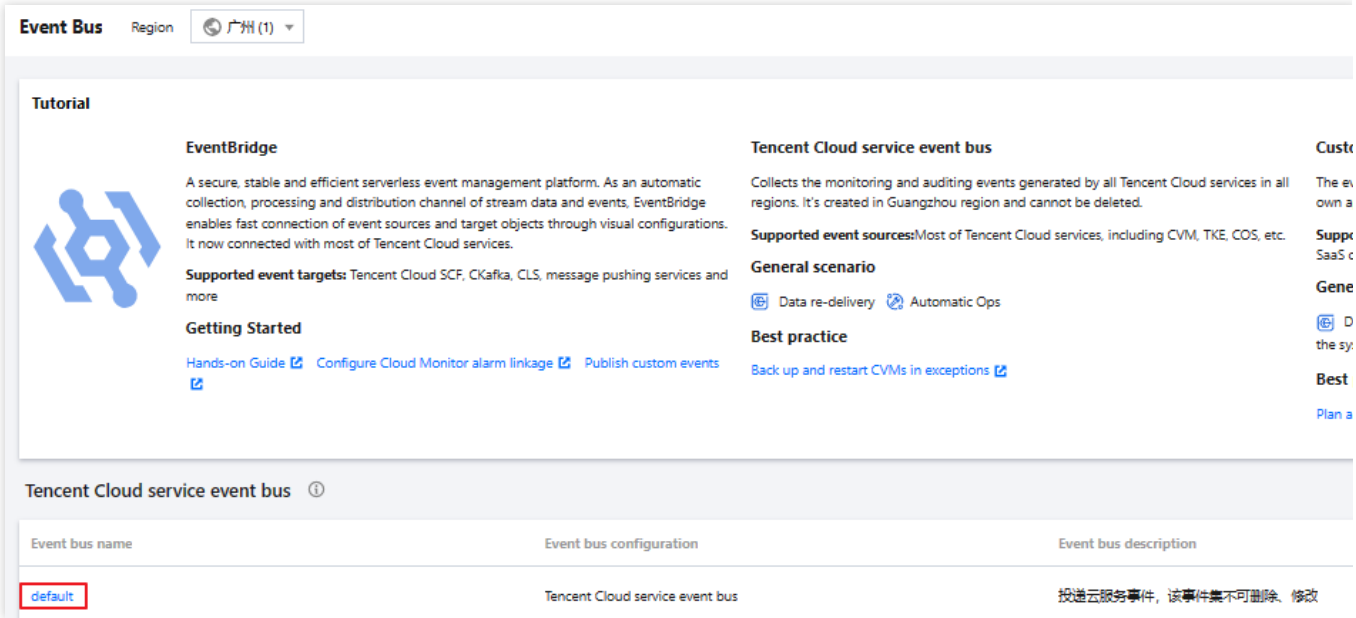
After activating EventBridge, you need to select the event source connection method. Currently, you can select TDSQL-C for MySQL monitoring events as an event source.

Note:

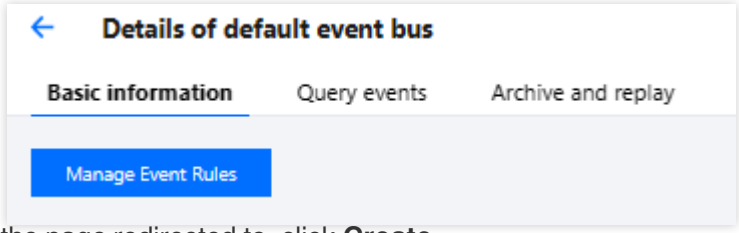
Ops events generated in TDSQL-C for MySQL such as alarms and audits will all be published to the **Tencent Cloud service event bus** by default, which cannot be changed or edited.

After EventBridge is activated, it will automatically create a **default Tencent Cloud service event bus** in **Guangzhou** region, to which alarm events (monitoring and audit events) generated by TDSQL-C for MySQL will be automatically published.

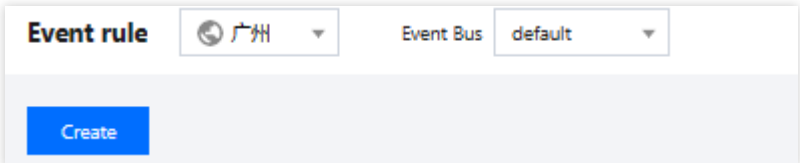
1. Log in to the EventBridge console.
2. Select the **Guangzhou** region at the top.
3. Click the **default** event bus below **Tencent Cloud Service Event Bus**.



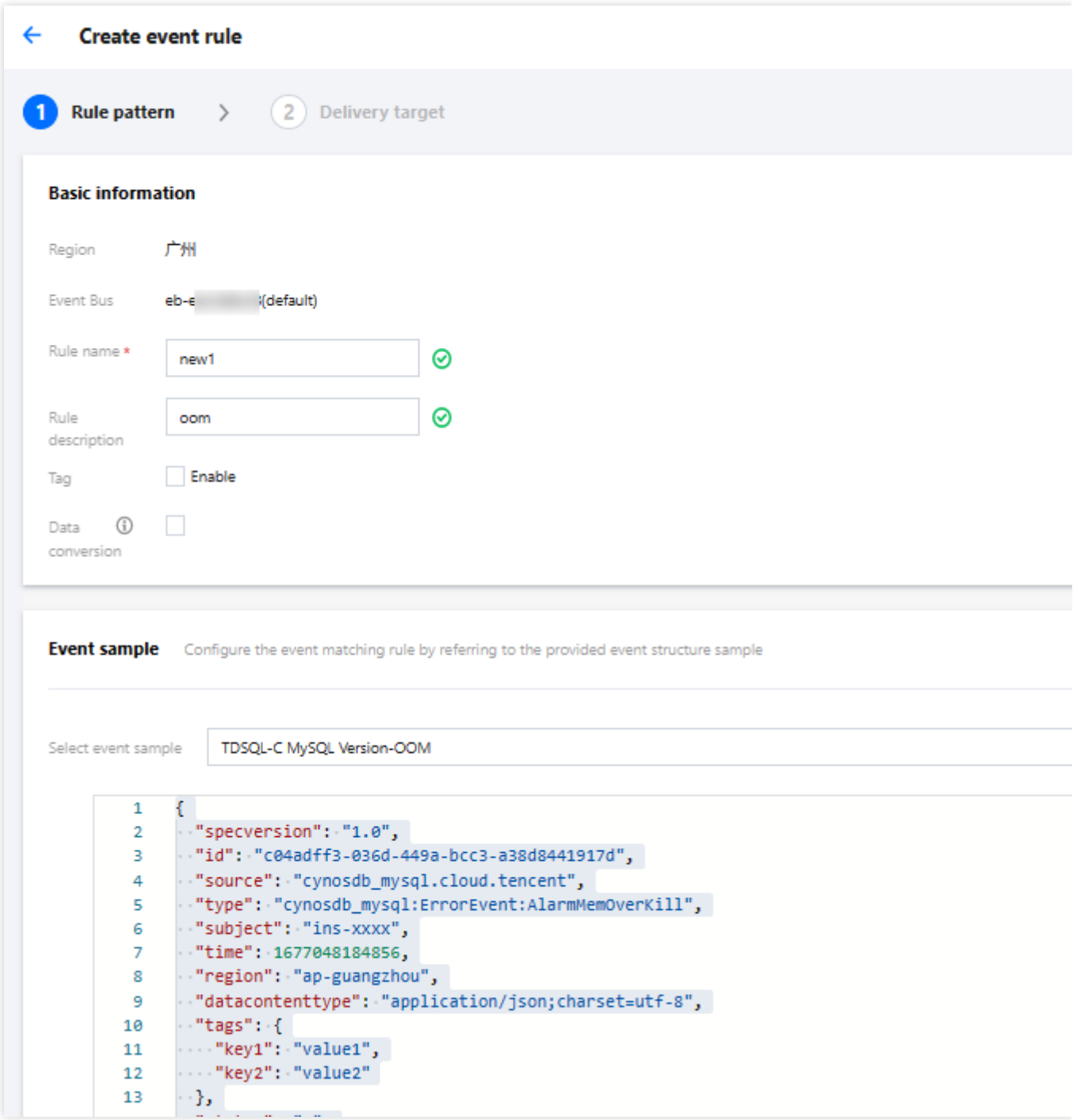
4. On the default event bus details page, click **Manage Event Rules**.

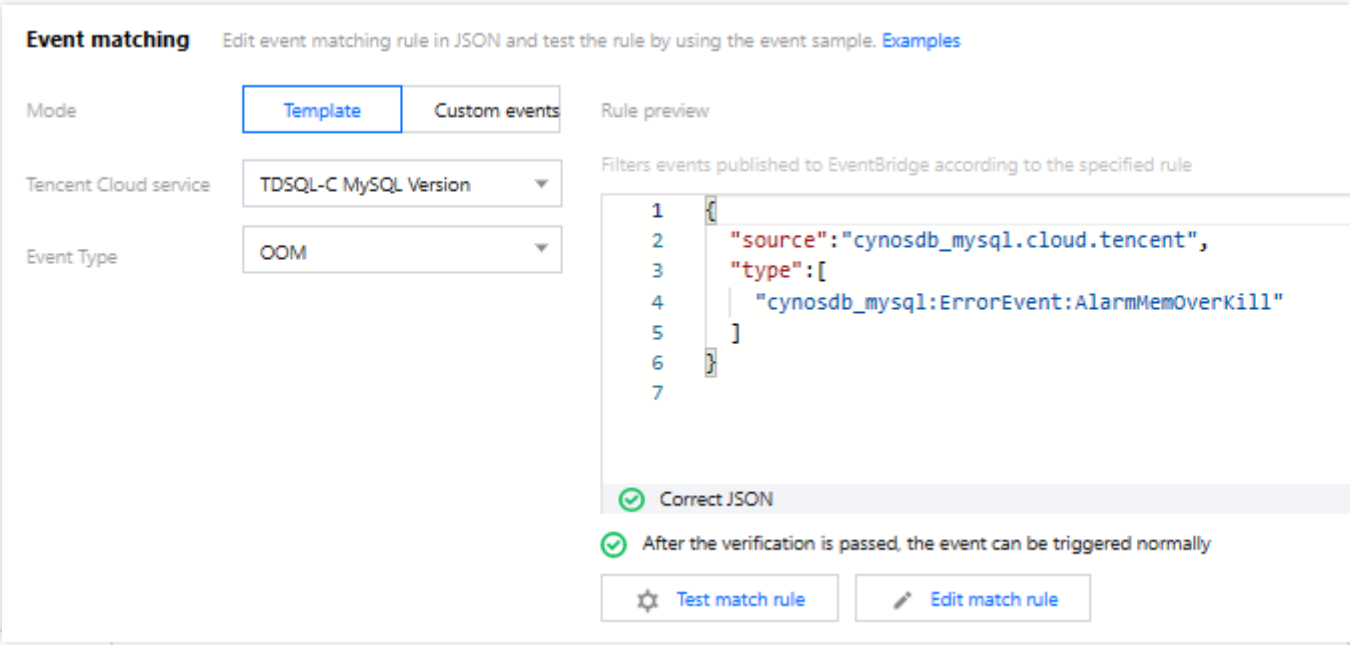


5. On the page redirected to, click **Create**.



6. On the **Create event rule** page, set the following configuration items and click **Next**.

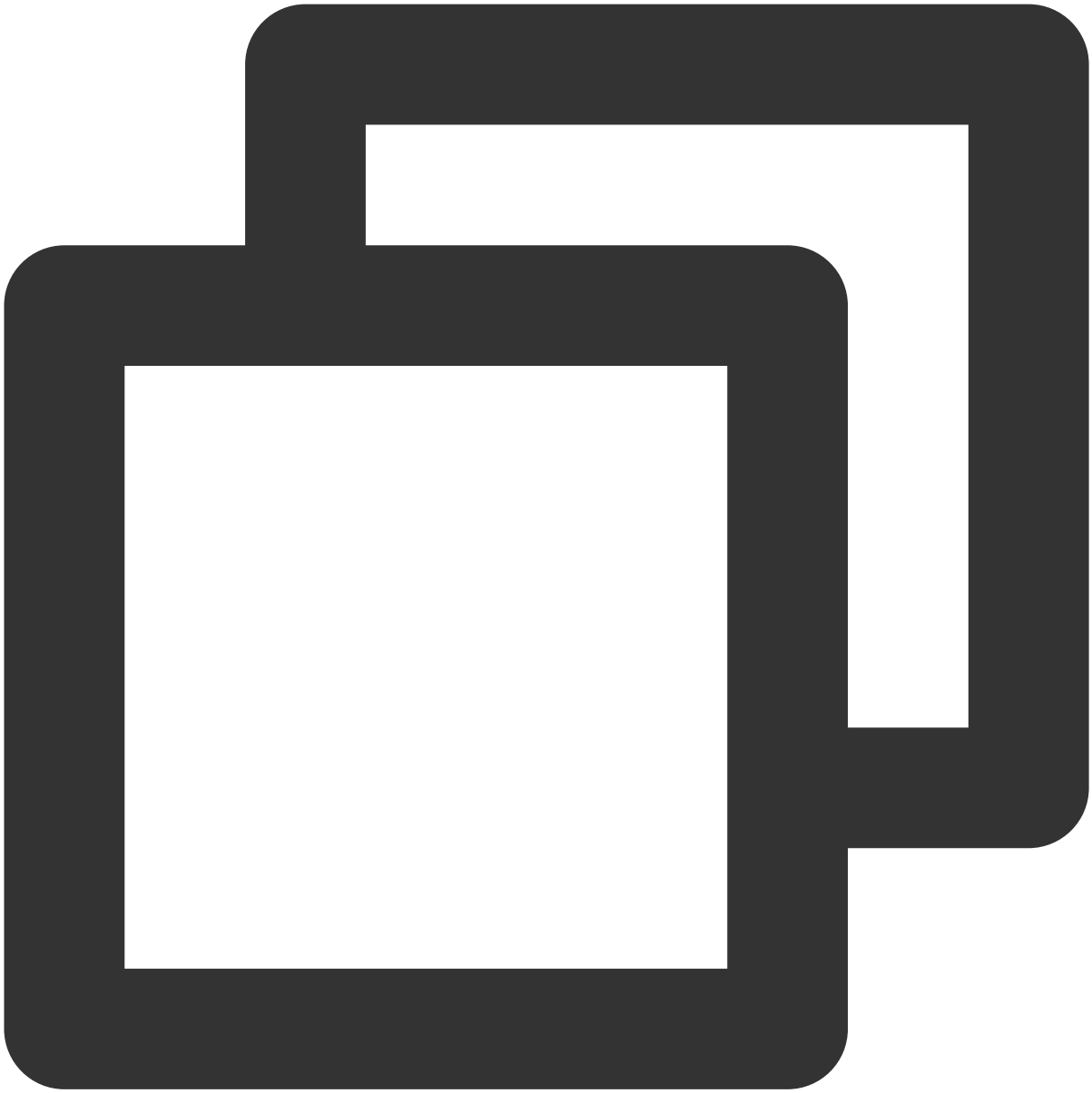




Parameter	Description
Rule name	Enter the rule name, which can contain 2–60 letters, digits, underscores, and hyphens. It must start with a letter and end with a digit or letter.
Rule description	Enter the rule description, which can contain up to 200 digits, letters, and symbols.
Tag	Select whether to enable tag. After tag is enabled, you can add tags to the event rule.
Data conversion	Data conversion processes the content of the event (extracting, parsing, and re-mapping fields) before the event is published to the target.
Sample event	A sample event structure is provided as a reference for configuring the event match rule. You can find the target template below.
Write mode	Form mode (recommended) and custom event are supported.
Tencent Cloud service	Select TDSQL-C for MySQL.
Event type	Select one or multiple alarm event types. For more information, see Supported Event Types.
Test match rule	Select the event type template selected in **Sample event** and click **Test match rule**. If the test is passed, proceed to the next step.

Note:

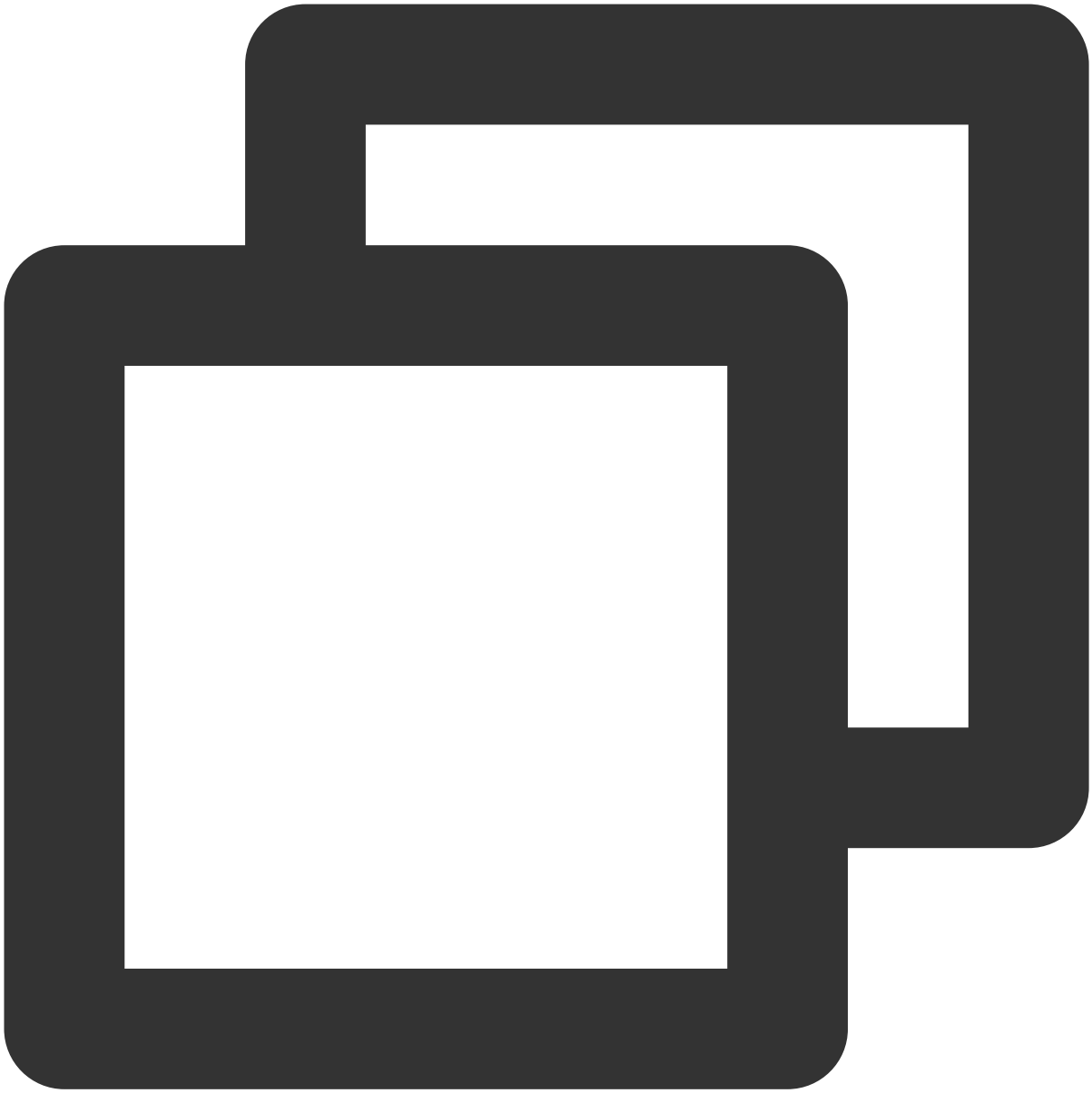
To receive events from the specified instance, configure the following rule:



```
{
  "source":"cynosdb_mysql.cloud.tencent",
  "subject":"ins-xxxxxx"
}
```

This rule indicates to receive and publish events of the TDSQL-C for MySQL instance `ins-XXX` only. Other events will be discarded.

You can also specify multiple resources by using an array.



```
{
  "source":"cynosdb_mysql.cloud.tencent",
  "subject":["ins-xxxxxx","ins-xxxxxx"]
}
```

7. On the **Delivery target** tab, set the following configuration items, select **Enable event rules now** and click **Complete**.

Parameter	Description
Trigger	Select **Notification message**.

method	
Message template	You can select **Monitoring alert template** or **General notification template**.
Alarm content	You can select Chinese or English.
Notification method	You can select **API callback**, **Publishing channel**, or **All methods**. Here, **Publishing channel** is selected for subsequent settings.
Recipients	Select users or user groups to receive notifications.
Notification period	Customize the time period for receiving notifications.
Delivery method	Select one or multiple delivery methods. An SMS message can contain up to 500 characters, and a phone call can contain up to 350 characters in the script. Events with excessive characters (probably caused by long instance names) won't be pushed. We recommend that you configure multiple methods.

Note:

To configure multiple event targets, click **Add**.

8. After successful creation, you can query and manage the event rule in the event rule list.

Basic SQL Operations

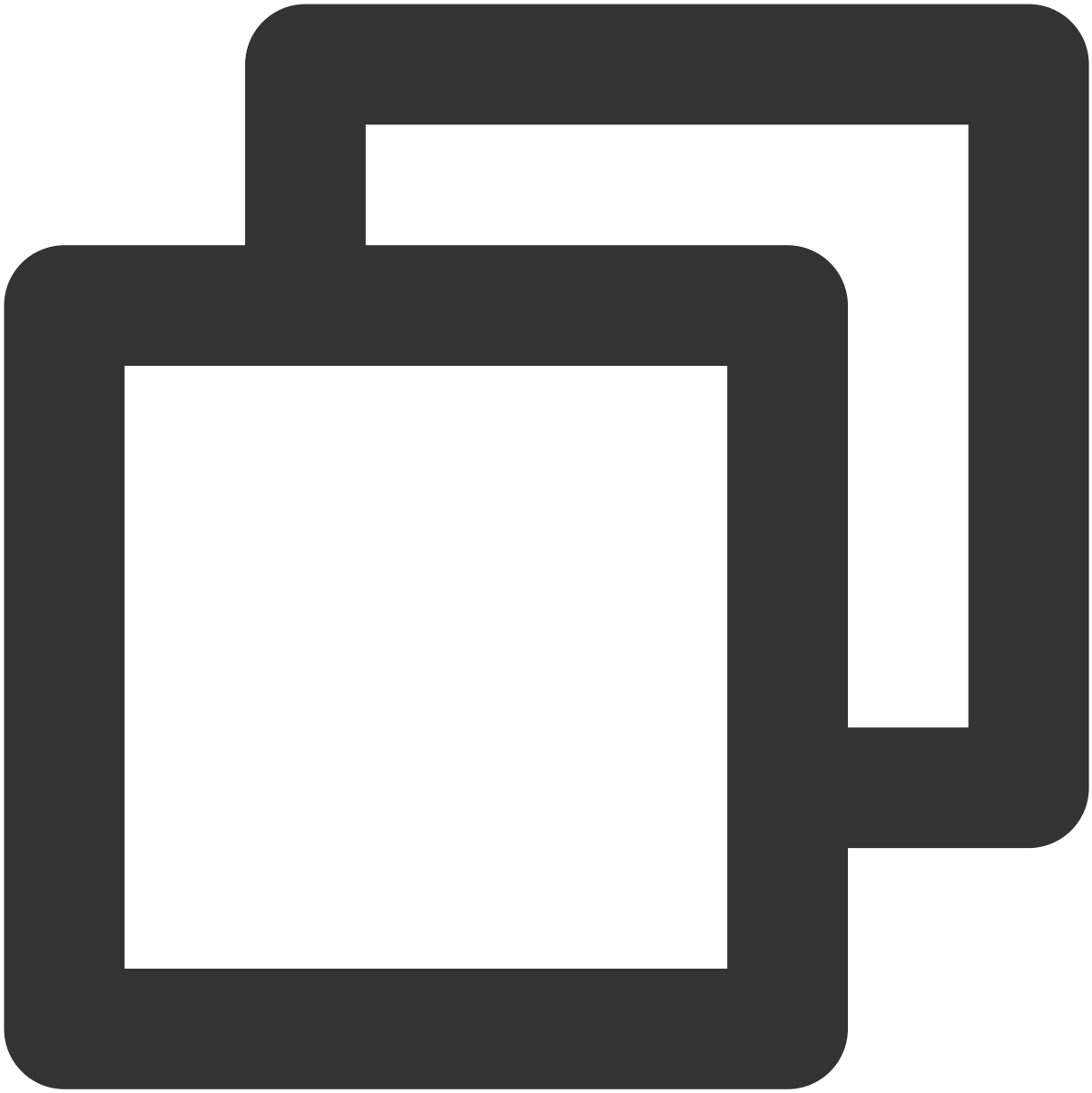
Last updated：2023-11-09 15:02:18

This document describes common SQL commands.

For more information on SQL commands, including command parameters and restrictions, see MySQL 5.7 Reference Manual.

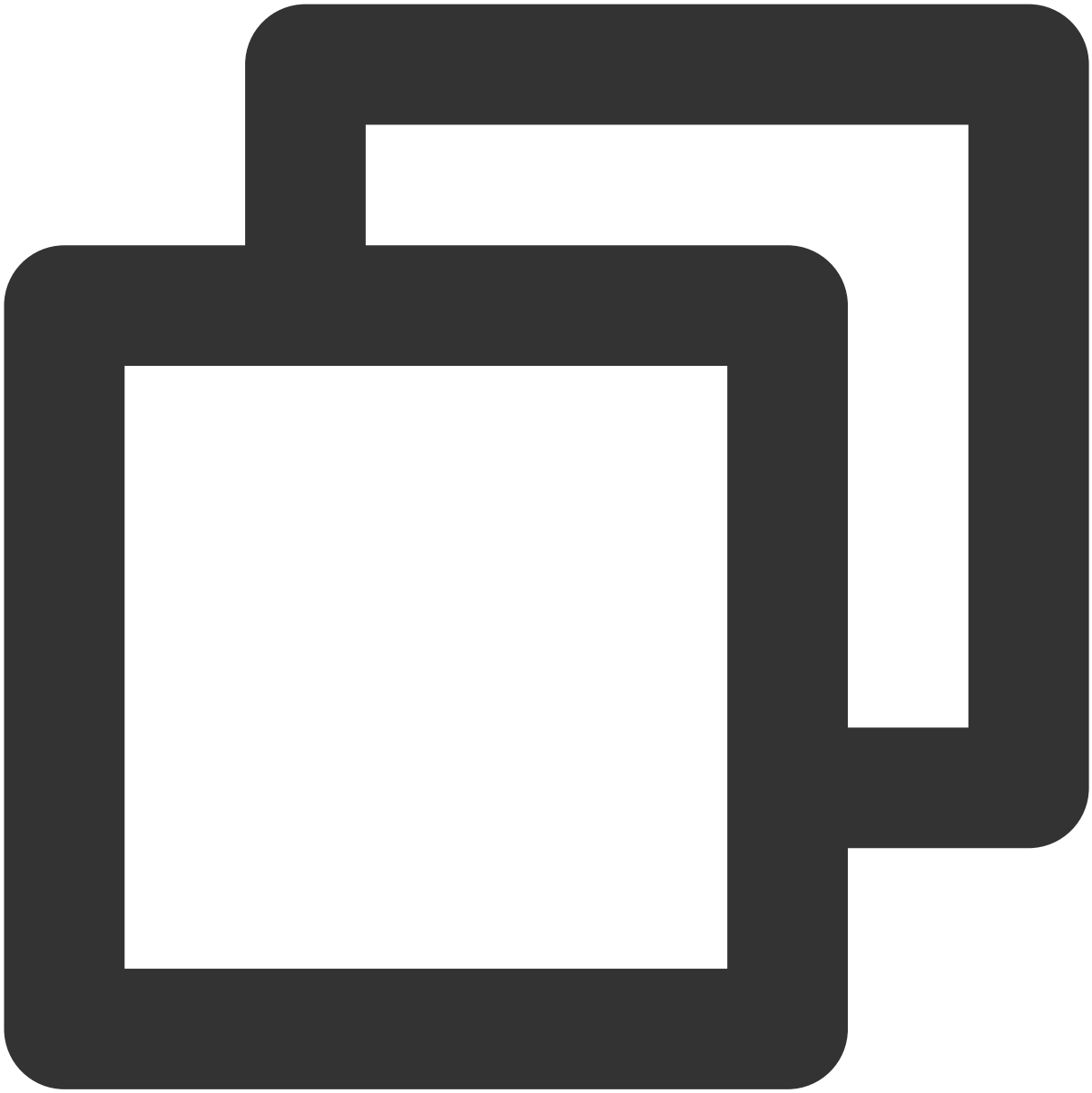
Querying the version

Option 1:



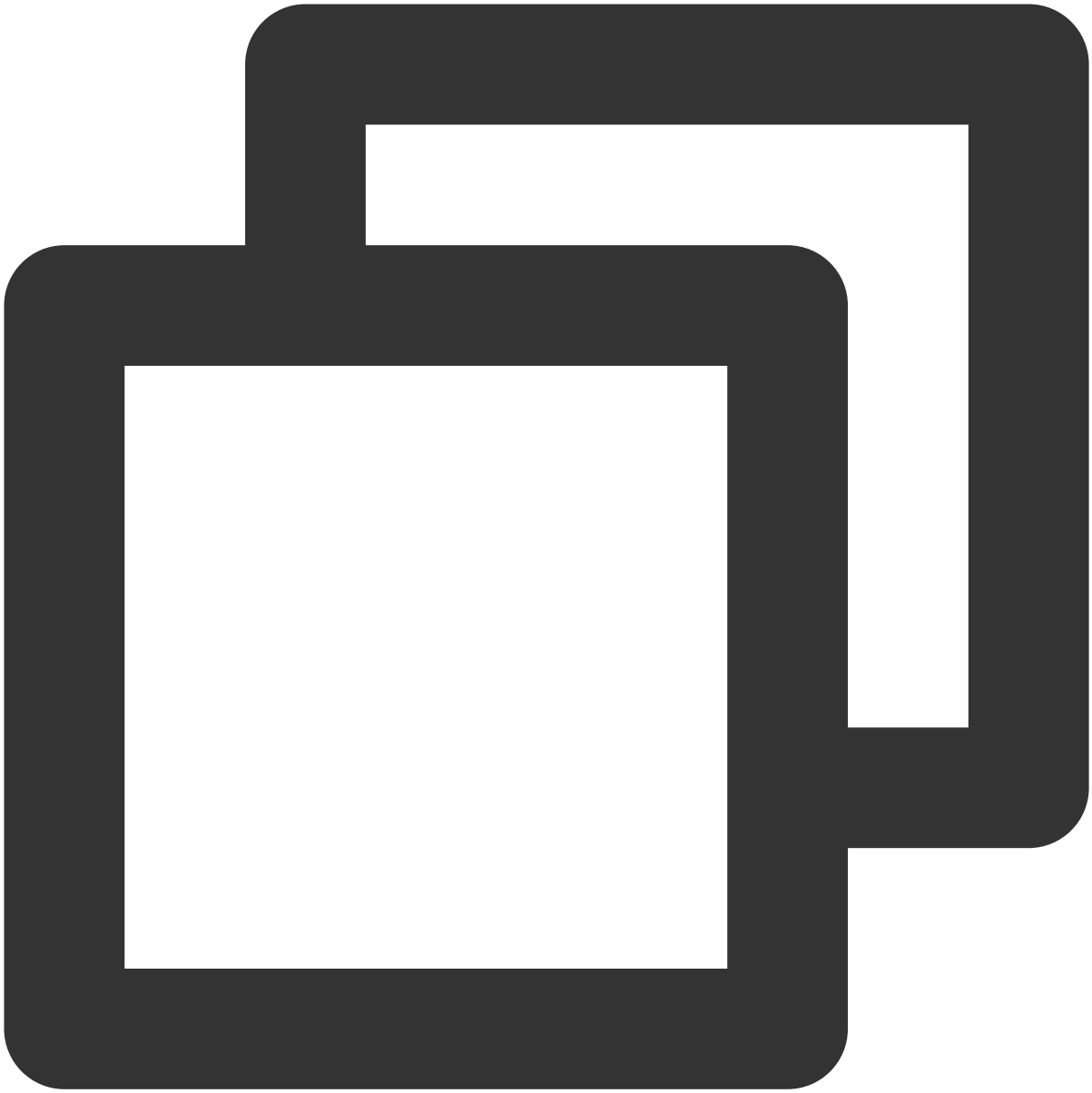
```
MySQL [(none)]> SELECT CYNOS_VERSION();
+-------------------------+
| CYNOS_VERSION() |
+-------------------------+
| 5.7.mysql_cynos.1.3.10 |
+-------------------------+
1 row in set (0.00 sec)
```

Option 2:



```
MySQL [(none)]> SELECT @@CYNOS_VERSION;
+------------------------+
| @@CYNOS_VERSION |
+------------------------+
| 5.7.mysql_cynos.1.3.10 |
+------------------------+
1 row in set (0.00 sec)
```

Option 3:



```
MySQL [(none)]> SHOW VARIABLES LIKE 'CYNOS_VERSION';
+---------------+-----------------------+
| Variable_name | Value |
+---------------+-----------------------+
| cynos_version | 5.7.mysql_cynos.1.3.10 |
+---------------+-----------------------+
1 row in set (0.01 sec)
```

Database commands

Command	Sample
Create a database and specify the character set	`CREATE DATABASE db01 DEFAULT CHARACTER SET gbk COLLATE gbk_chinese_ci;`
Delete a database	`DROP DATABASE db01;`

Account commands

Command	Sample
Create an account	`CREATE USER 'username'@'host' IDENTIFIED BY 'password';`
Delete an account	`DROP USER 'username'@'host';`
Grant permissions	`GRANT SELECT ON db01.* TO 'username'@'host';`
Query an account in the database	`SELECT user,host FROM mysql.user;` or `SHOW GRANTS FOR 'username'@'hostname';`
Revoke permissions	Revoke all permissions: `REVOKE ALL PRIVILEGES,GRANT OPTION FROM 'username'@'host';` Revoke specified permissions: `REVOKE UPDATE ON *.* FROM 'username'@'host';`

Connecting to TDSQL-C for MySQL Through SCF

Last updated：2023-03-01 14:33:46

If you need to use TDSQL-C for MySQL in SCF, you can use the connection pool or the SDK provided by the SCF team for connection. The connection pool supports automatic reconnection to effectively avoid connection unavailability due to connection release by the SCF underlying layer or database.
For detailed directions, see Connecting SCF to Database.

Tag
Tag Overview

Last updated：2023-01-04 10:12:50

A tag is a key-value pair provided by Tencent Cloud and can be used to easily identify resource. For more information, see Tag Overview.
You can use tags to categorize and manage TDSQL-C for MySQL resources in various dimensions such as business, purpose, and person-in-charge, so you can find the target resources quickly. Tags have no semantic meaning in Tencent Cloud. They are parsed and matched strictly based on strings. When using tags, you need to pay attention to the use limits.
The following example shows how a tag is used.

Example

Background
A company has 5 TDSQL-C for MySQL clusters in Tencent Cloud, which are owned by three departments: Ecommerce, Game, and Entertainment. The clusters are used for services such as marketing, gaming, and post-production. The Ops owners of the three departments are John, Jane, and Harry, respectively.
Setting a tag
For more efficient management, the company uses tags to categorize the database resources, and defines the following tag key-value pairs:

Tag Key	Tag Value
Department	Ecommerce, Game, and Entertainment
Business	Marketing, Gaming, and Post-production
Ops owner	John, Jane, and Harry

Then bind the above tag key-value pairs to TDSQL-C for MySQL. The relationship between resources and the tag key-value pairs is shown in the table below.

instance-id	Department	Service	Ops owner
ceshi-abc1	Ecommerce	Marketing	John
ceshi-abc2	Game	Gaming	Jane
ceshi-abc3	Game	Gaming	Jane

ceshi-abc4	Entertainment	Post-production	Harry
ceshi-abc5	Entertainment	Post-production	Harry

Related Operations

Creating Tag for Cluster
Editing and Deleting Tag

Creating Tag and Binding Resource

Last updated：2023-11-20 15:50:26

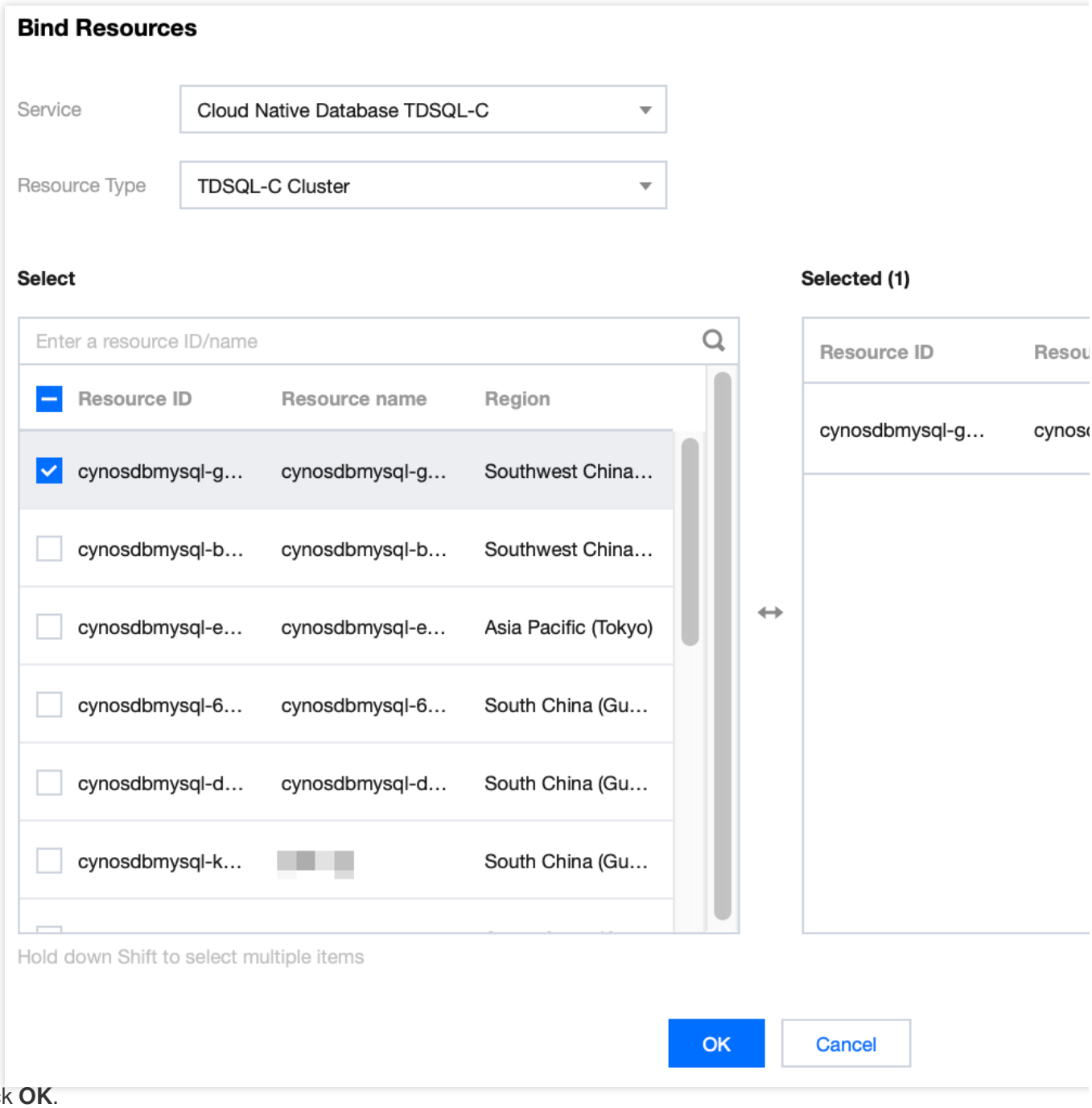
This document describes how to create tags and bind resources so that you can categorize and manage the resources of TDSQL-C for MySQL.

Creating tags

1. Log in to the Tag console.
2. Click **Tag List** on the left sidebar to enter the tag list page.
3. Click **Create Tag**. In the pop-up window, click **Add Tag Key** to create a tag or select an existing tag and add a tag value to it. You can create multiple tags at a time.
4. Click **OK**.

Binding resources

1. Find the tag key and tag value you created in the tag list and click **Bind Resources**.
2. On the **Bind Resources** page, select **Cloud Native Database TDSQL-C** for **Service**, and **TDSQL-C cluster** for **Resource Type**, and select the resources to be bound.



3. Click **OK**.

Editing and Deleting Tag

Last updated：2023-01-04 10:12:50

You can edit, modify, set, and unbind the cluster tags by the following steps.

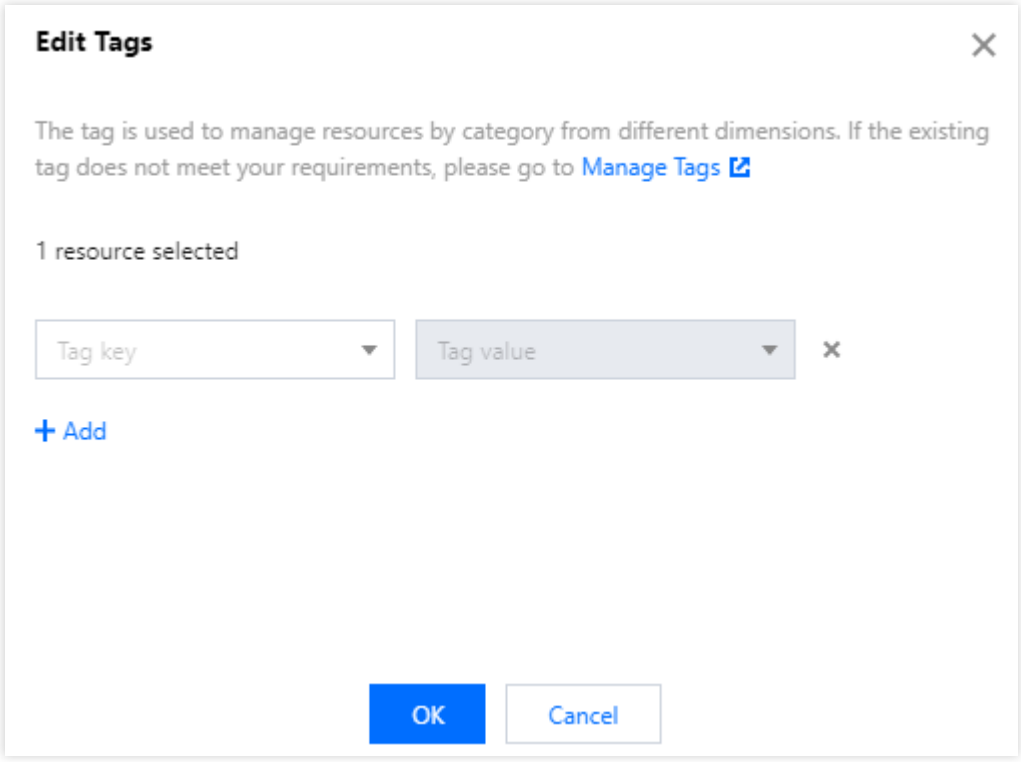
Prerequisite

You have created the tags and bound the resources. For more information, see Creating Tag and Binding Resource.

Directions

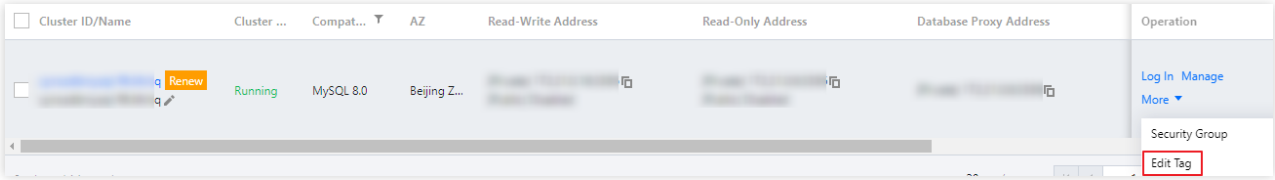
Option 1

1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top of the page and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster details page, click ![] (https://qcloudimg.tencent-cloud.cn/raw/8b38bb24eb40b11ef4d195fecebd19f5.png) after **Tag** in **Basic Info**.
4. In the pop-up window, you can edit, modify, set, and unbind the cluster tags. After confirming that everything is correct, click **OK**.



Option 2

1. Log in to the TDSQL-C for MySQL console.
2. Select a region at the top of the page, select **More** > **Edit Tag** in the **Operation** column of the target cluster in the cluster list.



3. In the pop-up window, you can edit, modify, set, and unbind the tags. After confirming that everything is correct, click **OK**.

Note:

To edit tags for multiple clusters, select the target clusters in the cluster list, and select **More** > **Edit Tag** at the top of the list.