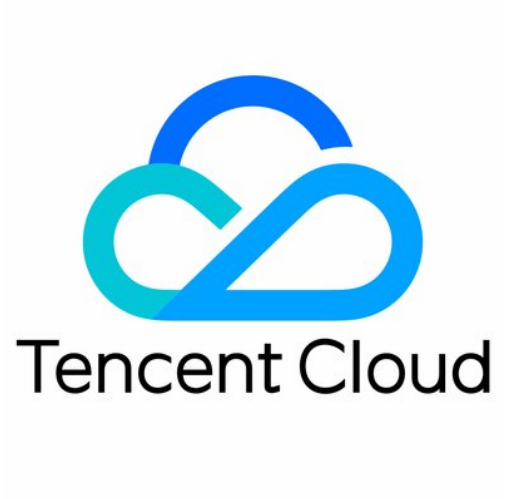
TDSQL-C for MySQL

Operation Guide

Product Documentation



Contents

Operation Guide
Instance Management
Instance Overview

Last updated：2022-04-01 23:18:38

Instance Overview

Cluster is the basic management unit of TDSQL-C for MySQL, which can contain instances of two types: read-write instance and read-only instance.
A cluster can contain only one read-write instance and multiple read-only instances.

> Note：
> Currently, a cluster can contain up to 15 read-only instances.

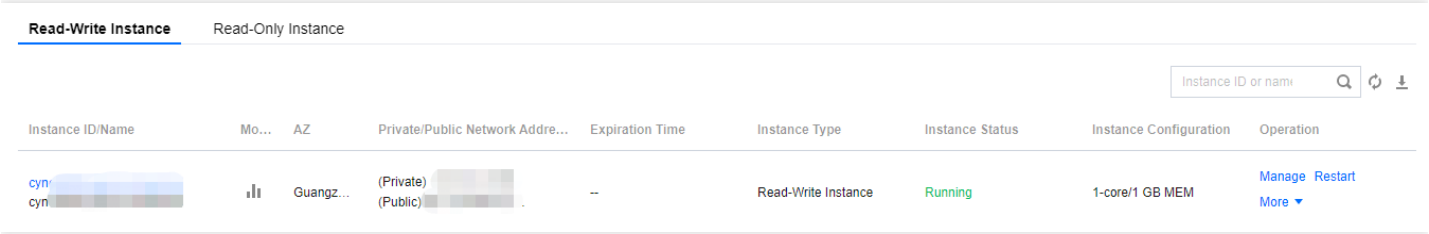
Access Address

TDSQL-C for MySQL supports read-write and read-only addresses. The read-write address is used to provide the read/write service of a TDSQL-C for MySQL primary instance (read-write instance). The read-only address corresponds to one or multiple read-only instances on the backend for load balancing.

Viewing Instance List

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 **Instance List** tab to view the list of all instances in the current cluster.

- The instance list displays the **Instance ID/Name**, **Monitoring**, **AZ**, **Private/Public Network Address**, **Expiration Time**, **Instance Type**, and **Instance Status** by default.
- You can also enter an instance ID or name keyword in the search box on the right to quickly locate the target instance.



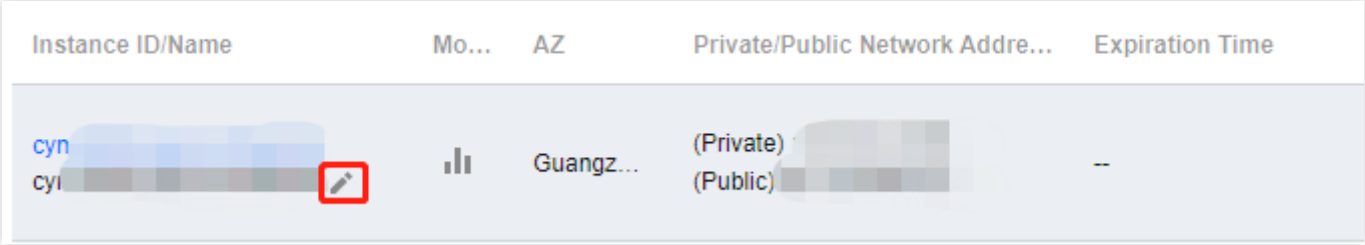
Renaming Instance

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.



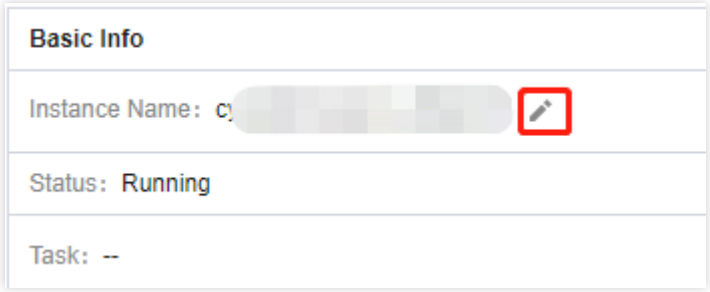
3. In the pop-up window, enter a new name and click **OK**.

> 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.

Adjusting Instance Configuration

Last updated：2022-05-31 10:41:43

An underprovisioned instance will underperform, while an overprovisioned instance will be faster but more expensive, you can adjust the instance specifications to strike a balance between cost and performance.
This document describes how to modify the computing specification and storage capacity of a TDSQL-C for MySQL instance.

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.
During instance configuration adjustment, a secondary database with high specs will be created automatically. Then, it will be automatically promoted to the primary database, and the original primary database will be disconnected to complete the configuration adjustment.

Directions

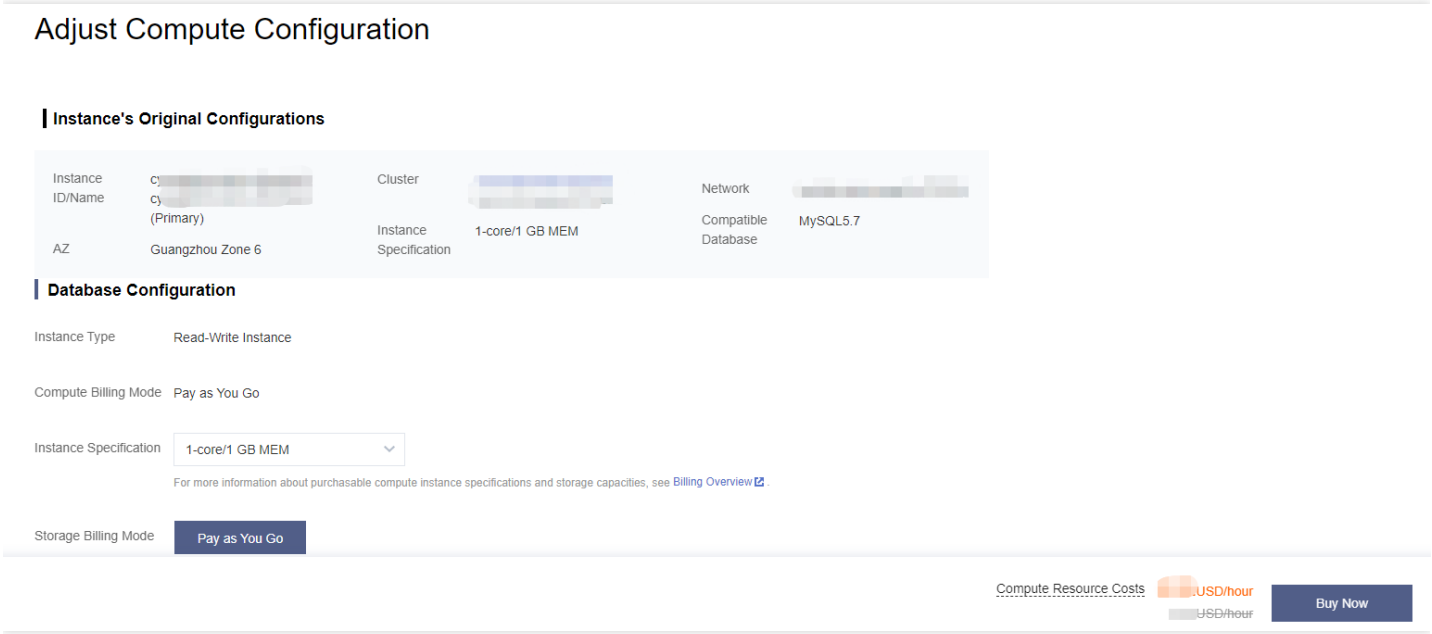
1. Log in to the console and click a cluster ID in the cluster list to enter the cluster management page.
2. On the **Instance List** tab, locate the target instance and select **More** > **Adjust Configurations** in the **Operation** column.
3. On the instance configuration adjustment page, select the target configuration and click **Buy Now**.

> Note：
> Instance configuration adjustment may cause a momentary disconnection, which may affect your business. We recommend you do so during off-peak hours.

You can also specify the time to automatically execute the configuration adjustment.

- If you select **Upon upgrade completion**, a momentary disconnection will be immediately triggered once the configuration adjustment is completed.
- If you select **During maintenance time**, a momentary disconnection and switch will be triggered within the **maintenance window** of the instance. For more information on the maintenance window, see Modifying Instance

Maintenance Window.



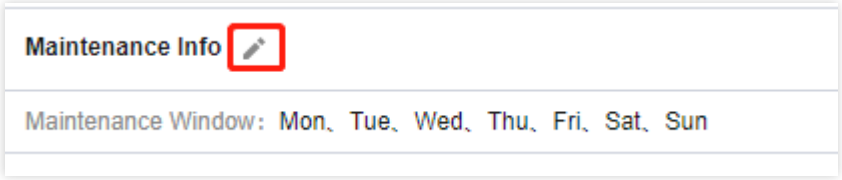
Modifying Instance Maintenance Window

Last updated：2022-04-02 00:01:27

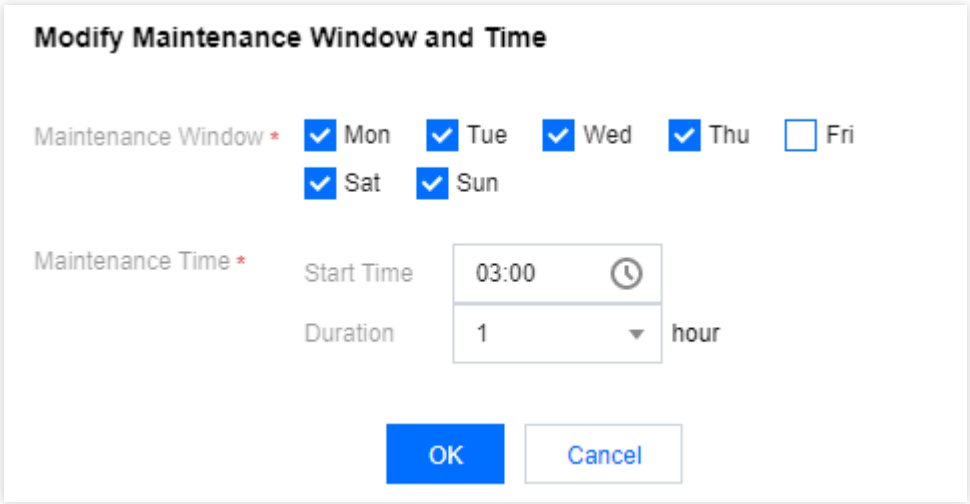
This document describes how to modify the instance maintenance time. If you want to perform configuration adjustment or other operations that will cause a momentary disconnection or affect the instance, you can specify the instance maintenance time.

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

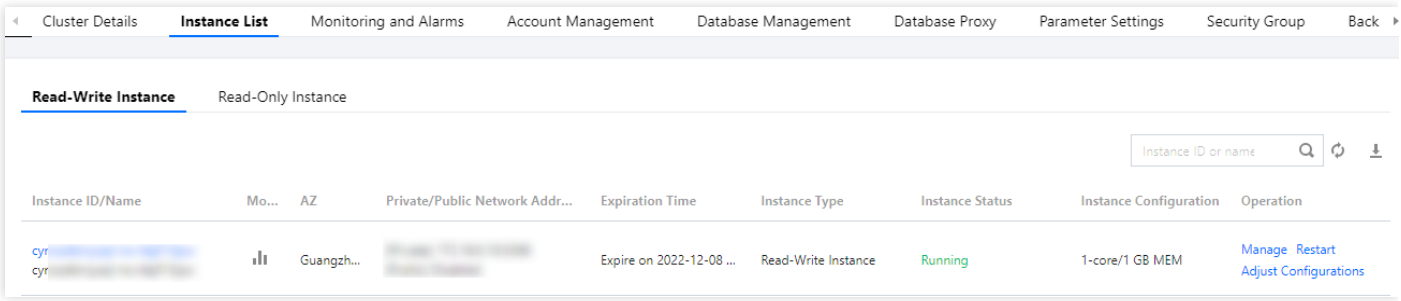
Last updated：2022-12-01 11:21:44

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.

Directions

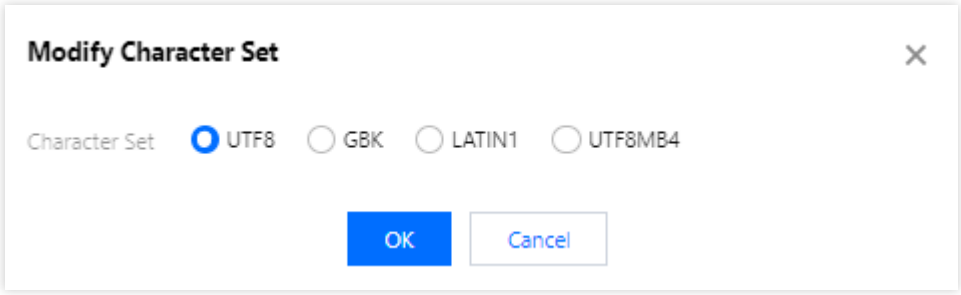
1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list to enter the cluster management page.
2. On the instance management page, select the **Instance List** tab and click the ID of the target read-write or read-only instance to enter the instance details page.



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

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

> Note：
> The supported character sets include UTF-8, GBK, Latin-1, and UTF8MB4. For more information, see MySQL's official documentation.



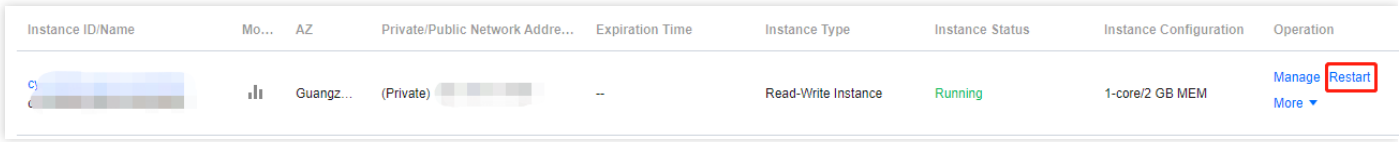
Restarting Instance

Last updated：2022-12-01 11:21:44

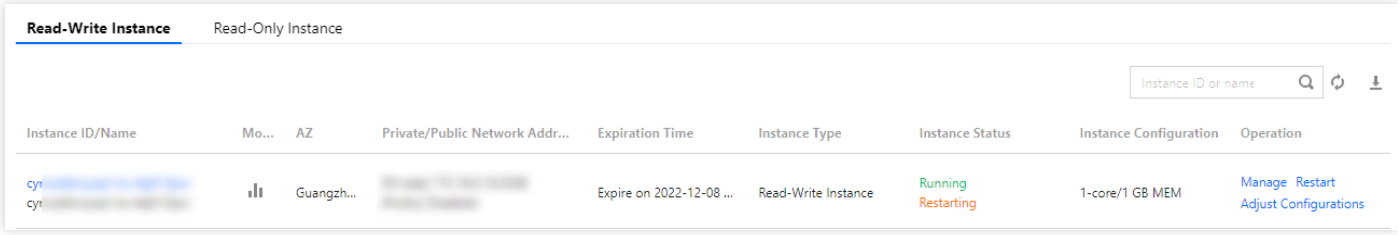
You can manually restart an instance if you want to maintain it routinely, clear the database buffer, or if a performance problem occurs.

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：2022-12-01 11:21:44

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.
- The read-write instance in a cluster can be deleted only after all read-only instances in the cluster are deleted.
- If you delete a cluster, all instances in the cluster will be deleted.
- Data is deleted only when the read-write node is deleted. Deleting read-only instances only deletes computing resources.

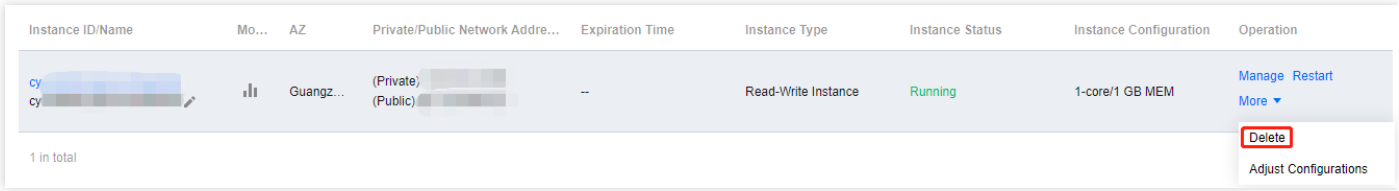
Lifecycle

- 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 instance is deleted, it will be moved to the recycle bin and retained there for 24 hours. During the retention period, the instance cannot be accessed, but it can be restored after renewal.

Directions

Started serverless or non-serverless instances

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** > **Delete** in the **Operation** column.



3. In the pop-up window, confirm that the information about the instance to be deleted is correct and click **OK**.

Paused serverless clusters

If a serverless cluster is paused, you cannot enter its management page from the console, so you cannot delete instances from it in the way described above. You can either start the cluster and then delete instances or delete the cluster directly.

Restoring Instance

Last updated：2022-12-01 11:21:44

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.

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 **Terminate** 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：2022-11-15 10:20:10

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 describes how to adjust the compute specifications of a TDSQL-C for MySQL instance.

Note：

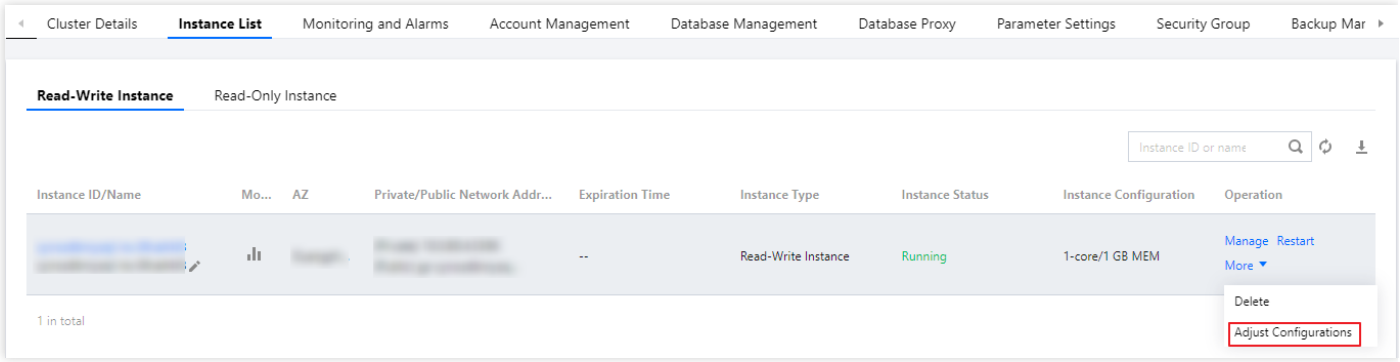
- You can upgrade monthly subscribed instance in the console. To downgrade it, submit a ticket.
- You can upgrade and downgrade the pay-as-you-go instance in the console.

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.
During instance configuration adjustment, a high-specced secondary database will be created automatically. Then, it will be automatically promoted to the primary database, and the original primary database will be disconnected to complete the configuration adjustment.

Directions

1. Log in to the TDSQL-C for MySQL console, find the target cluster and click the cluster ID or **Management** 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.

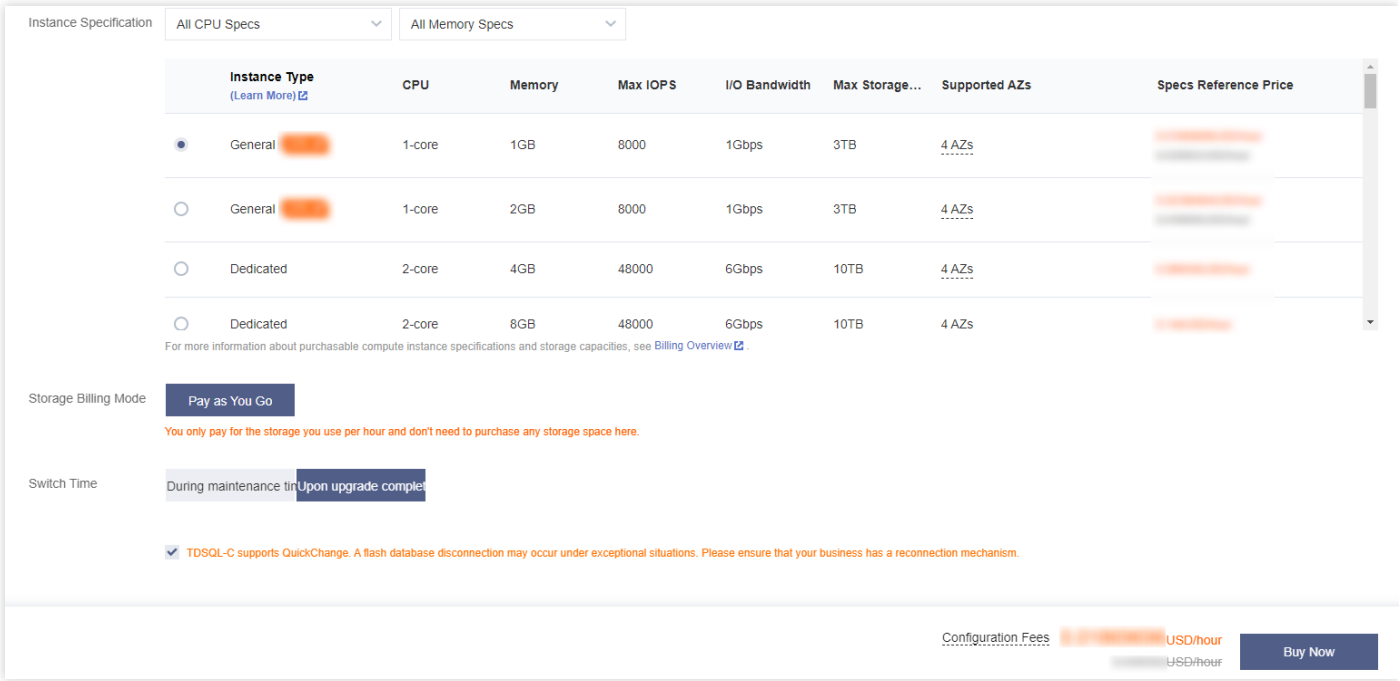


3. On the instance configuration adjustment page, select the target configuration and click **Buy Now**.

> Note：
> Instance configuration adjustment may cause a momentary disconnection, which may affect your business. We recommend that you do so during off-peak hours.

You can also specify the time to automatically execute the configuration adjustment.

- If you select **Upon upgrade completion**, a momentary disconnection will be immediately triggered once the configuration adjustment is completed.
- If you select **During maintenance time**, a momentary disconnection and switch will be triggered within the **maintenance window** of the instance. For more information, see Modifying Instance Maintenance Window.



Adjusting Storage Space

Last updated：2022-11-15 10:20:10

If the cluster storage space cannot meet your business needs, you can adjust it as needed.
This document describes how to adjust the storage space of a TDSQL-C for MySQL cluster.

Note：

- If the storage space is monthly subscribed, and you need to reduce it, submit a ticket.
- If the storage space is monthly subscribed, you can expand it in the console. To use a storage space larger than the maximum storage space of the compute specifications, upgrade the compute specifications for the read-write instance. For more information, see 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.
- 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 more information, see Product Specifications

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.
During instance configuration adjustment, a high-specced secondary database will be created automatically. Then, it will be automatically promoted to the primary database, and the original primary database will be disconnected to complete the configuration adjustment.

Directions

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

2. Select **Cluster Details** > **Configuration Info** > **Database Storage (Used/Total)** and click  .

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

Note：
If the storage space is monthly subscribed, the new storage space after adjustment must be larger than the used one. The change in storage billing mode has no impact on the existing business.

Upgrading Kernel Minor Version

Last updated：2022-04-01 22:49:31

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 a severe bug or security vulnerability occurs in TDSQL-C for MySQL, 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: when a TDSQL-C for MySQL cluster is migrated due to cluster configuration upgrade/downgrade, storage capacity expansion/reduction, or database version upgrade, etc., the system will automatically upgrade the cluster's kernel to the latest minor version.
- **Manual upgrade**:
  You can also manually upgrade kernel minor version in the console.

Upgrade Rules

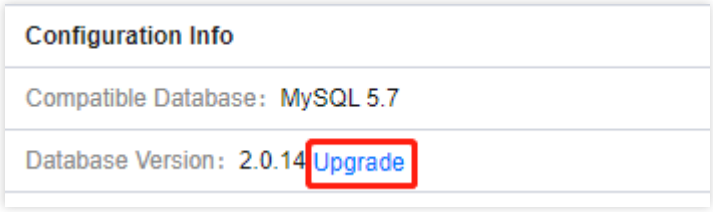
- To ensure database replication consistency, the kernel minor versions of all associated instances (read-write and read-only instances) in the cluster will also be upgraded.
- TDSQL-C for MySQL upgrade involves data migration. The time it takes for migration depends on the data size. Your business will 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.

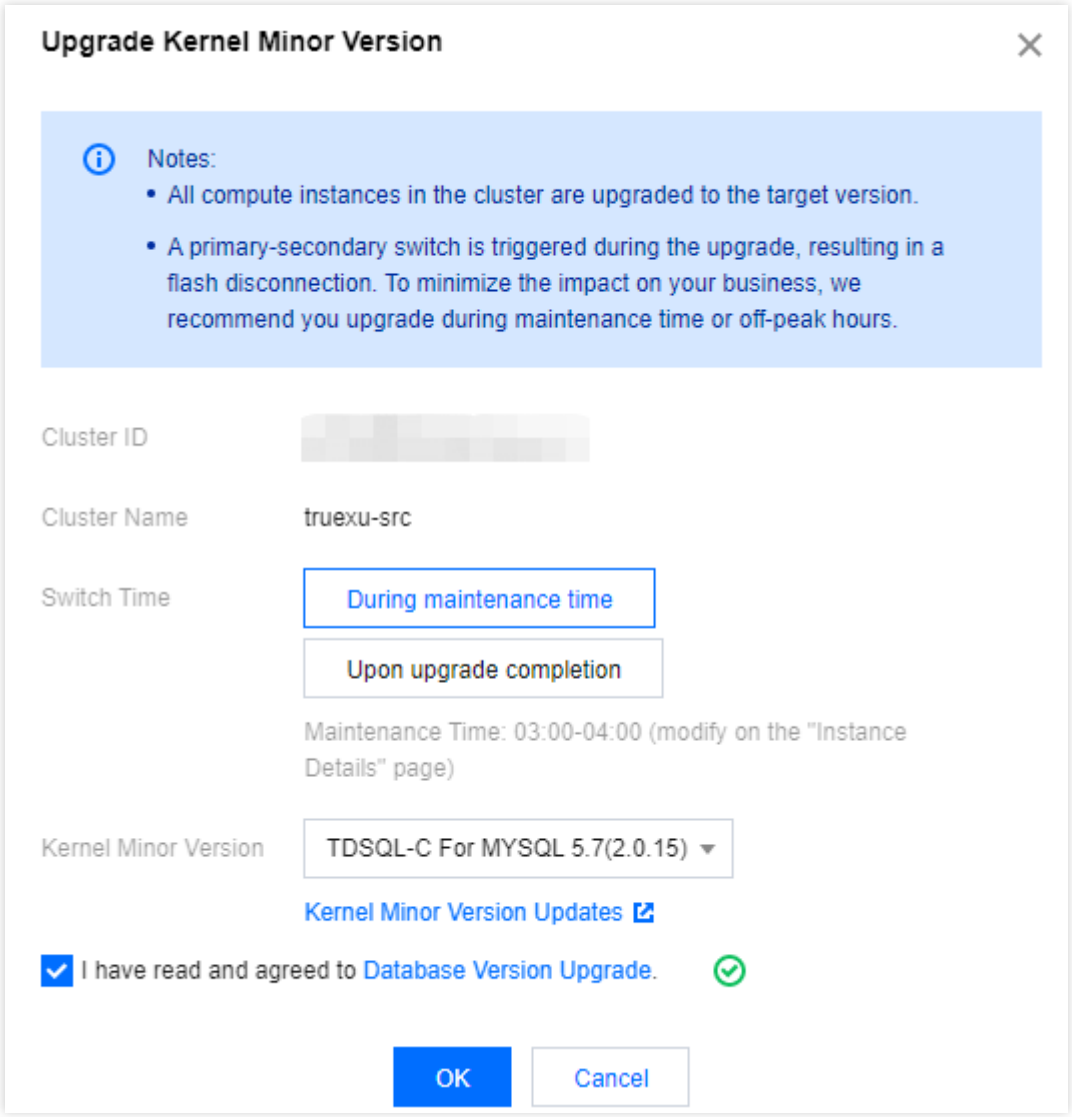
Directions

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, click **Upgrade** after **Database Version** in the **Configuration Info** section.



3. In the pop-up window, select the switch time, indicate your consent, 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.



Cluster Management

Modifying Tag

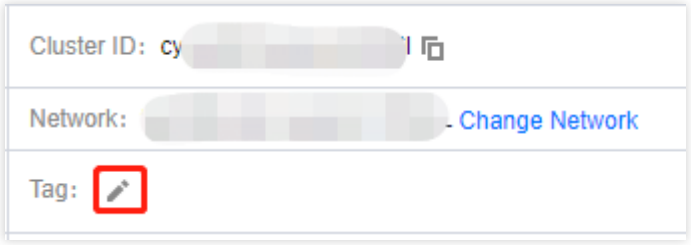
Last updated：2022-04-01 23:04:57

Tags are key-value pairs provided by Tencent Cloud to easily identify resources. For more information, see Overview.

You can use tags to categorize and manage database resources by various metrics such as business, purpose, and owner. You can also easily find a resource by its tag. In Tencent Cloud, the tag key-value pairs have no semantic meaning and are strictly parsed and matched as strings. To use tags, pay attention to the use limits first.

Directions

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  next to **Tag**.



3. In the pop-up window, edit the tag information and click **OK**.

Modifying Connection Information Enabling/Disabling Public Network Address

Last updated：2022-09-20 00:03:26

This document describes how to enable/disable public read-write and read-only addresses of a cluster in the console.

Note：

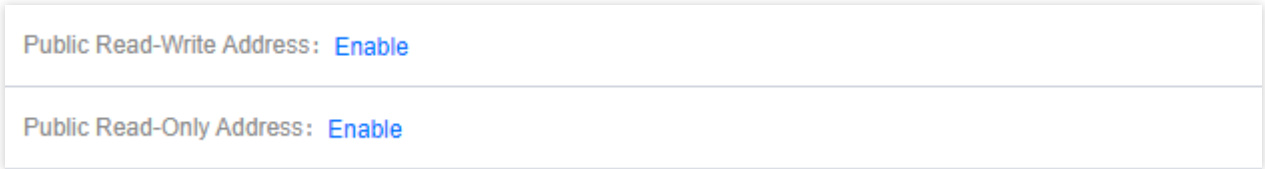
- After enabling public network access, you can connect to TDSQL-C for MySQL by using system-assigned domain name and port. It takes about five minutes for the configuration to take effect.
- TDSQL-C for MySQL public network access should be used only for development or auxiliary management. For business production, be sure to use the private network access.
- 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.
- Currently, the public network access feature is free of charge, but the stability of the public network bandwidth and traffic cannot be guaranteed.

Limit on regions

Currently, public network access can be enabled only for read-write and read-only instances in Beijing, Shanghai, Guangzhou, Chengdu, and Nanjing regions.

Enabling a cluster's public read-write/read-only addresses

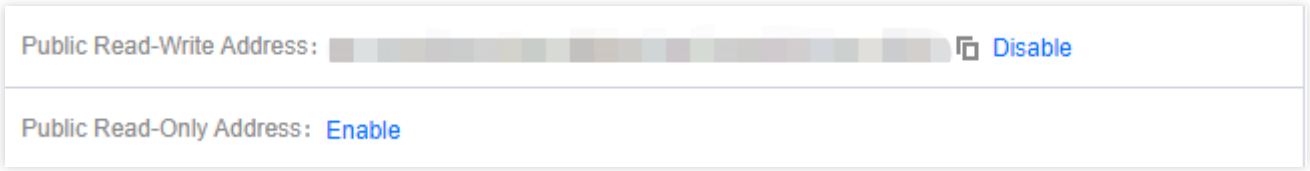
1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list to enter the cluster details page.
2. On the cluster details page, click **Enable** after the target public network address.



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

Disabling a cluster's public read-write/read-only addresses

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 **Disable** after the target public network address.



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

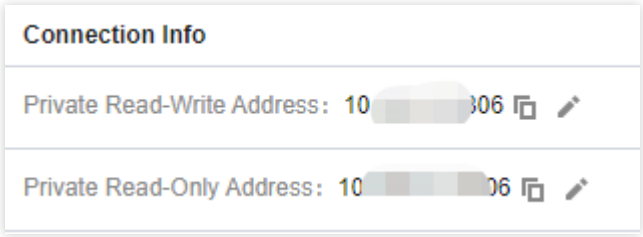
Changing Private Network Address

Last updated：2022-04-01 22:59:34

This document describes how to modify the private read-write/read-only addresses of a cluster in the console.

Operation Guide

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 target 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.

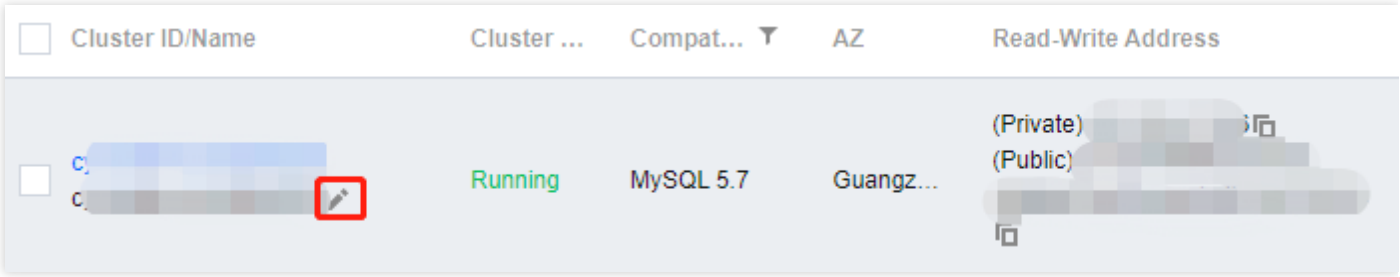
Renaming Cluster

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.



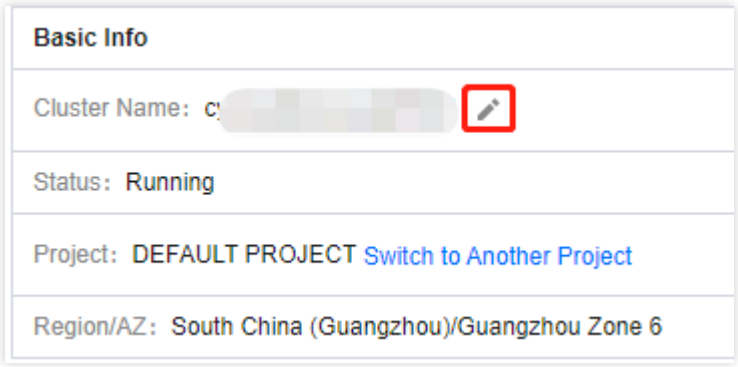
2. In the pop-up window, enter the new cluster name and click **OK**.

> 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

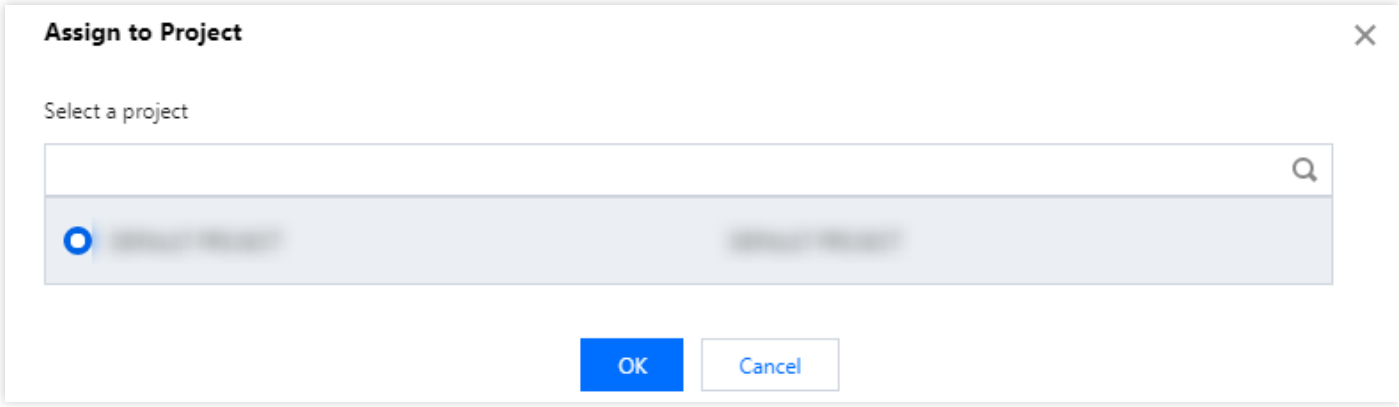
TDSQL-C for MySQL allows you to assign a cluster to different projects for management.

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：2022-12-01 11:21:44

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.

Lifecycle

- After a monthly subscribed cluster/instance is deleted, it will be moved to the recycle bin and retained there for seven days. During the retention period, the cluster/instance cannot be accessed, but it can be restored after renewal.
- After a pay-as-you-go cluster/instance is deleted, it will be moved to the recycle bin and retained there for 24 hours. During the retention period, the cluster/instance cannot be accessed, but it can be restored after renewal.

Directions

Started serverless or non-serverless clusters

1. Log in to the TDSQL-C for MySQL console.
2. Find the target cluster in the cluster list and click **More** > **Delete** in the **Operation** column.
3. In the pop-up window, read and agree to the termination rules and click **OK**.

> 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 subscribed 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 from it.

4. After a cluster is deleted, instances in it will also be moved to the recycle bin automatically. Instances in the recycle bin cannot be accessed or managed and don't incur any fees. During the isolation period, you can restore the instances from the recycle bin.

Paused serverless clusters

1. Log in to the TDSQL-C for MySQL console.
2. Find the target cluster in the cluster list and click **Delete** in the **Operation** column.
3. In the pop-up window, click **OK**.

Scaling Instance
Creating Read-Only Instance

Last updated：2022-12-01 11:21:44

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: source instance (read-write instance) and replica instance (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-write 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.

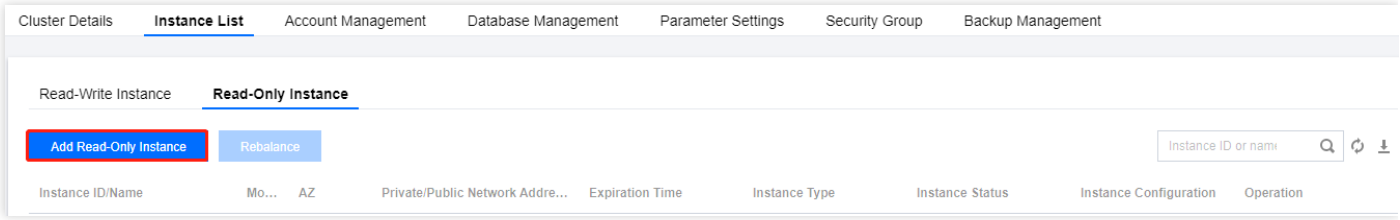
Notes

- 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 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

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.

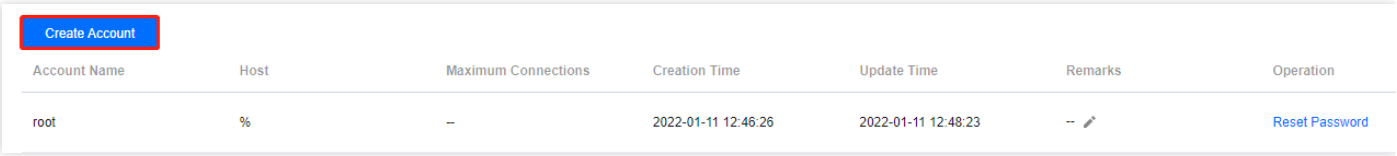
Account Management
Database Account
Creating Account

Last updated：2022-04-01 22:43:04

Besides the default root 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 and click a cluster ID in the cluster list to enter the cluster management page.
2. On the cluster management page, select the **Account Management** tab and click **Create Account**.



3. In the pop-up window, enter the account name, host, password, and confirm password and click **OK**.

- **Account Name**: the database account name can contain 1–16 letters, digits, and underscores and must begin with a letter and end with a letter or digit.
- **Host**: specify a host address to access the database. It can be an IP and contain `%` (indicating not to limit the IP range). Multiple hosts should be separated by line breaks, spaces, semicolons, commas, or vertical bars.
  - Example 1: enter `%` to indicate not to limit the IP range, that is, clients at all IP addresses are allowed to use this account to connect to the database.
  - Example 2: enter `10.5.10.%`, which means that clients whose IP range is within `10.5.10.%` are allowed to use this account to connect to the database.
- **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 `~!@#$%^&*_-+=|\(){}[]:;'<>,.?/`.

4. After successful creation, the database account can be managed in the current account list.

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 Parameters	You can modify the feature parameters to flexibly adjust the password strength settings.
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

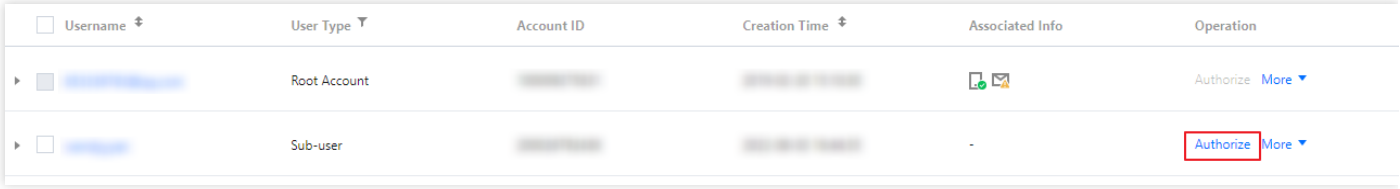
Your root account has all permissions of the custom password strength feature with no additional settings needed. By default, sub-users don't have permissions of this feature. Therefore, you need to create policies to allow the target sub-user to use the feature.

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

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

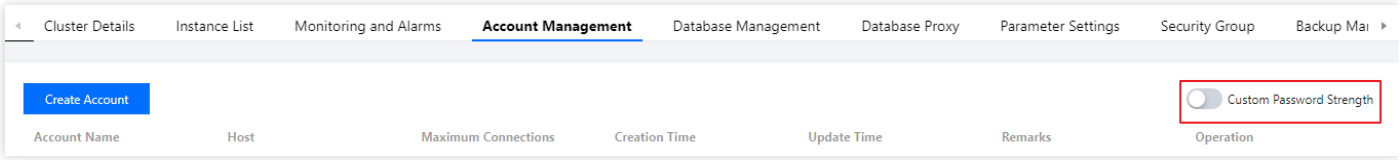
The custom password strength feature is supported by the following versions:

- 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.

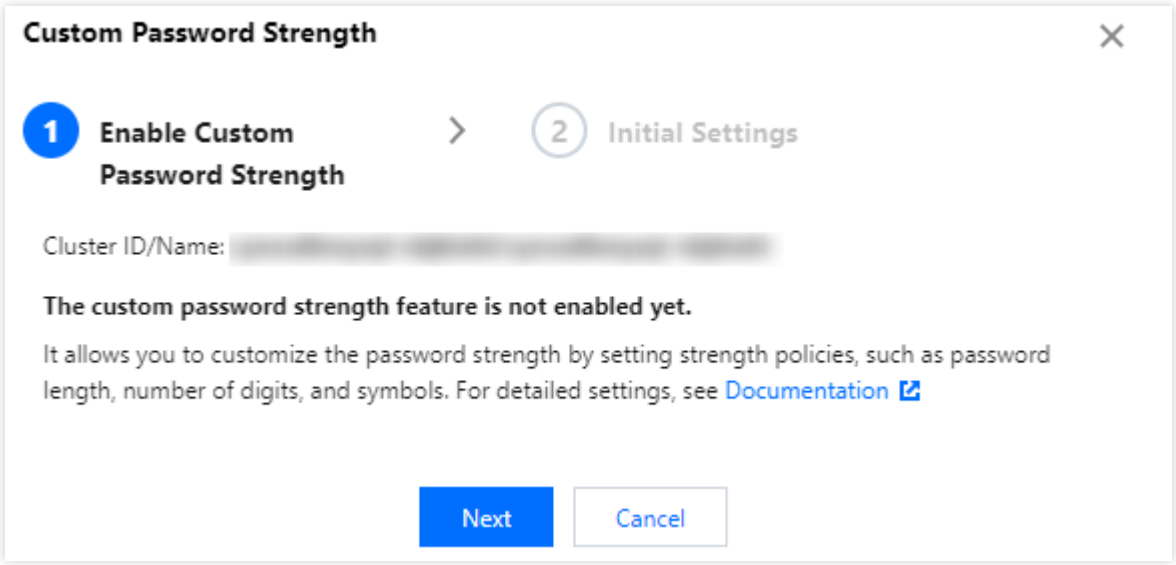
You can use this feature only after upgrading the kernel to the above versions. For detailed directions, see Upgrading Kernel Minor Version.

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.



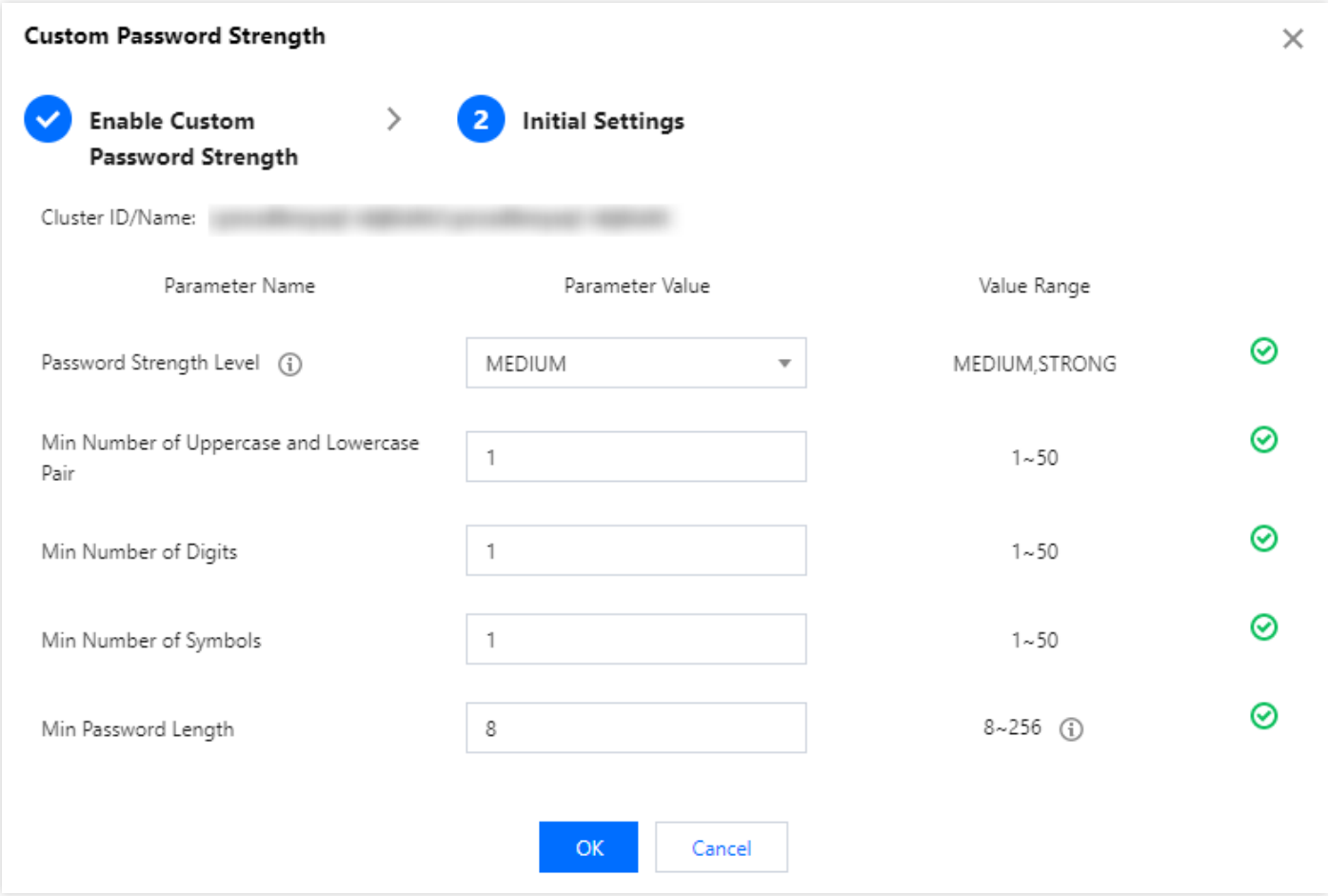
4. In the pop-up window, click **Next** to enter the initial settings page.



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.

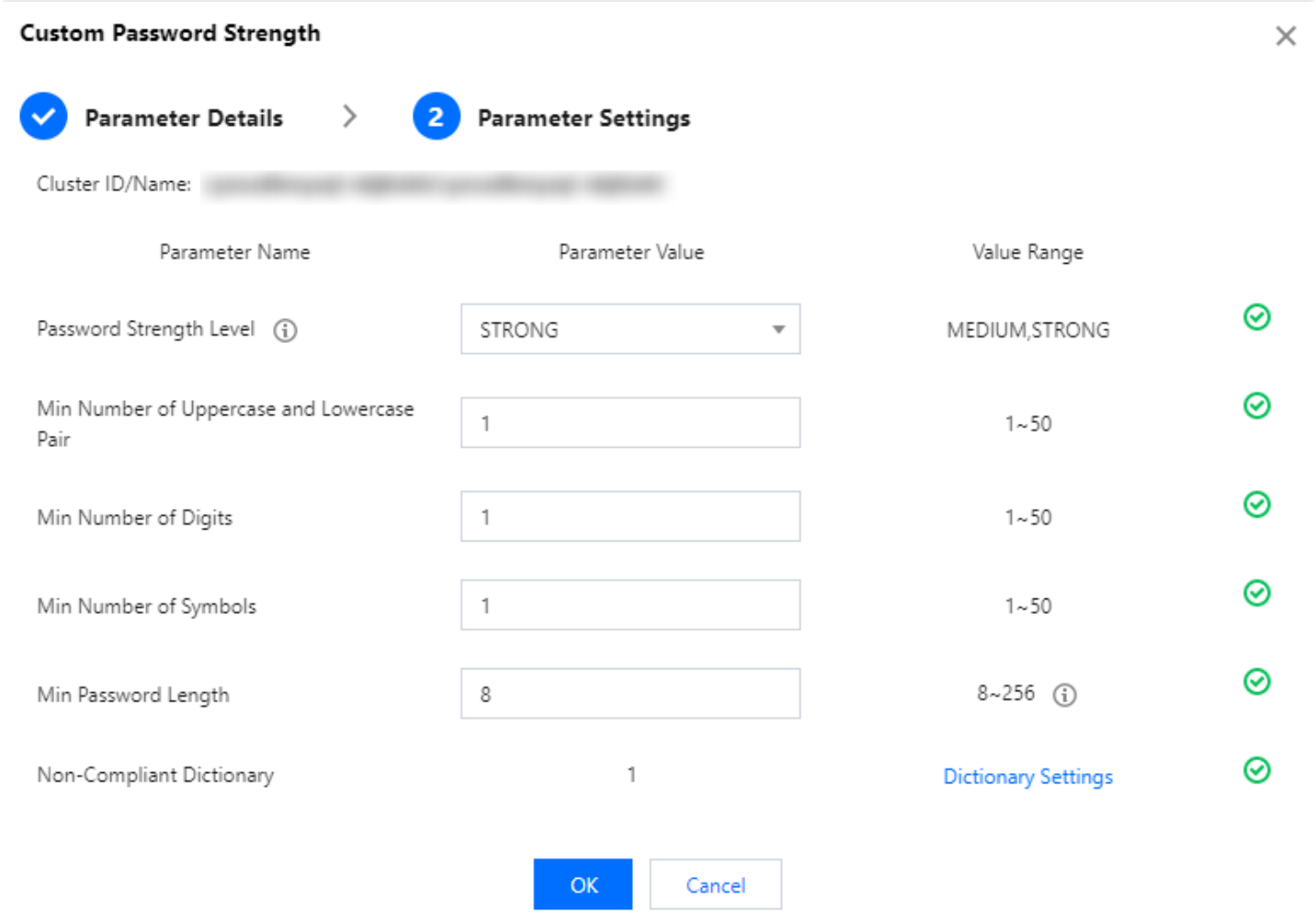
Scenario 1: The password strength level is **MEDIUM**.



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 Number of Digits	The minimum number of digits that the password must contain. Default value: `1`. Value range: 1–50.

Parameter	Description
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**.

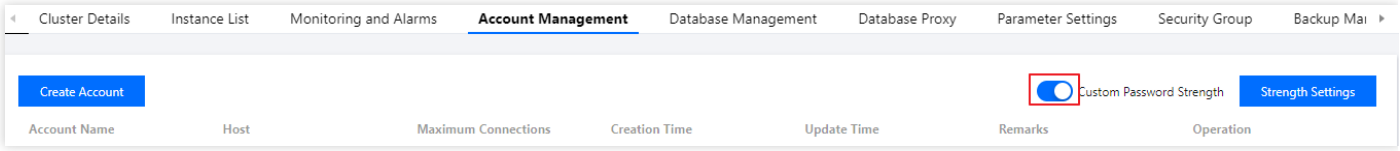


Parameter	Description

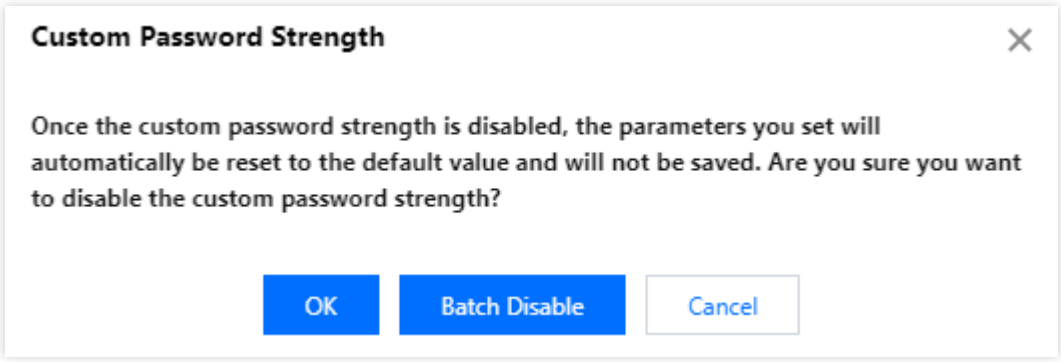
Parameter	Description
Password Strength Level	You can select **MEDIUM** or **STRONG** as the strength level. • MEDIUM: The feature under this setting will check the length, digits, letters, and symbols. • STRONG: The feature under this setting will check the length, digits, letters, symbols, and non-compliant 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：2022-09-19 15:47:05

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.

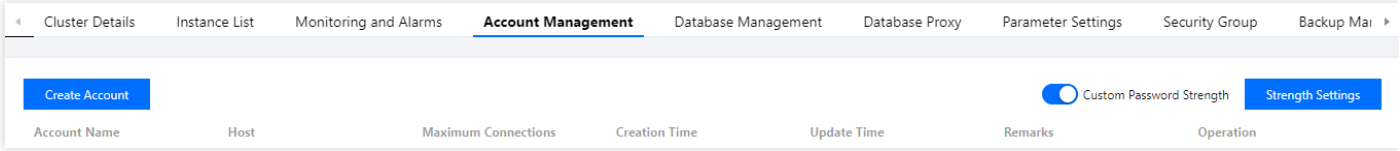
Prerequisites

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

Modifying custom password strength parameters

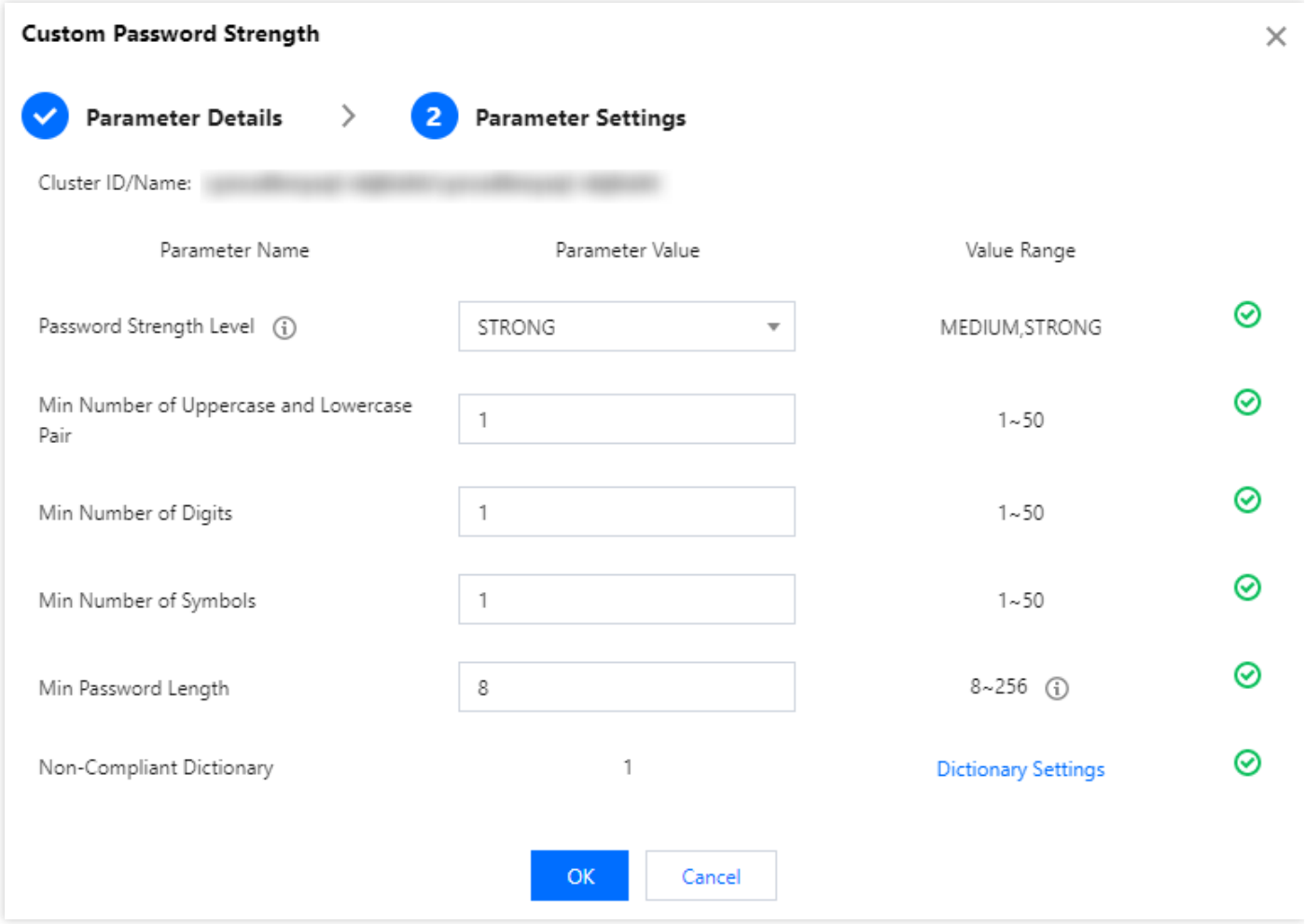
> Note：
> To modify the custom password strength parameters, click **Strength Settings**, and the console will display parameters based on the password strength level configured when you enabled the feature (**MEDIUM** or **STRONG**; the **Non-Compliant Dictionary** parameter will be displayed only under the **STRONG** level). You can modify the parameters as needed.

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. Set parameters and click **OK**.

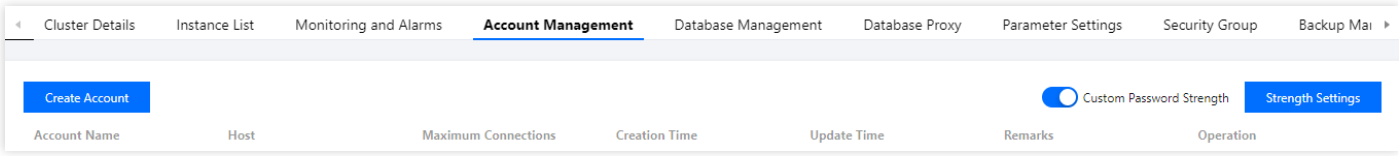


Setting parameter sync

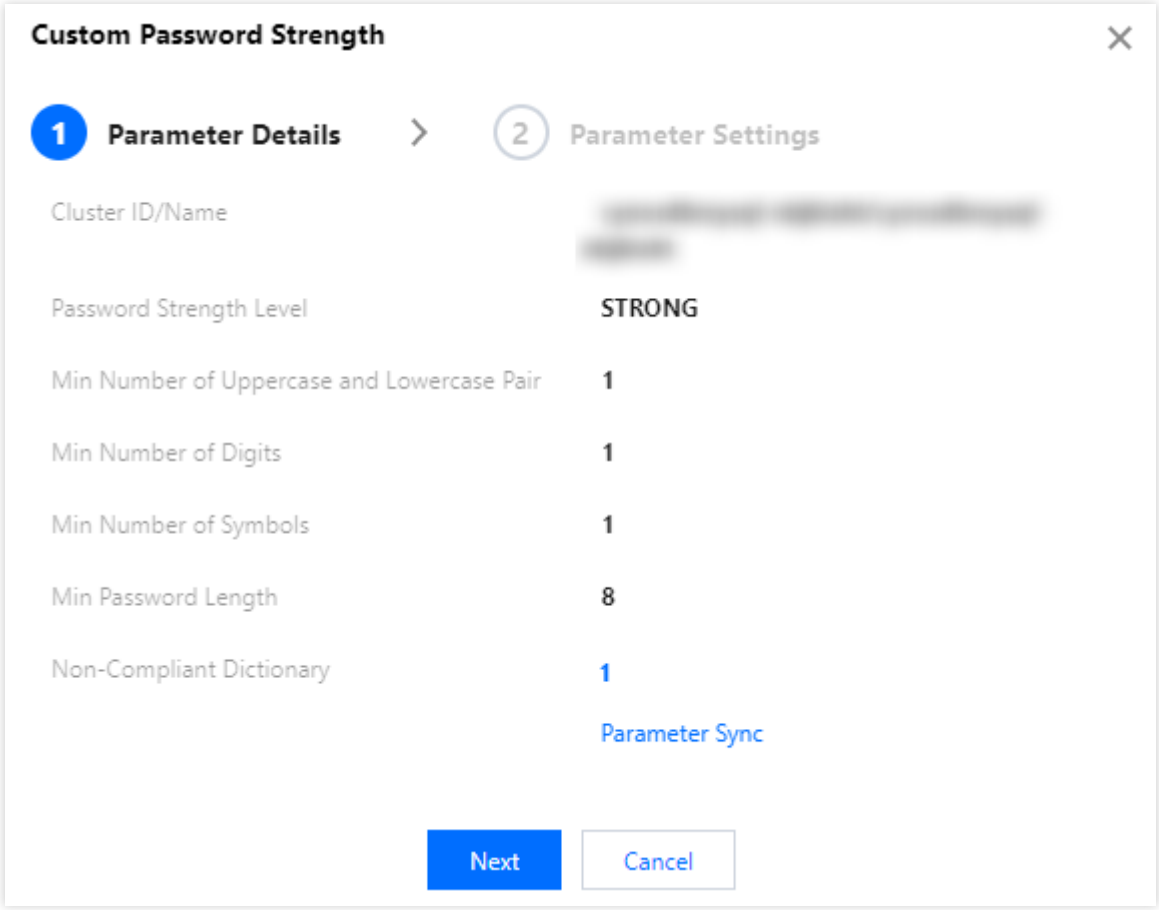
After enabling the custom password strength feature in a cluster, you can use the parameter sync feature to sync the configured custom password strength parameters, including **Non-Compliant Dictionary**, to other clusters in the same region under the current Tencent Cloud account. If this feature is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

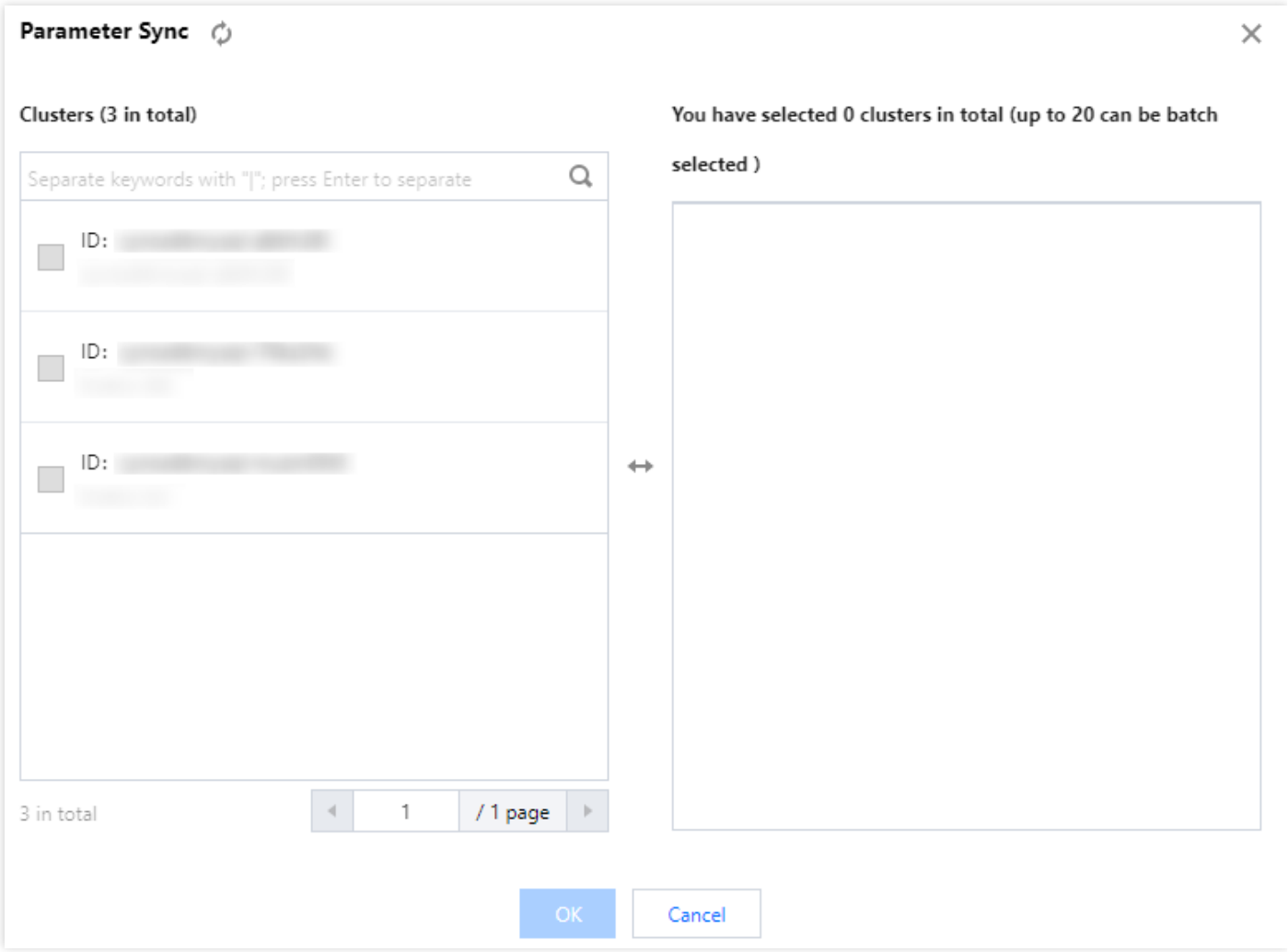
3. On the cluster management page, select **Account Management** and click **Strength Settings** on the right.



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**.



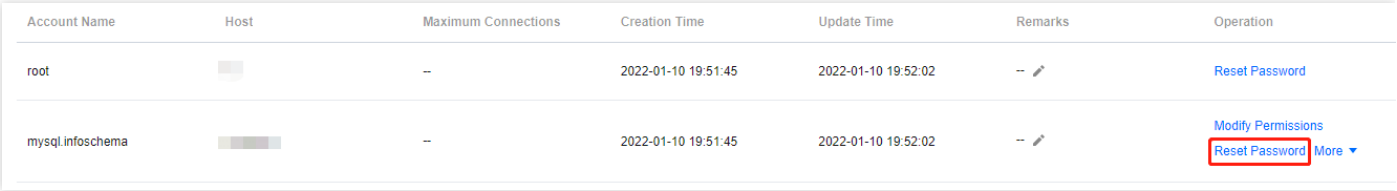
Resetting Password

Last updated：2022-04-01 22:39:02

This document describes how to change the password of the DBA account in the console.

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 **Account Management** tab and click **Reset Password** in the **Operation** column.



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

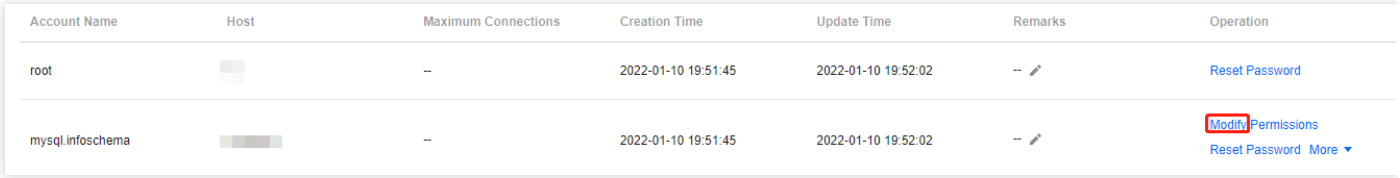
Modifying Account Permission

Last updated：2022-04-01 22:38:09

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

Directions

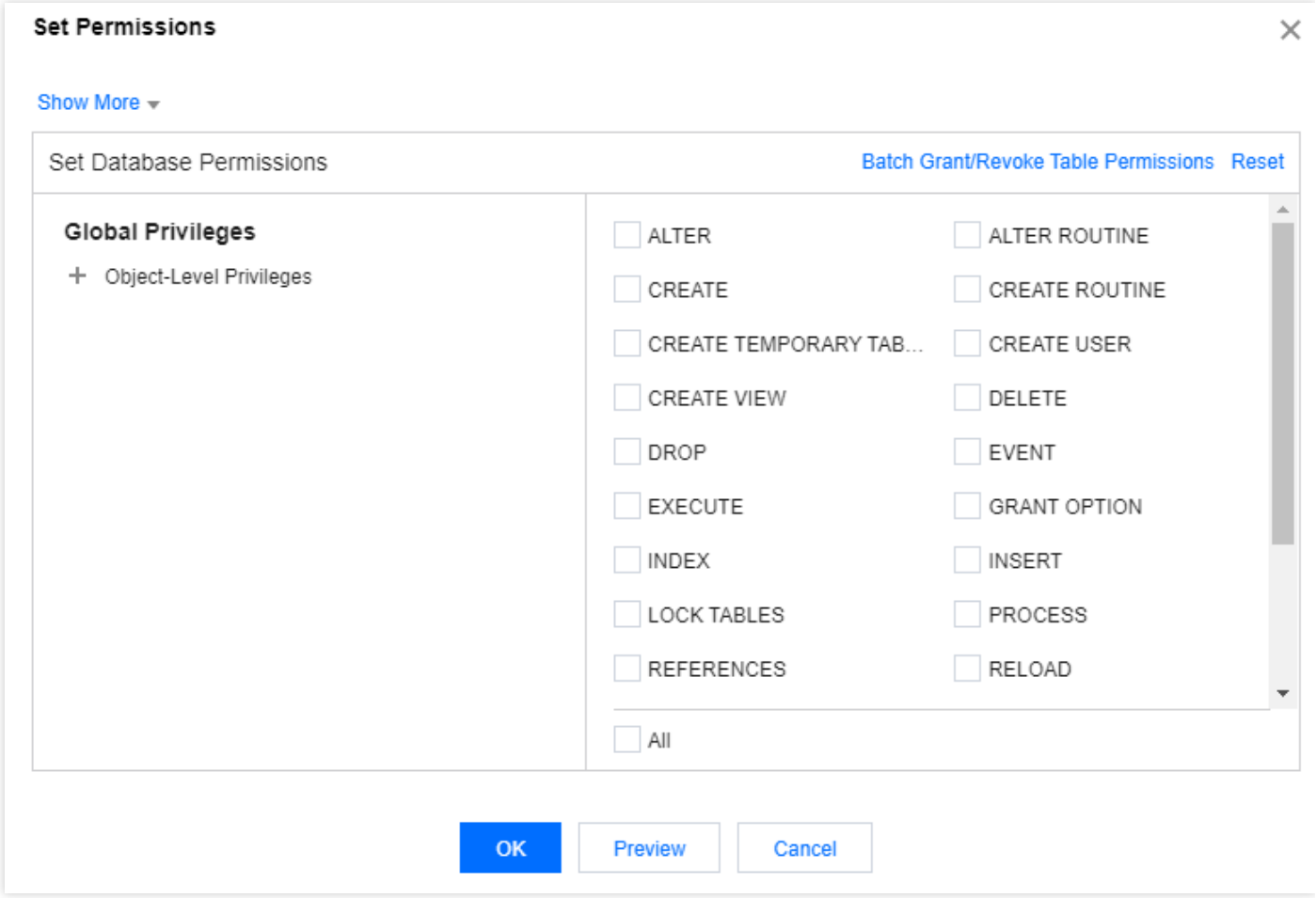
1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list to enter the cluster management page.
2. On the cluster management page, select the **Account Management** tab and click **Modify Permissions** in the **Operation** column.



3. In the pop-up window, select or deselect permissions and click **OK**.

- **Global Privileges**: grant global permissions of all databases in the instance.

- **Object-Level Privileges**: grant permissions of certain databases in the instance.



Modifying Authorized Host Address

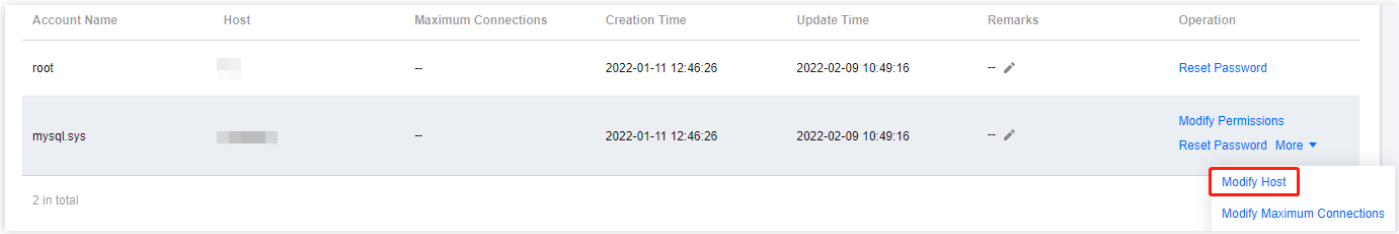
Last updated：2022-04-01 22:37:12

You can modify the host addresses authorized by a database account in the console to control the access to the database, thus enhancing database access security.

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 **Account Management** tab and select **More** > **Modify Host** in the **Operation** column.

> Note：
> You cannot modify the host address of the `root` account.



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

> Note：
> The host address can be an IP or `%` (indicating not to limit the IP range).
>
> - Example 1: enter `%` to indicate not to limit 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.%`, which means that clients whose IP range is within `10.5.10.%` are allowed to use this account to access the database.

Deleting Account

Last updated：2022-04-01 22:36:04

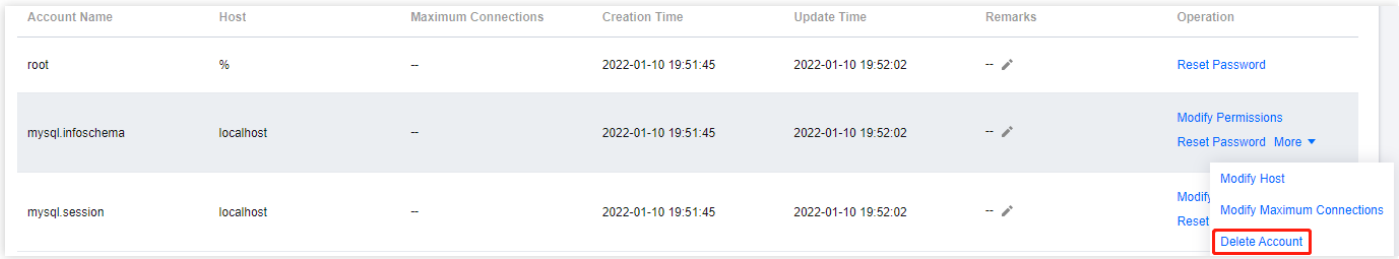
To disable a created database account, you can delete it in the console.

Note：

- A database account cannot be recovered once deleted.
- In order to avoid accidental deletion from interrupting normal use by your business, you need to make sure that the database account to be deleted is no longer used by any applications.

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 **Account Management** tab and select **More** > **Delete Account** in the **Operation** column.



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

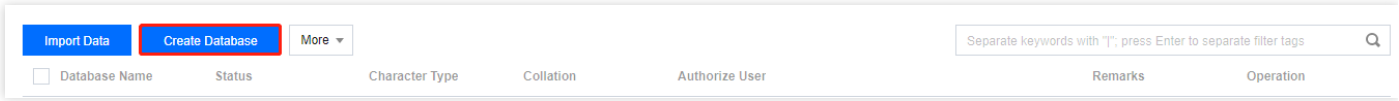
Database Management
Creating Database

Last updated：2022-04-01 22:35:24

This document describes how to create a database in the console.

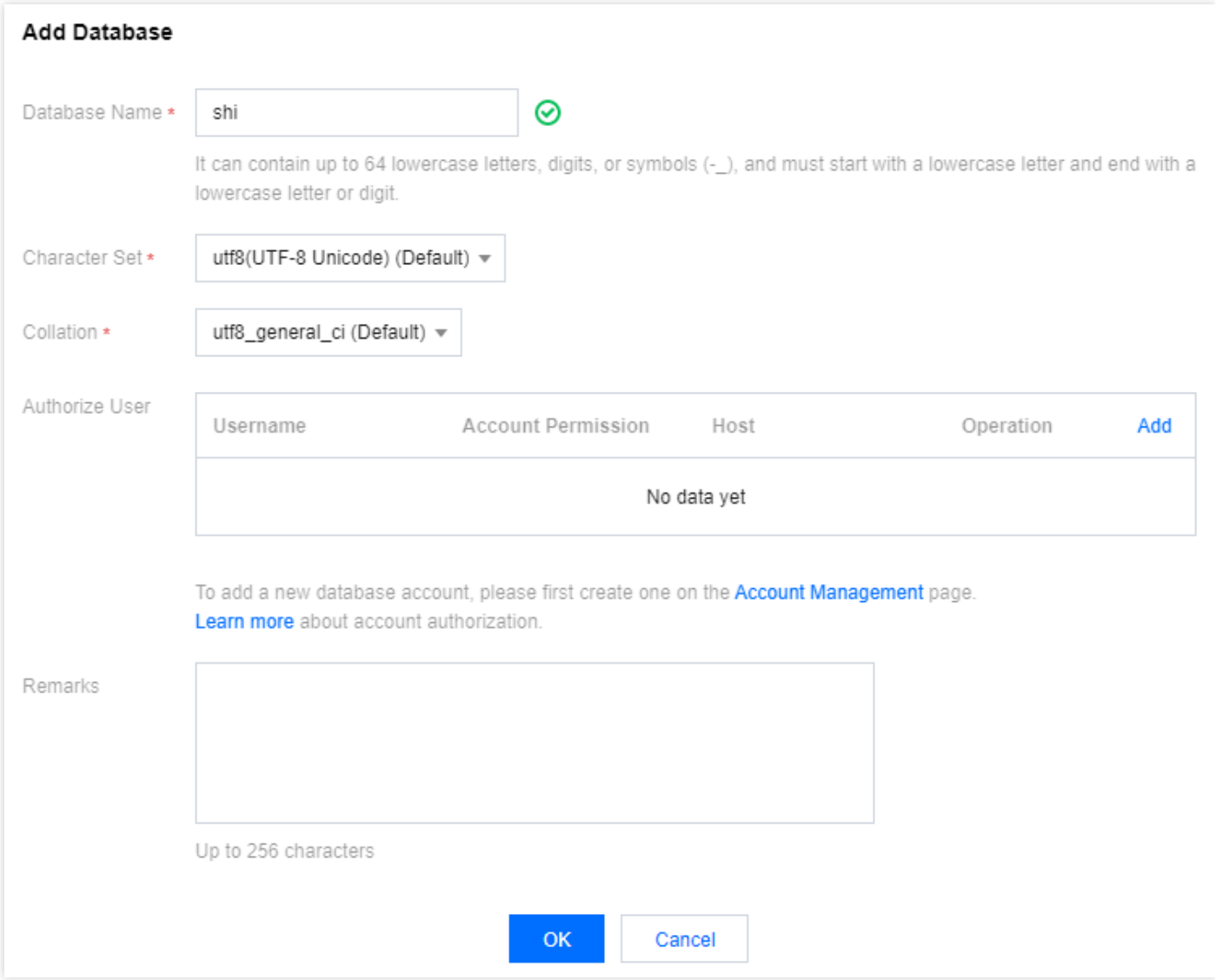
Directions

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 **Database Management** page and click **Create Database**.



3. In the pop-up window, enter a database name, select a character set, and click **OK**.

- **Database Name**: it can contain up to 64 lowercase letters, digits, hyphens, and underscores and must start with a letter and end with a letter or digit.
- **Character Set/Collation**: for more information on the character set and collation, see MySQL's official documentation.
- **Authorize User**: enter the username of the user to be authorized, account permission, and host information. You can also delete the record.



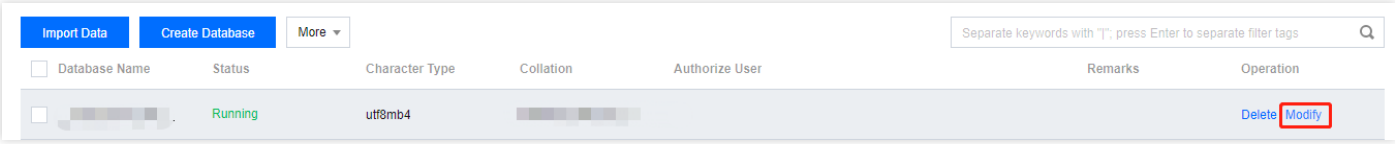
Modifying Database User Permission

Last updated：2022-04-01 22:33:49

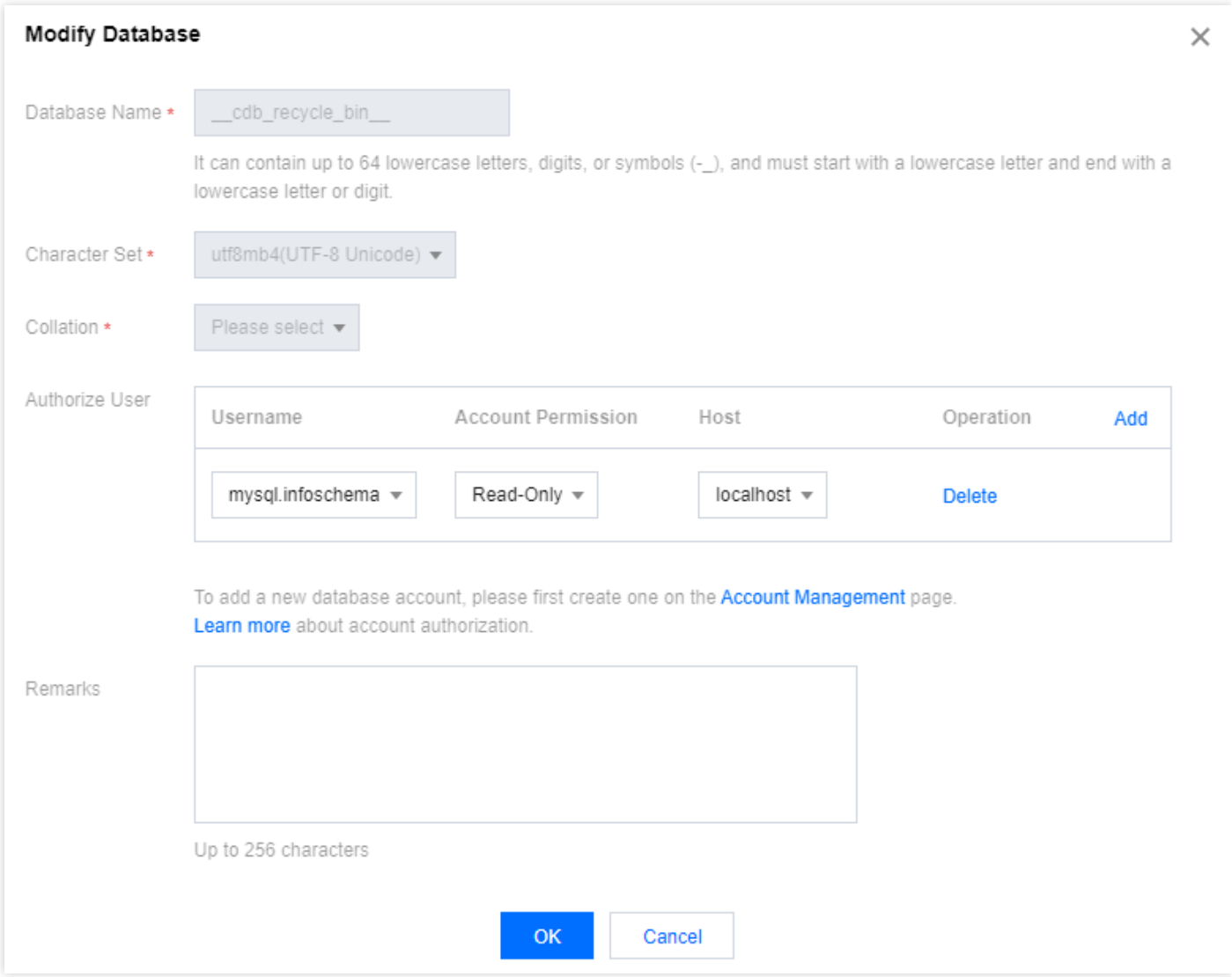
This document describes how to modify the database user permission in the console.

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 **Database Management** tab and click **Modify** in the **Operation** column.



3. In the pop-up window, modify the host authorized to the user and click **OK**.



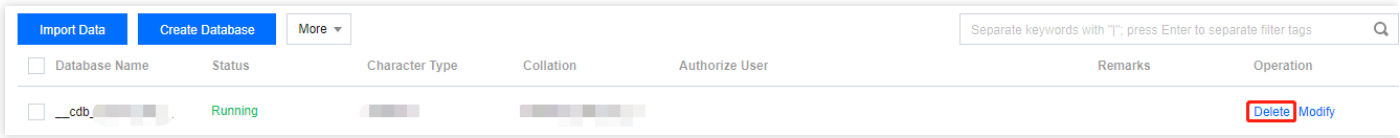
Deleting Database

Last updated：2022-04-01 22:31:07

This document describes how to delete a created database in the console.

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 **Database Management** tab and click **Delete** in the **Operation** column.



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

Parameter Configuration
Setting Instance Parameters

Last updated：2022-12-01 11:21:44

You can view and modify certain parameters and query parameter modification logs in the console.

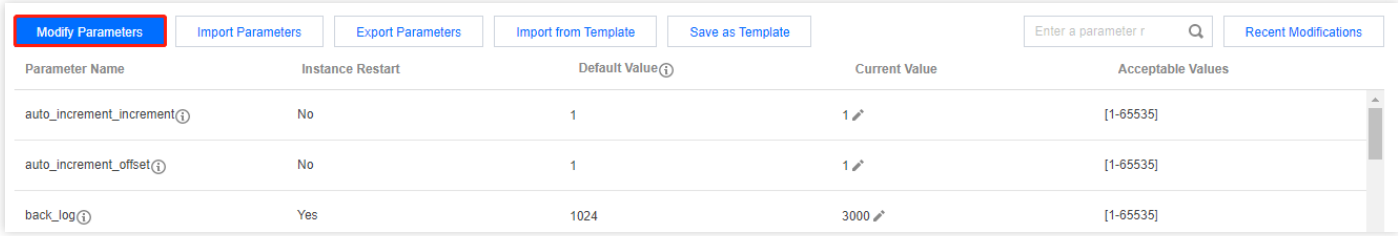
Notes

- To ensure instance stability, only some parameters can be modified in the console. These parameters are displayed on the **Parameter Settings** page.
- If the modified parameter requires instance restart to take effect, the system will ask you if you wish to restart. We recommend that you do so during off-peak hours and ensure that your application has a reconnection mechanism.
- Currently, you cannot enter a custom formula to set a parameter.
- If you want to return to the default formula, clear the entered parameters and apply.
- Parameter modification task:
  - You can query 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.

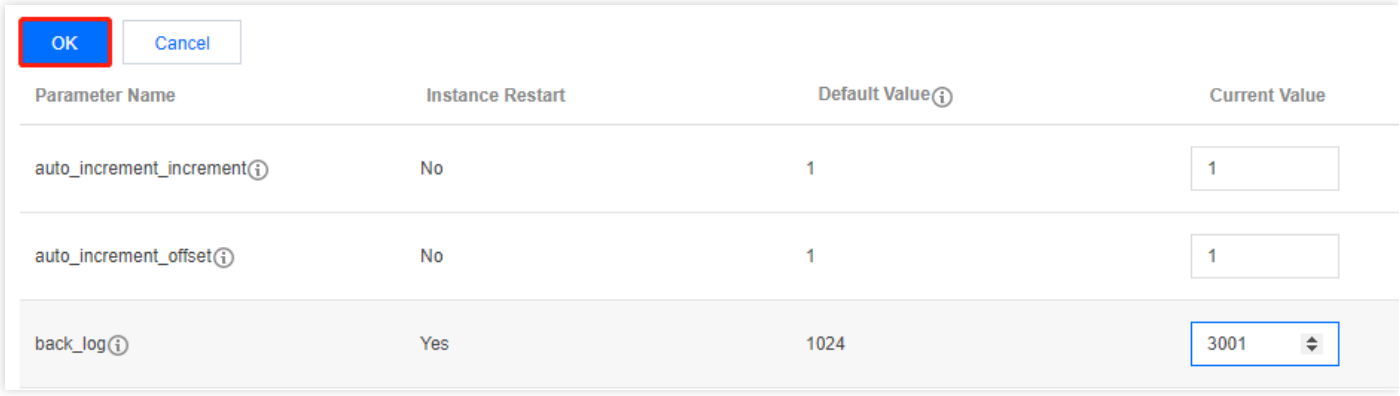
Modifying Parameters in the Parameter List

Modifying parameters in batches

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 cluster management page, select the **Parameter Settings** tab and click **Modify Parameters**.



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



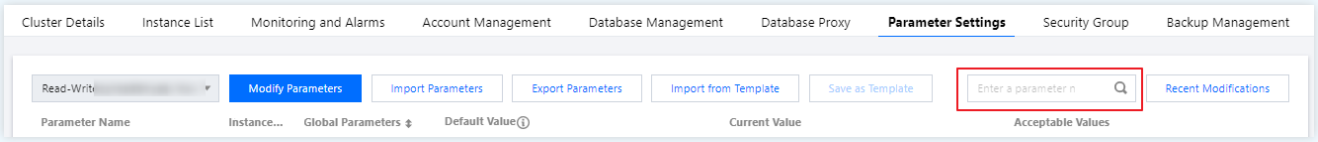
4. In the pop-up window, confirm that everything is correct and click **OK**.

Modifying one parameter

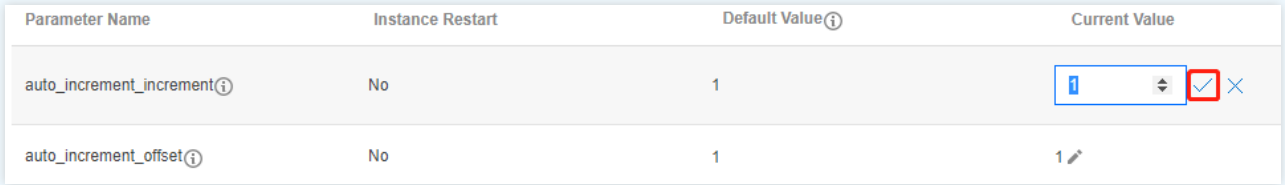
1. Log in to the TDSQL-C for MySQL console and click the ID of the target cluster in the cluster list to enter the cluster management page.
2. On the **Parameter Settings** tab, locate the target parameter in the parameter list and click in the **Current Value** column.

Note：

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



iii. Modify the value within the restrictions stated in the **Acceptable Values** column and click to save the modification. You can click to cancel the operation.

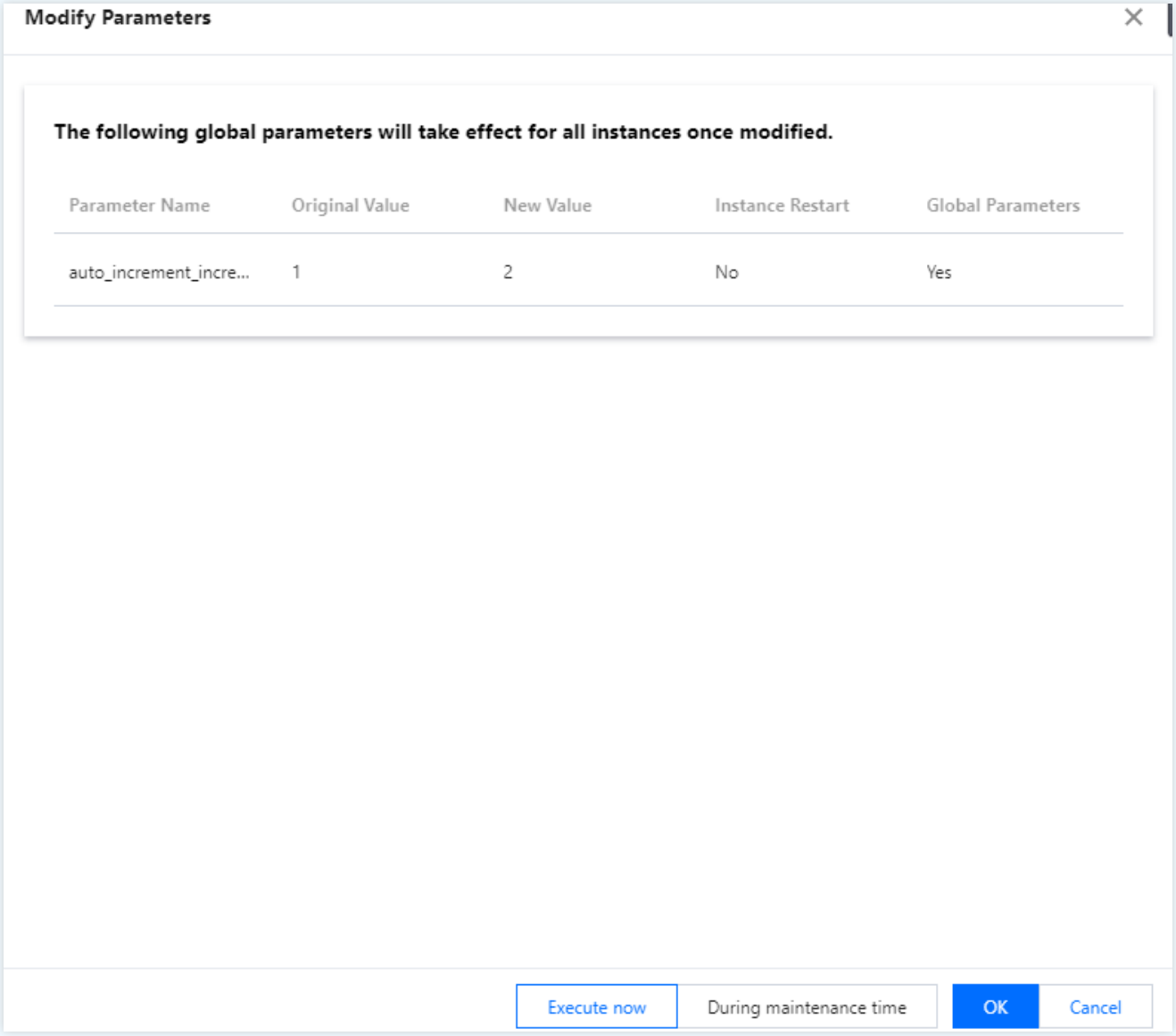


3. In the pop-up window, confirm that everything is correct and click **OK**.

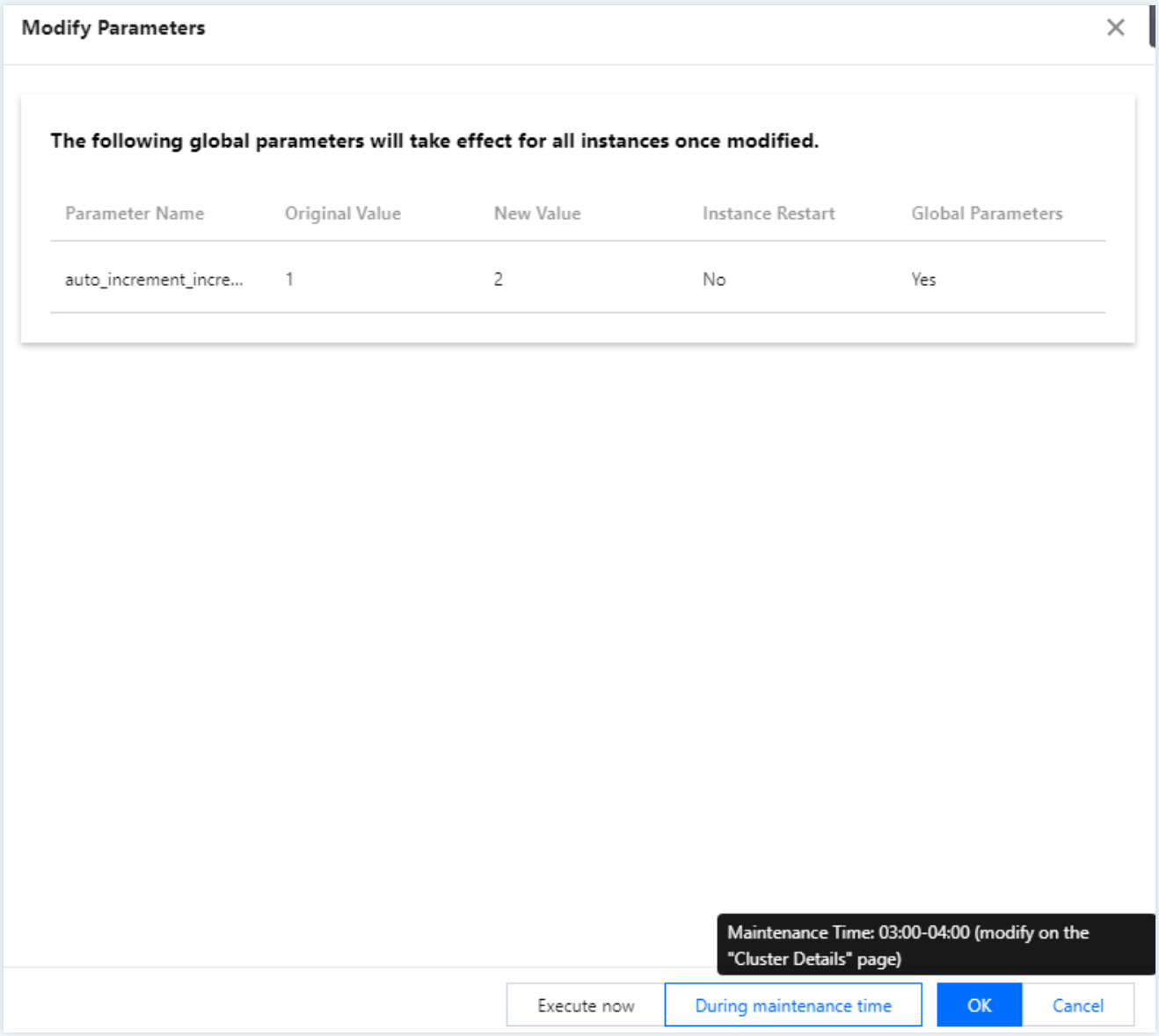
Note：

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 prompt when you modify a parameter that doesn't involve restart is as follows:



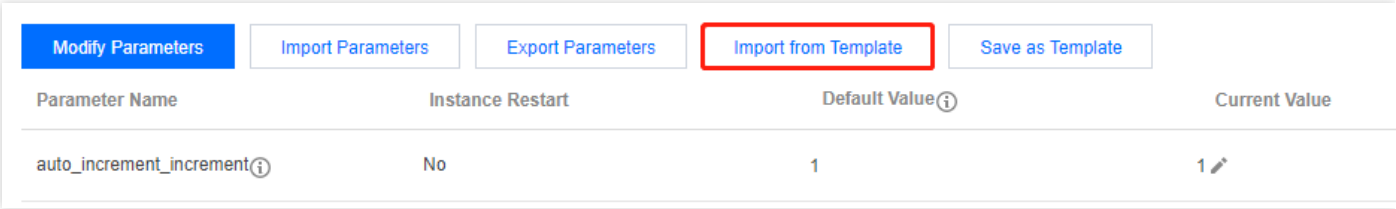
The prompt when you modify a parameter that involves restart is as follows:



Modifying Parameters by Importing a Parameter Template

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.

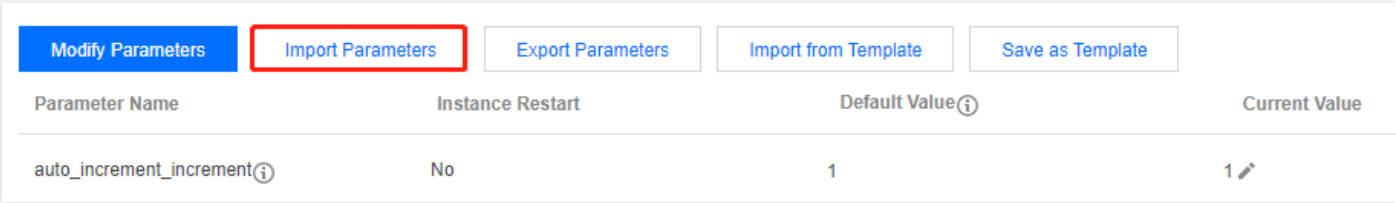
2. On the cluster management page, select the **Parameter Settings** tab and click **Import from Template**.



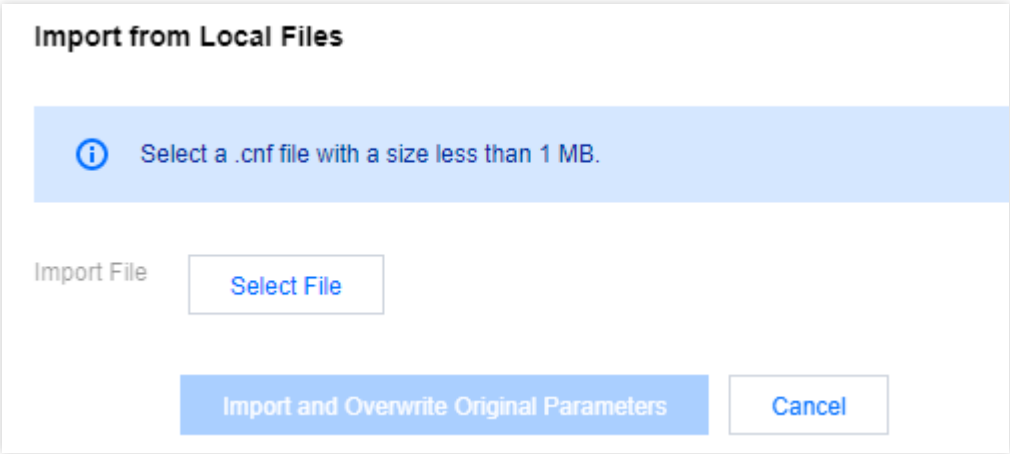
3. In the pop-up window, select a parameter template and click **OK**.
4. After confirming that everything is correct, click **OK** in the top-left corner.
5. In the pop-up window, confirm that everything is correct and click **OK**.

Modifying Parameters by Importing a Parameter Configuration File

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 cluster management page, select the **Parameter Settings** tab and click **Import Parameters**.



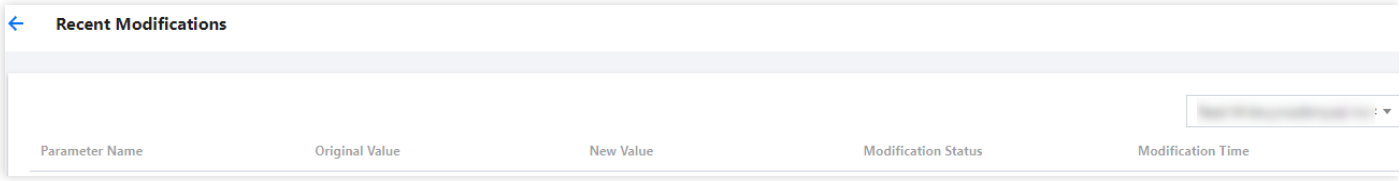
3. In the pop-up window, select the file to be uploaded and click **Import and Overwrite Original Parameters**.



4. After confirming that everything is correct, click **OK** in the top-left corner.
5. In the pop-up window, confirm that everything is correct and click **OK**.

Querying Parameter Modification Logs

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 cluster management page, select the **Parameter Settings** tab and click **Recent Modifications** on the right.



Exporting Parameter Configuration as a File

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 cluster management page, select the **Parameter Settings** tab and click **Export Parameters** to export the parameter configuration file.

Exporting Parameter Configuration as a Template

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 cluster management page, select the **Parameter Settings** tab and click **Save as Template** to save the existing parameter configuration as a parameter template.

Subsequent Operations

- You can use templates to manage database parameters in batches. For more information, see Applying Parameter Template.
- For suggestions on configuring key parameters, see Suggestions on Parameter Configuration.

Applying Parameter Template

Last updated：2022-04-01 22:26:47

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 use 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 TDSQL-C for MySQL console and click **Parameter Template** on the left sidebar to view parameters. The following parameter template features are supported:

- Use system default parameter templates, including high-performance and high-stability parameter templates.
- Create custom templates by modifying a default parameter template.
- Generate templates by importing parameters from configuration file `my.conf` .
- Save parameter configurations as templates.
- Import parameters from templates to apply to one or more instances.
- Compare two parameter templates.

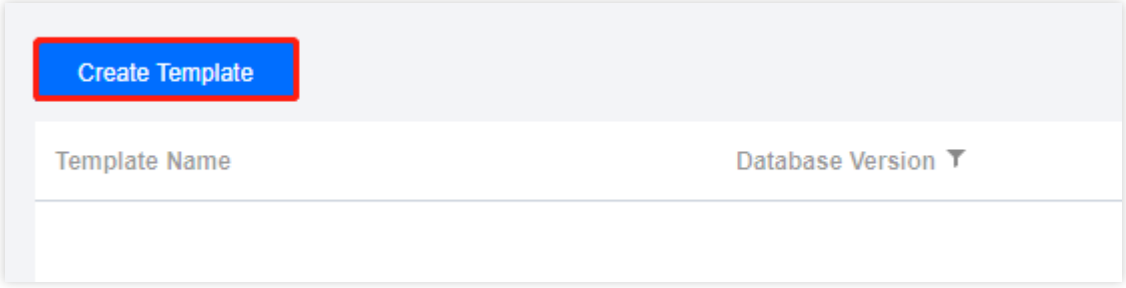
Note：

- 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 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 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**:

- **Template Name**: enter a unique template name.
- **Template Description**: enter a brief description of the parameter template.
- **Database Version**: select a database version.

3. After the creation is completed, you can modify, import, and export parameters on the template details page.

Copying 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.

Method 1. Copying an existing parameter template

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 **Save as Template** at the top of the template details page.
3. In the pop-up window, specify the following configurations:

- **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 **Save**.

Method 2. Saving parameters of an instance as a parameter template

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 **Parameter Settings** tab and click **Save as Template**.
3. In the pop-up window, specify the following configurations:

- **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 **Save**.

Modifying Parameter Values in Parameter Template

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 **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 Parameter Template

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. 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 Parameter Template

Method 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, locate the target template and click **Export** in the **Operation** column.

Method 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 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, locate the target template and click **Delete** in the **Operation** column.
3. In the pop-up window, click **OK**.

Subsequent Operations

For suggestions for configuration of key parameters, see Suggestions on Parameter Settings.

Suggestions on Parameter Configuration

Last updated：2022-02-11 16:58:45

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
- Whether restart is 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 contains the characters used by all countries/regions in the world and is an international encoding with a high universality. 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
- Whether restart is required: yes
- Role: 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_DATE, NO_ZERO_DATE, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENG
INE_SUBSTITUTION (v5.7)
```

- Whether restart is required: no
- Role: 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`.

    Where:
    - 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 whose password is empty.
    - `NO_ENGINE_SUBSTITUTION` means that if the storage engine is disabled or not compiled, an error will be thrown.
- 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
- Whether restart is required: no
- Role: 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.

Multi-AZ Deployment Overview

Last updated：2022-06-07 15:42:00

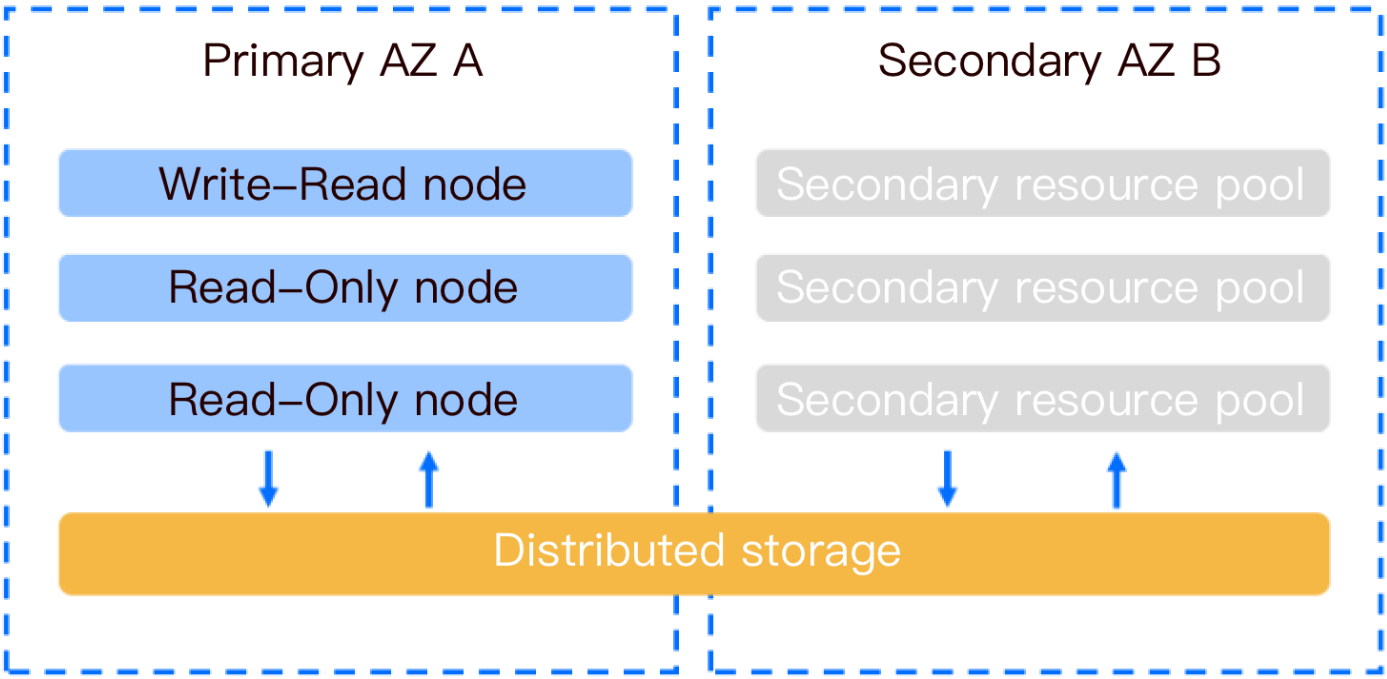
The engine of TDSQL-C for MySQL allows deploying a cluster across AZs. A multi-AZ cluster has superior disaster recovery capabilities than a single-AZ cluster and can protect your database against database instance failures, AZ outages, and even IDC-level failures.

The multi-AZ deployment scheme of TDSQL-C for MySQL guarantees the high availability and failover capability of database instances by combining multiple AZs into a single "multi-AZ".

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 Region	Supported AZ
North China (Beijing)	Beijing Zone 5
	Beijing Zone 7
South China (Guangzhou)	Guangzhou Zone 4
	Guangzhou Zone 6
East China (Shanghai)	Shanghai Zone 2
	Shanghai Zone 4

How to Implement Multi-AZ Architecture

You can create a cluster in multi-AZ deployment mode in the TDSQL-C for MySQL console. Your existing single-AZ clusters will also be upgraded to the multi-AZ mode. The upgrade will be automatically completed through online data migration without affecting your business. For more information, see Setting Multi-AZ Deployment.

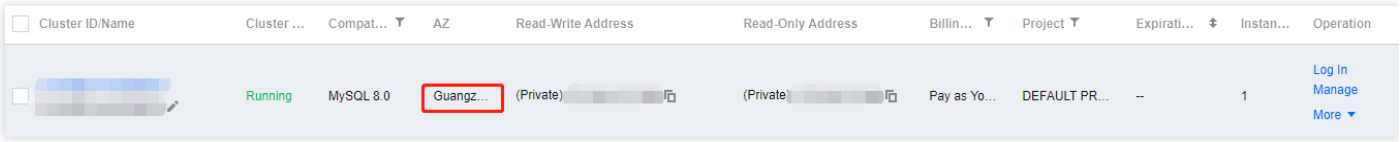
Multi-AZ Billing Description

There are no additional fees for the multi-AZ feature for the time being.

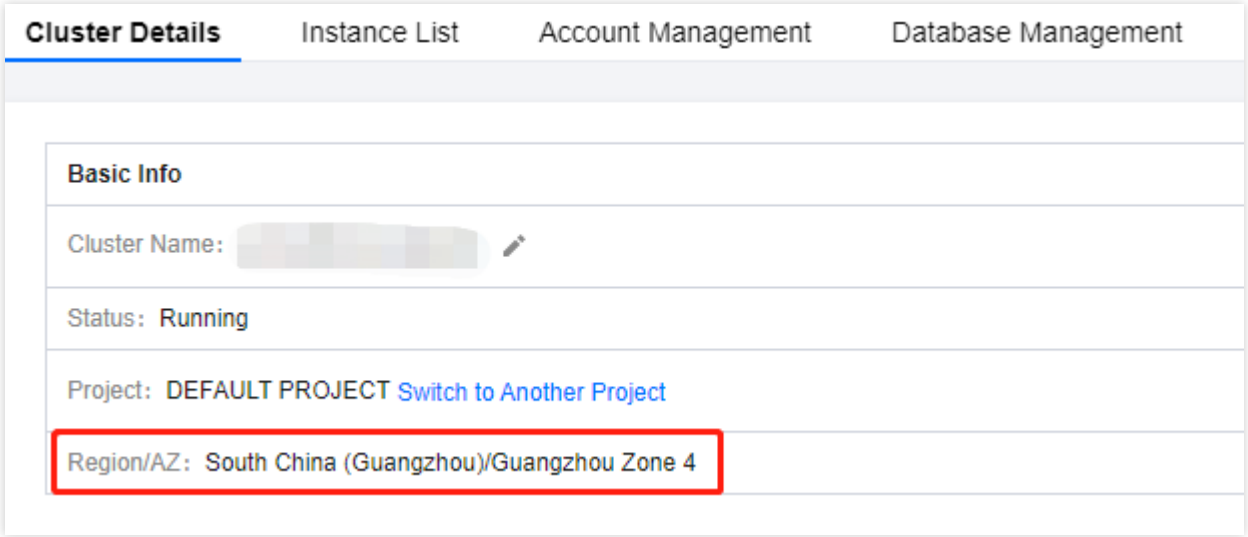
> Note：
> Currently, single-AZ clusters can also be upgraded to multi-AZ clusters for free.

Multi-AZ Information Display

1. The **cluster list** page displays the information of the cluster's primary AZ and supports filtering.



2. On the cluster details page, you can view the AZs where data is distributed in **Basic Info** > **Availability Info**.



Setting Multi-AZ Deployment

Last updated：2022-07-07 17:04:49

This document describes how to select the primary and secondary AZs of a cluster on the cluster purchase page in the TDSQL-C for MySQL console.

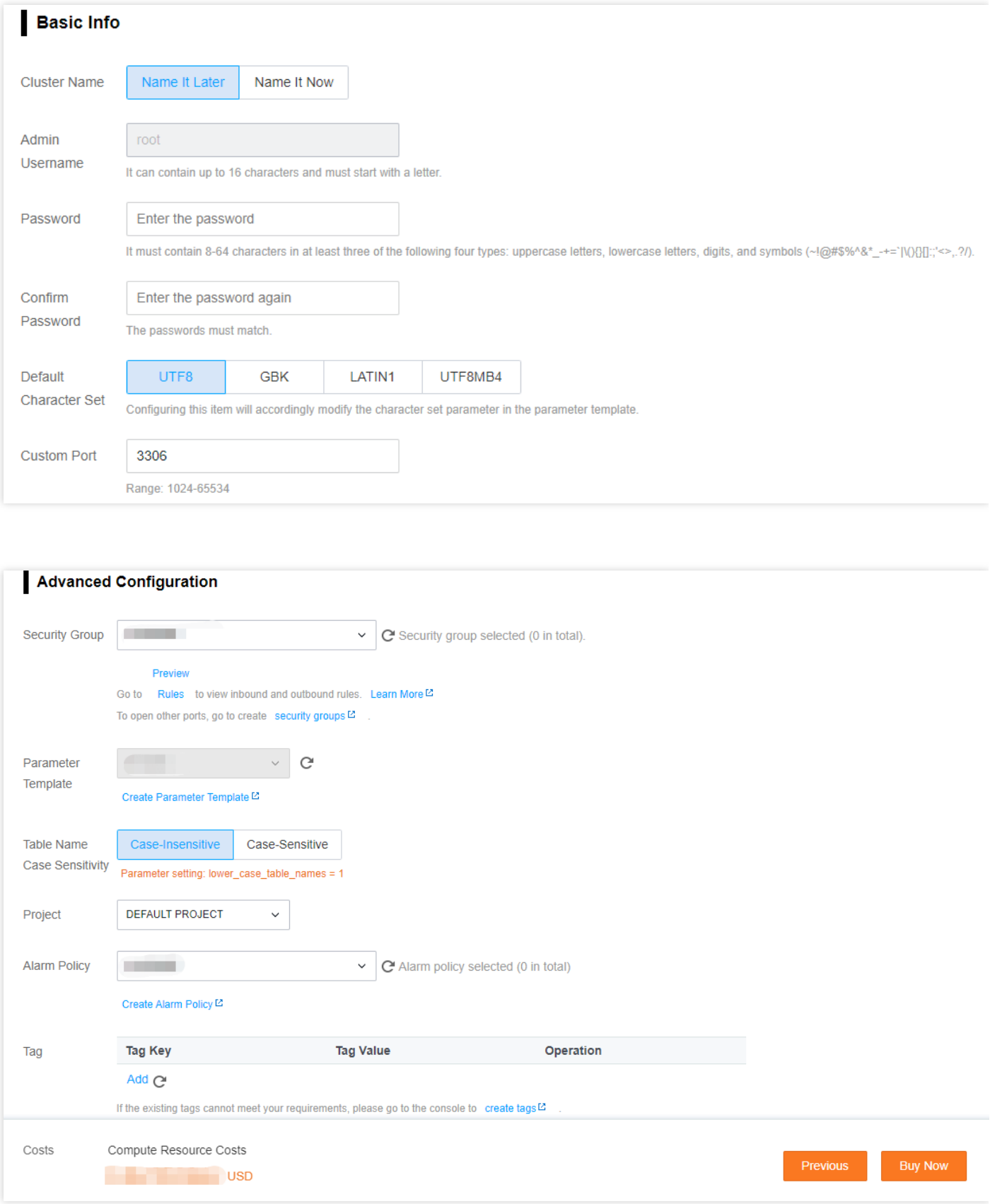
Overview

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.

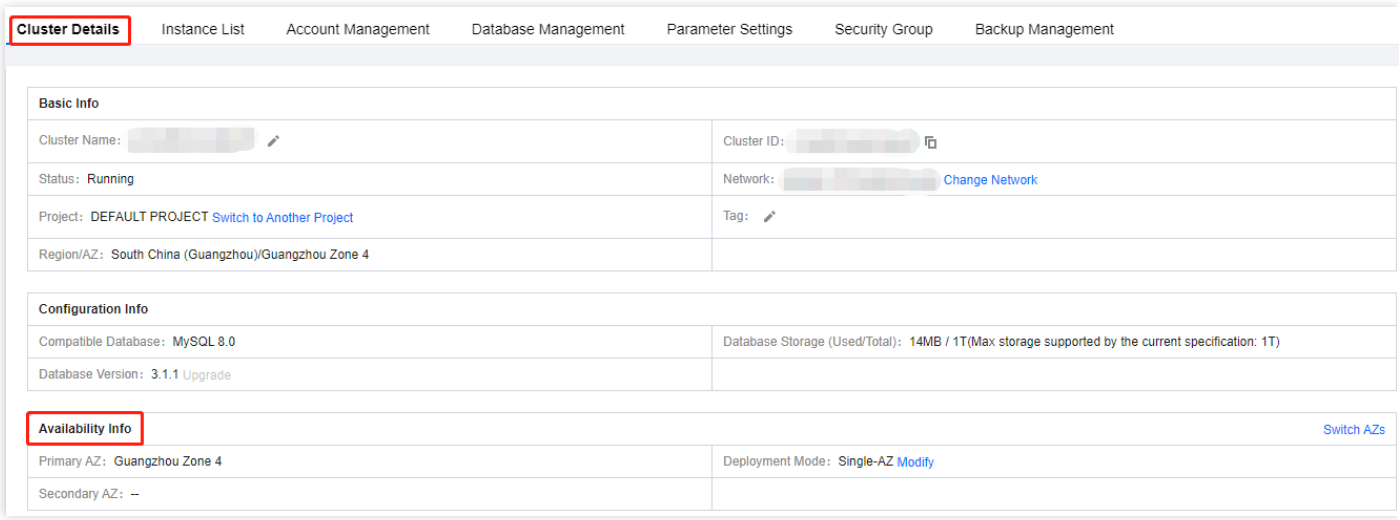
Selecting AZs on purchase page when creating cluster

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.



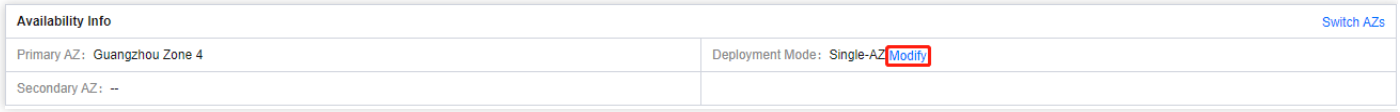
4. After completing the configuration and confirming that everything is correct, click **Buy Now**.

5. After the purchase is completed, you will be redirected to the cluster list. After the status of the newly created cluster becomes **Running**, you can go to the cluster list page or click the cluster ID to enter the **Cluster Details** page, where you can query the AZs in **Availability Info**.



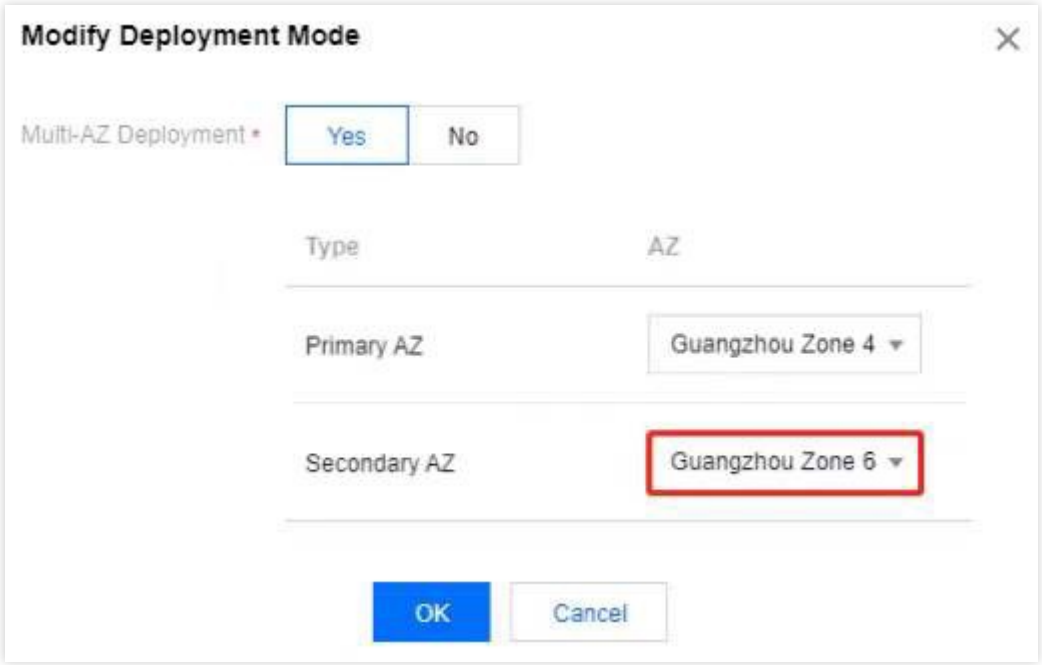
Modifying AZ information

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** to modify the AZ.



3. In the pop-up window, select the desired configuration and click **OK**.

> Note：
> Currently, only the secondary AZ can be modified, but not the primary AZ.



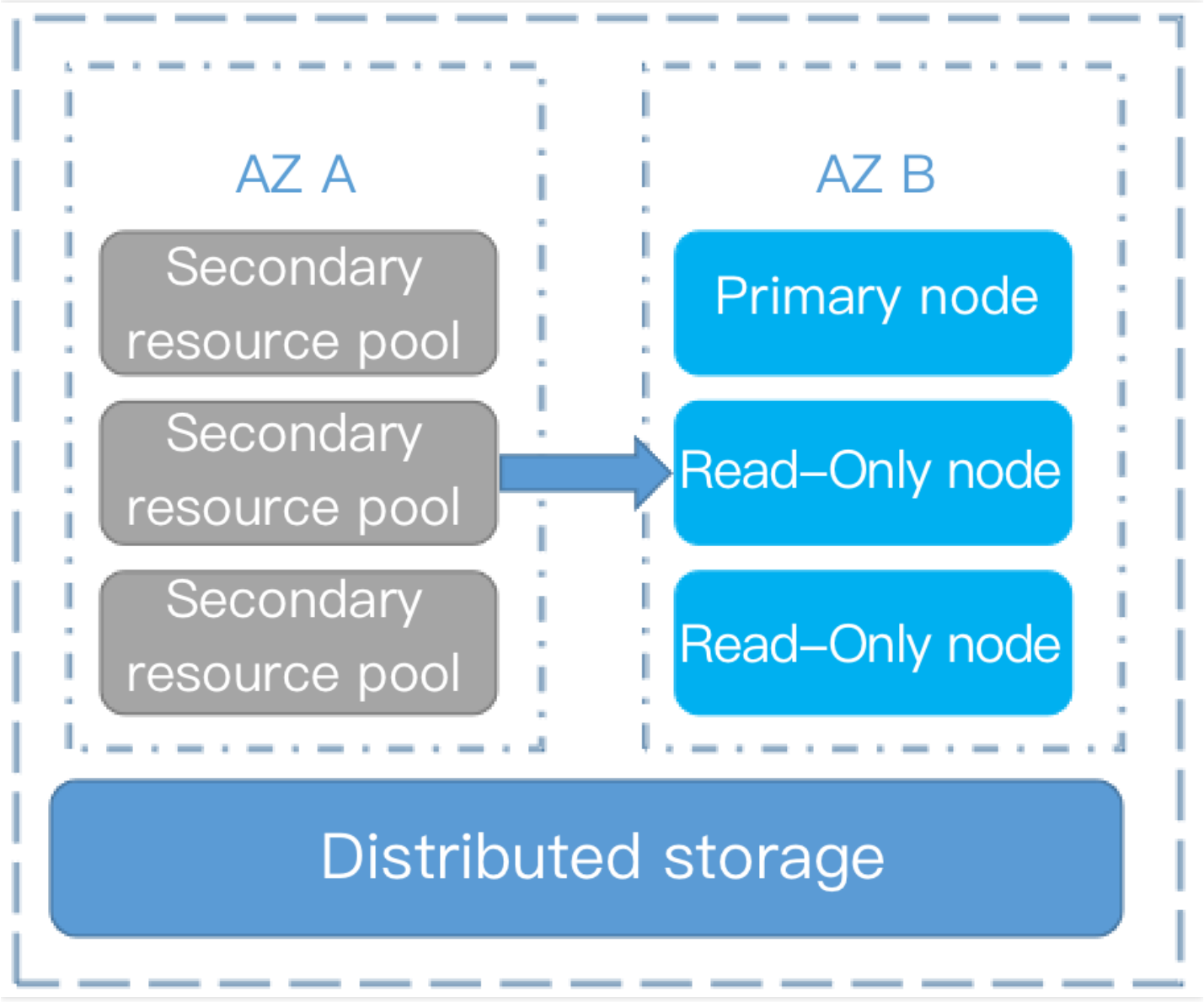
Switching AZs

Last updated：2022-01-18 16:31:24

TDSQL-C for MySQL allows you to change the primary and secondary AZs manually or automatically. You can use this feature to migrate the compute nodes of your database cluster to other AZs.

Use Cases

The primary/secondary AZ switch feature is suitable for use cases such as disaster recovery and device failure recovery.



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 entire cluster is switched; that is, both read-write and read-only nodes are switched (partial node switch is not supported).
- The read-write and read-only addresses will not be changed after the switch.

Directions

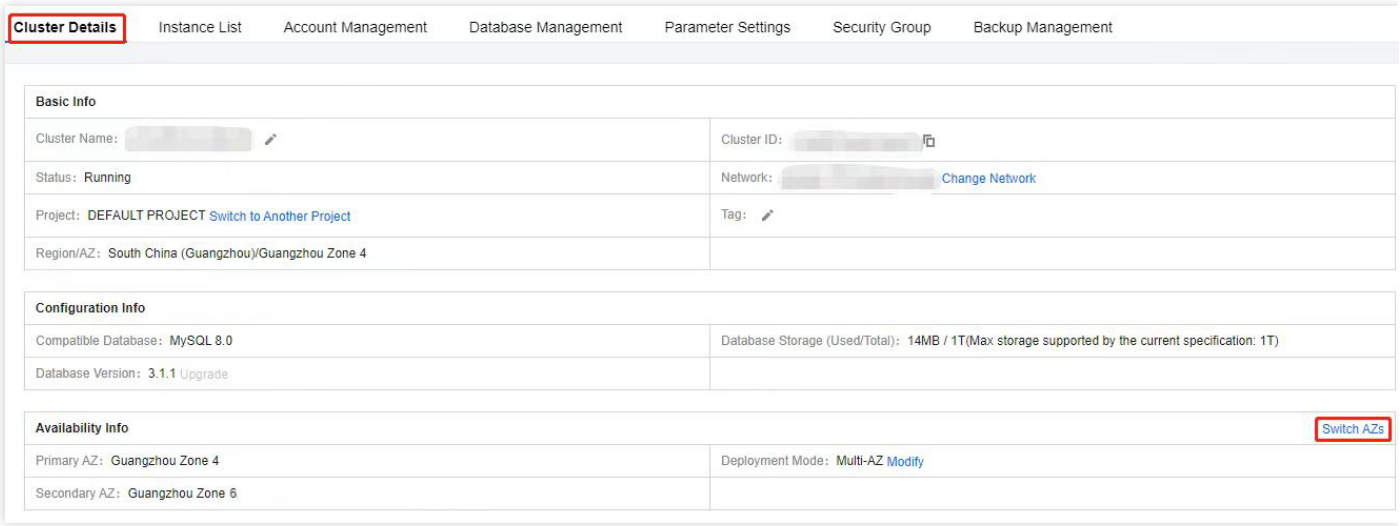
Automatic switch

This feature supports automatic primary/secondary AZ switch in case of AZ failures, ensuring your business continuity and protecting your database.

Manual switch

You can manually switch the cluster's primary and secondary AZs as needed by your business.

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 **Switch AZs**.

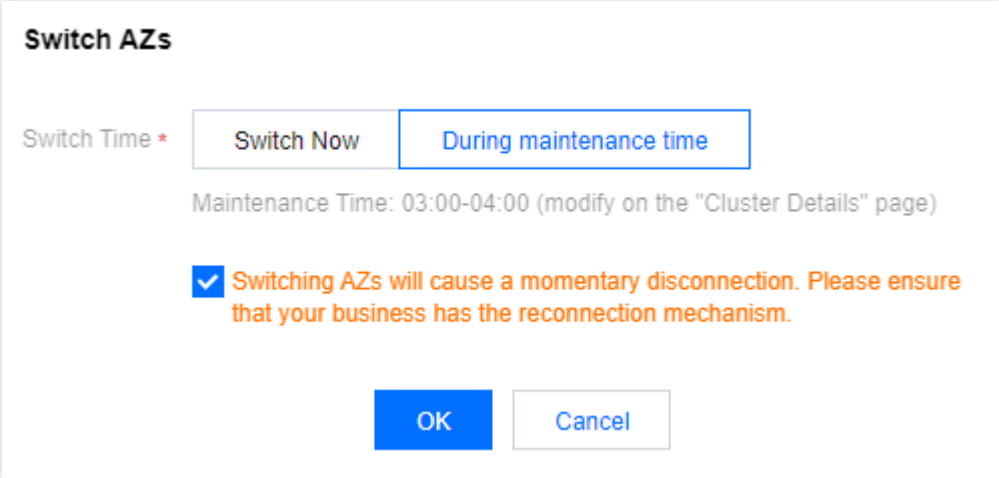


3. In the pop-up window, select **Switch now** or **During maintenance time**, click **Switching AZs will cause a momentary disconnection. Please ensure that your business has the reconnection mechanism.**, and

click **OK**.

Note：

- Switching AZs will cause a momentary disconnection. Ensure that your business has the reconnection mechanism.
- Select the **Instance List** page, click the target instance ID to enter the instance details page, and click **Edit** after **Maintenance Info** to set or modify the maintenance time.



4. When the cluster status becomes **Running**, you can view the information of the switched AZ in **Availability Info** 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 is switched, all read-write and read-only nodes will be switched, but the read-write and read-only addresses will remain unchanged.

After manual/automatic switch, if an instance is upgraded, will the upgrade affect or reset the information of the primary and secondary AZs?

No.

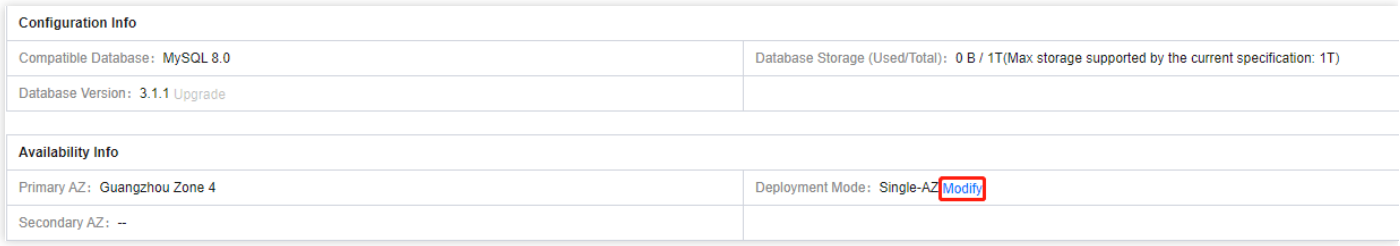
Modifying AZ Deployment

Last updated：2022-01-18 16:31:25

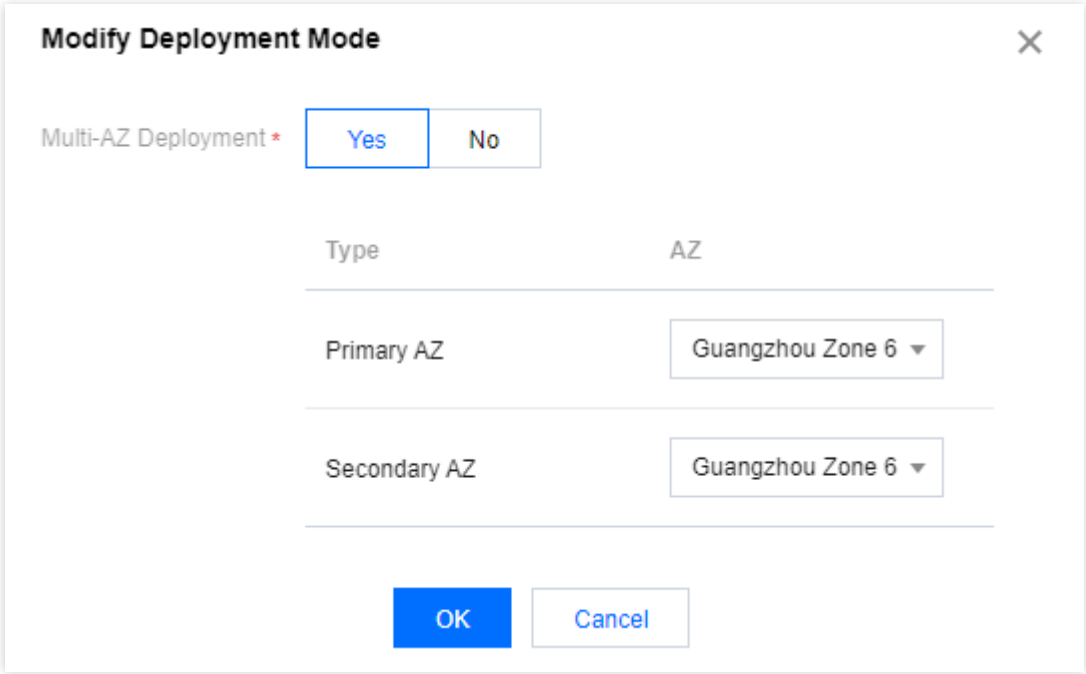
This document describes how to switch the instance deployment mode between single-AZ and multi-AZ in the TDSQL-C for MySQL console.

Upgrading from Single-AZ to Multi-AZ

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**.



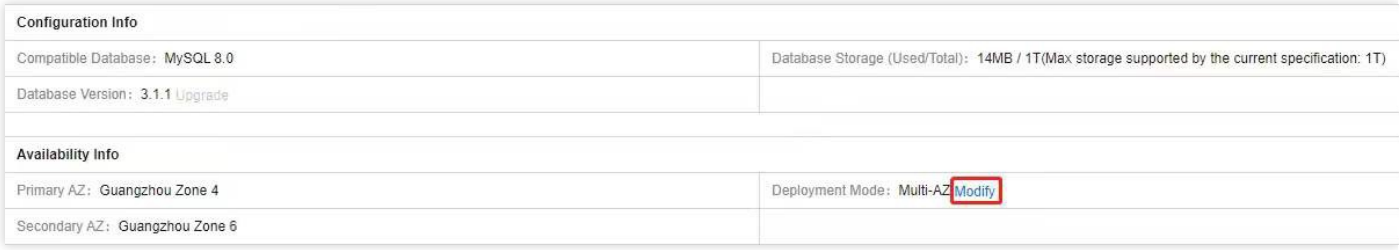
3. In the pop-up window, select **Yes** for **Multi-AZ Deployment**, select the secondary AZ, and click **OK**.



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

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 **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?

- 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.

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：2022-07-26 19:56:01

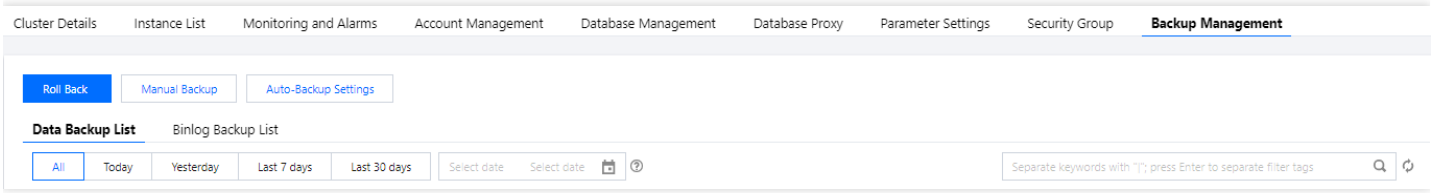
Data is the core asset of enterprises. As your business develops, your data grows in a large-scale and explosive manner. Business applications require real-time, online, and fast data processing. It is more and more challenging for database Ops engineers to protect the data integrity, as data loss may 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 lower 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%.

Notes

- 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 according to your business needs.

Rollback Method

- Rollback by backup file: This method restores the cluster to the data file state of any backup file within 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：2022-07-26 19:56:01

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：2022-12-06 14:46:38

TDSQL-C for MySQL supports automatic backup and manual backup. Automatic backup is enabled by default. You can configure backup policies for TDSQL-C for MySQL to automatically back up data as scheduled for guaranteed data security.

For a newly created cluster, TDSQL-C for MySQL automatically backs up data once a day (between 2:00 AM and 6:00 AM by default). You can customize the backup start time and retention period as needed in the console.

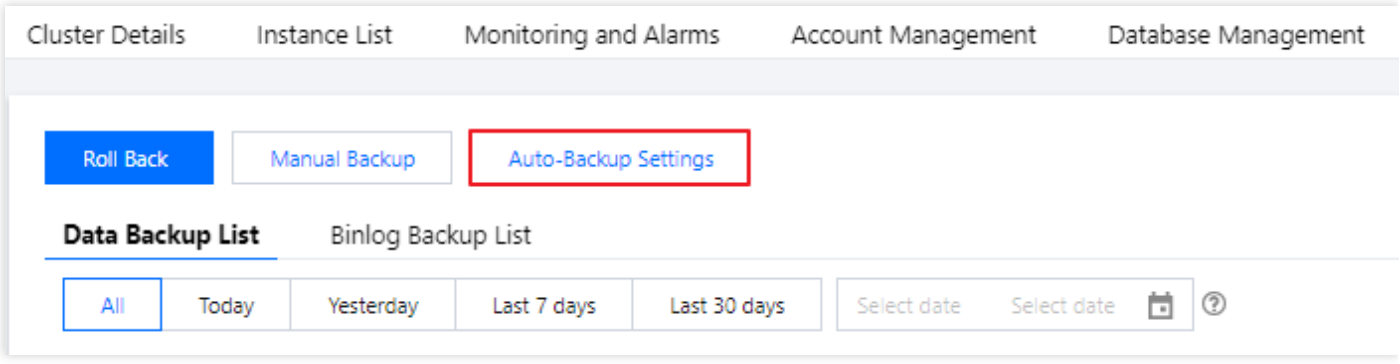
This document describes how to configure automatic backup in the console.

Notes

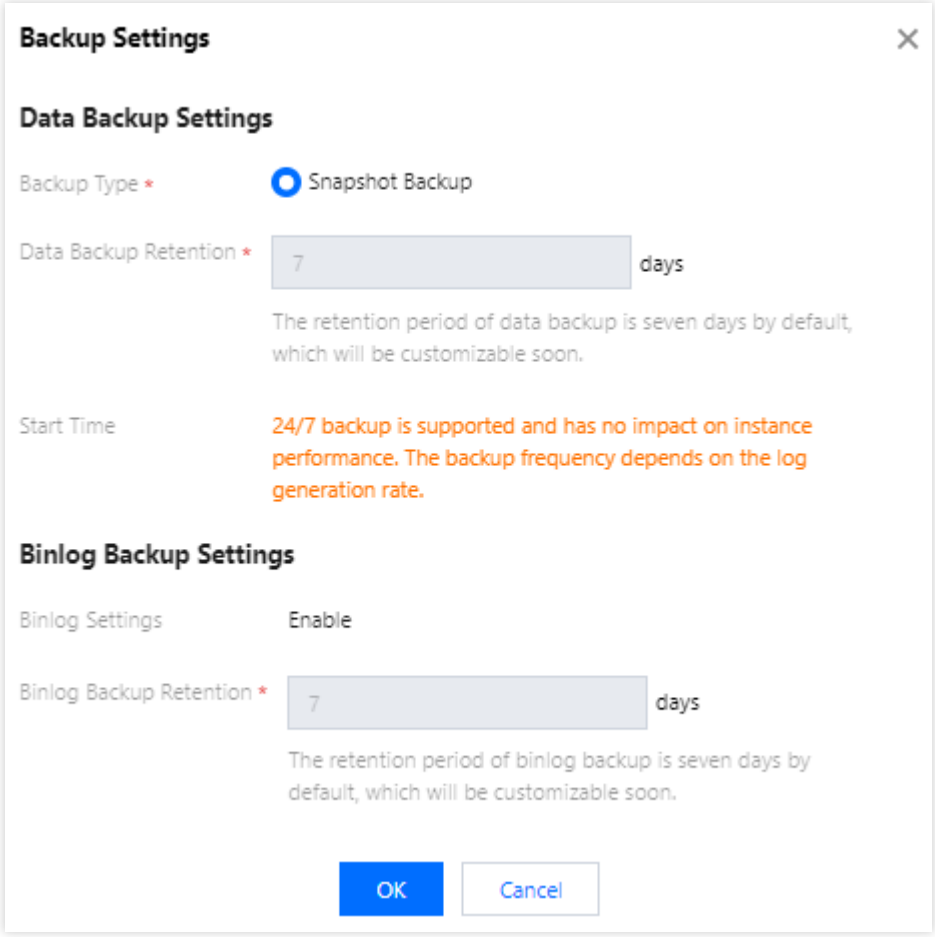
- Automatic backup files cannot be deleted, but you can set the data and binlog backup retention period, so that they will be deleted upon expiration.
- You can view the **Backup Time** field in the data backup list to determine the exact time point of the data image in a backup file.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select 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 **Auto-Backup Settings**.

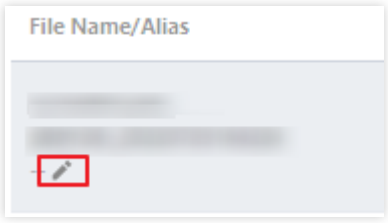


4. In the pop-up window, configure the following items and click **OK**.

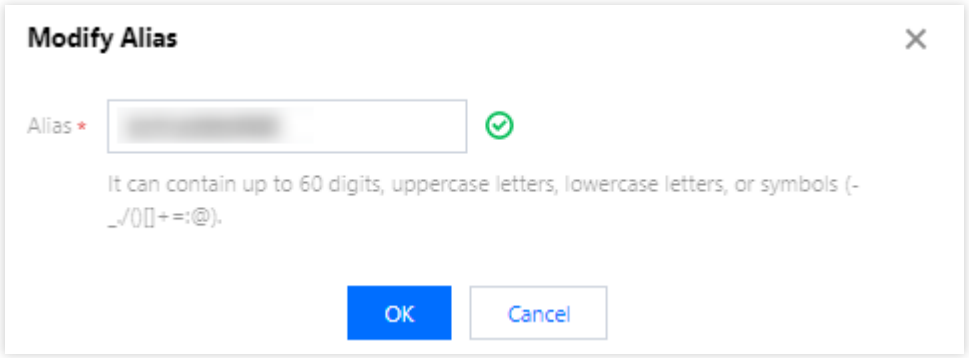


Parameter	Description
Backup Type	Snapshot backup is selected by default.
Data Backup Retention	Data backup files can be retained for 7–1,830 days.
Start Time	◦ Backup start time. You can customize a time range as needed. We recommend you back up your database during off-peak hours. ◦ The default time range is 02:00–06:00 AM. This is just the start time of the backup process and does not indicate the end time. For example, if the time range is set to 02:00–06:00 AM, the system will initiate a backup at a time point during 02:00–06:00 AM, which depends on the backend backup policy and backup system conditions.
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：2022-07-26 19:56:02

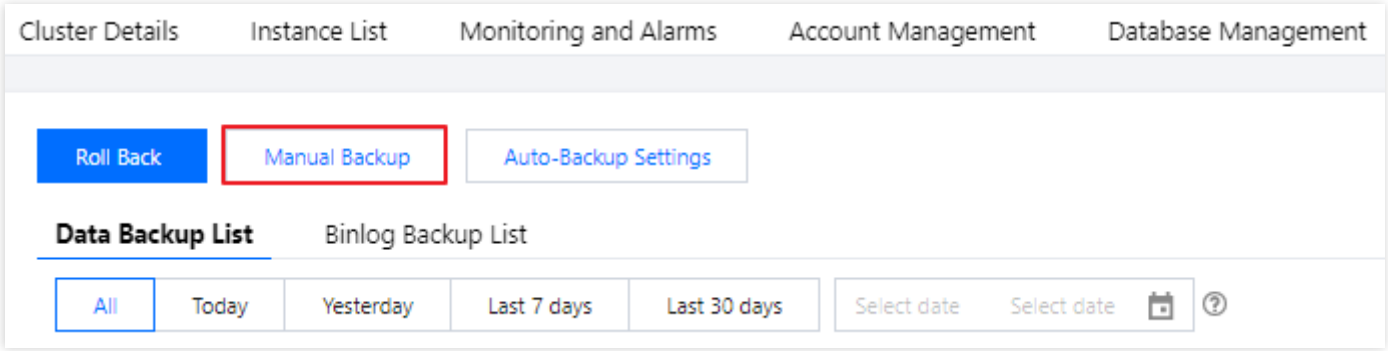
TDSQL-C for MySQL supports automatic backup and manual backup. This document describes how to create a manual backup in the console.

Note：

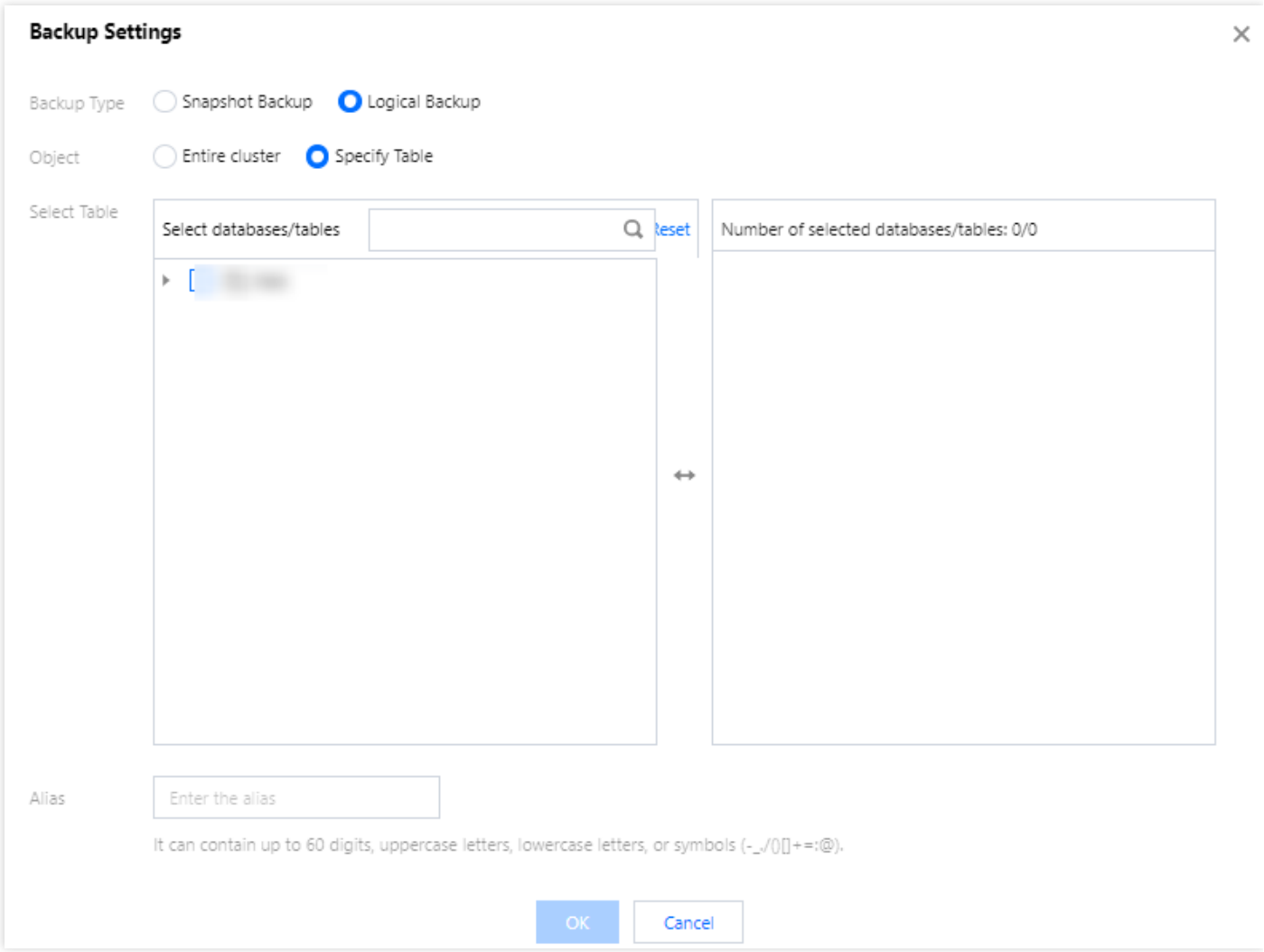
- With manual backup, you can use snapshot backup to manually back up the entire cluster or use logical backup to back up the entire cluster or specified databases/tables.
- You can delete manual backup files from the backup list to free up the space.

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**.



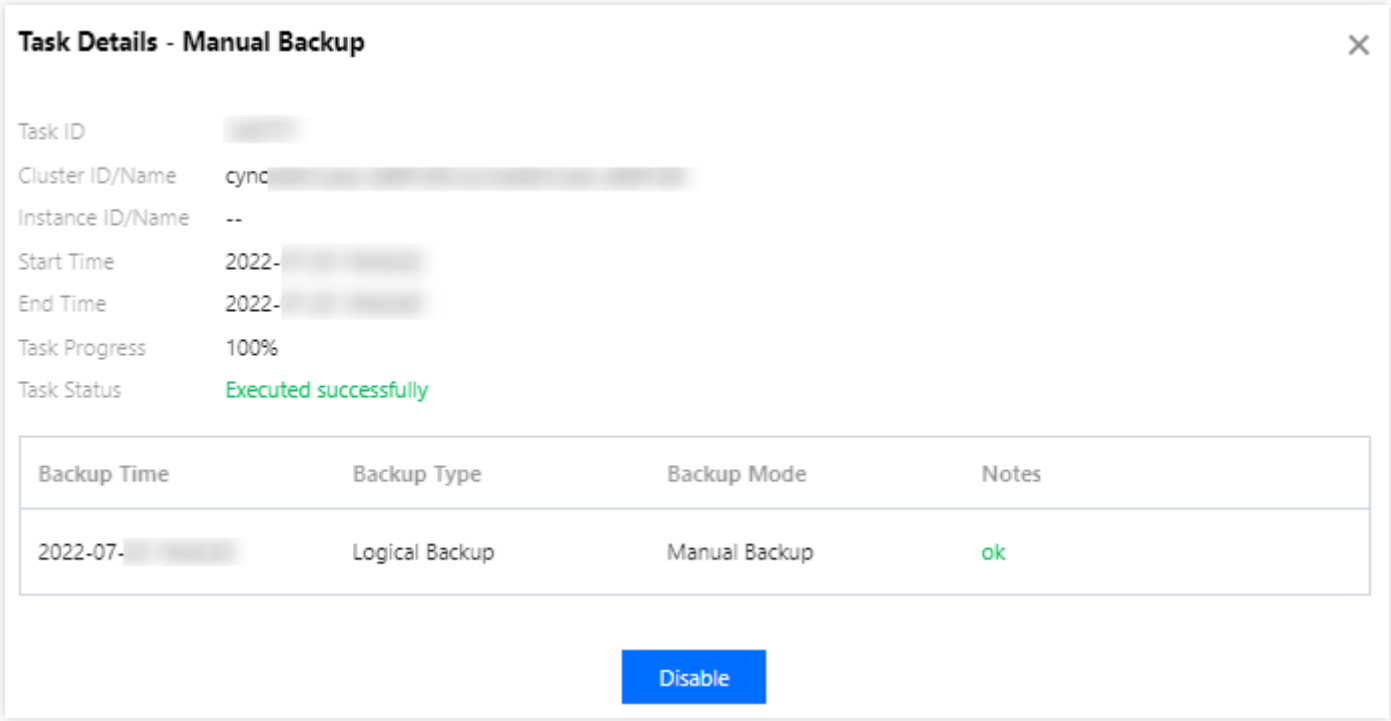
4. In the pop-up window, select the backup type and object, set the backup file alias, and click **OK**.



- **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.



Managing Backup
Viewing Backup List

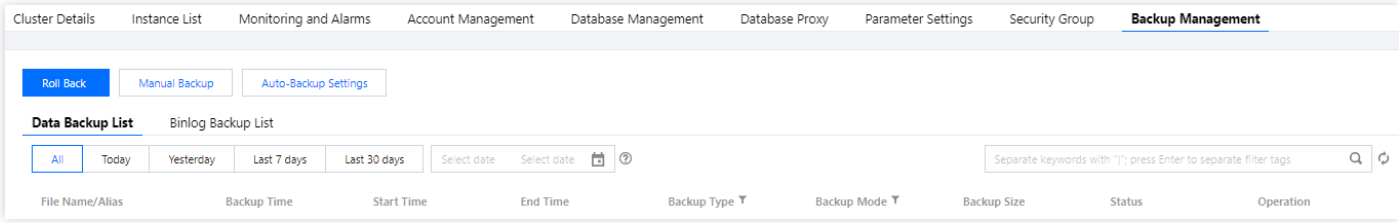
Last updated：2022-07-26 19:56:02

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 backup list in the console.

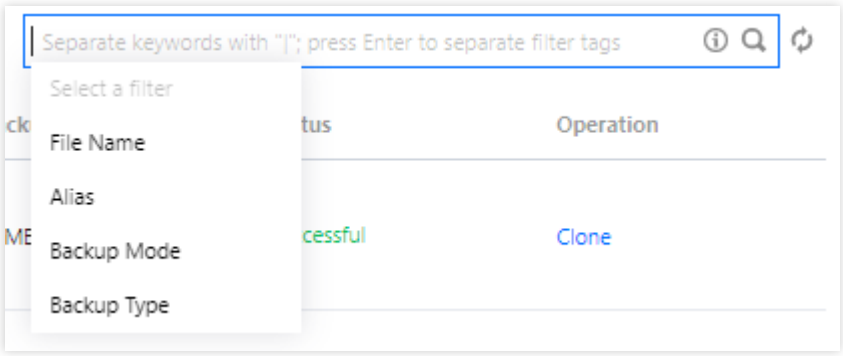
Directions

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 **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 full and snapshot full backup files in the TDSQL-C for MySQL console.
- You can manually back up data in the console. Manual backup files will be retained in the backup list as long as they are not manually deleted.

> Note：
> Manual backups will also take up the backup space. We recommend you plan the usage of the backup space appropriately to reduce costs.

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 backups can be manually deleted from the backup list in the TDSQL-C for MySQL console.

How can I reduce the backup space costs?

- Delete manual backups that are no longer used as instructed in 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：2022-07-26 19:56:02

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.

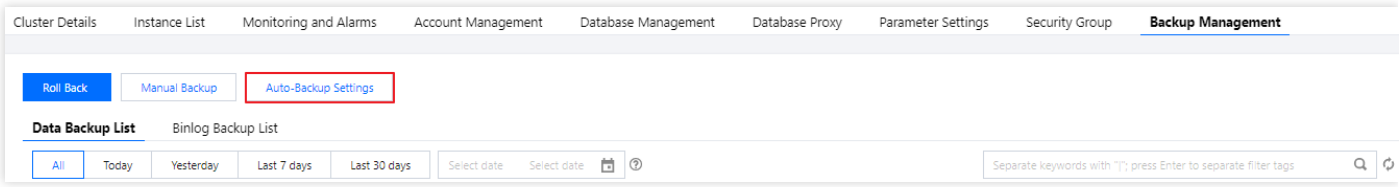
This document describes how to set the backup retention period in the console.

Note：

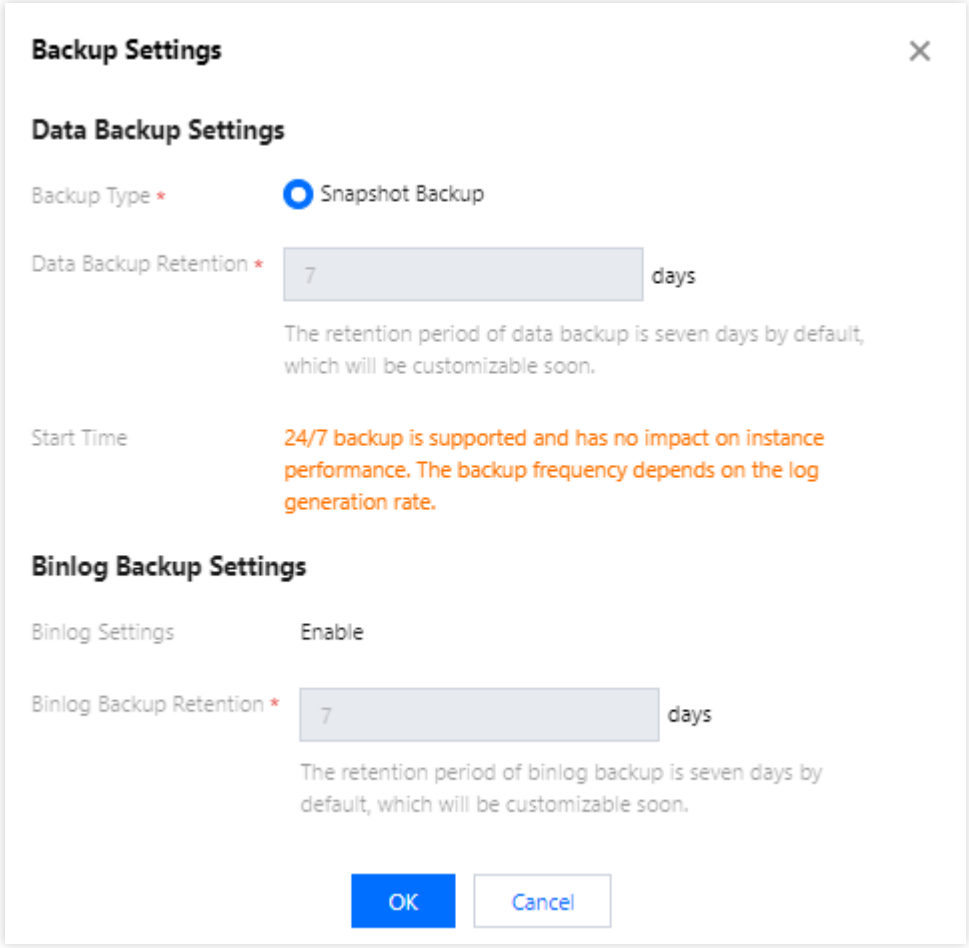
- The backup retention period is 7 days by default. The feature of modifying the backup retention period is currently in beta test. To try it out, submit a ticket. Then, you can adjust the period in the range of 7–1,830 days.
- The binlog backup retention period cannot be shorter than the data backup retention period.

Directions

1. Log in to the TDSQL-C for MySQL console.
2. Select the region, 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：2022-08-22 18:05:02

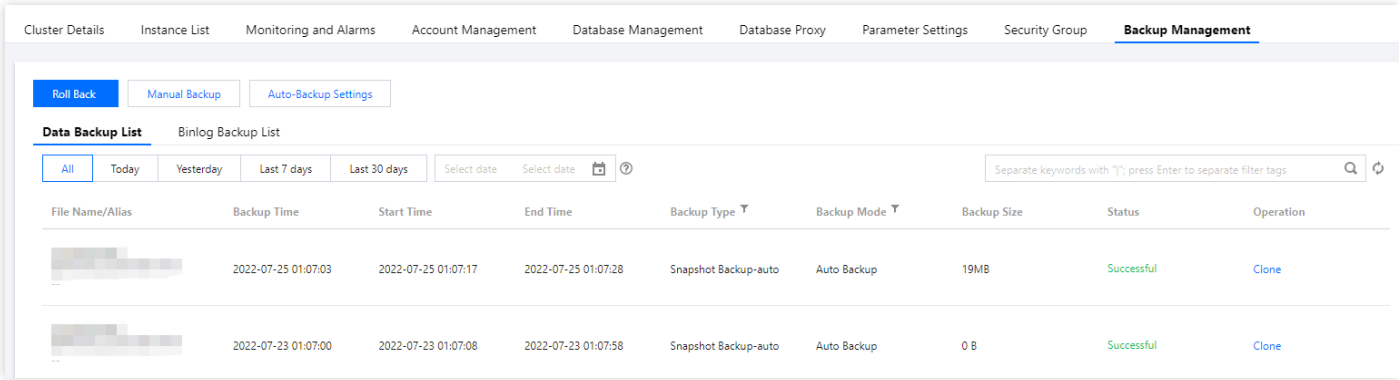
The TDSQL-C for MySQL console provides the list of logical backup files that can be downloaded. They may then be used to restore data from one database to another (such as a self-built database or a database provided by another cloud vendor).

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

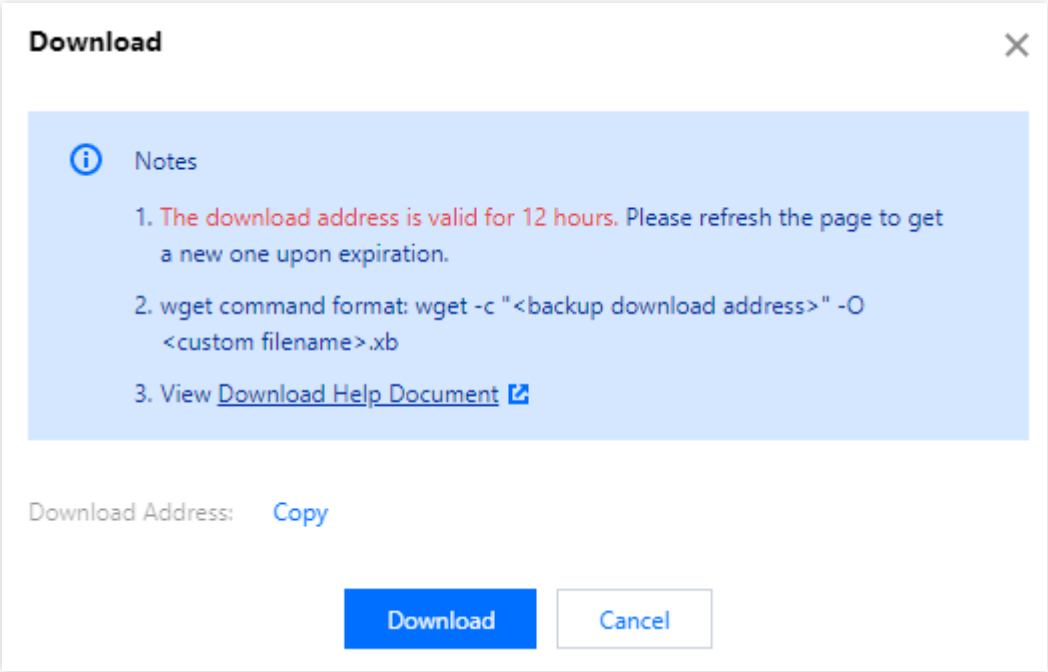
1. Log in to the TDSQL-C for MySQL console.
2. Select the region at the top, find the target cluster, and click the cluster ID or **Manage** in the **Operation** column to enter the cluster management page.
3. On the cluster management page, select **Backup Management** > **Data Backup List** to view 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 over the private network by running the `wget` command, or directly click **Download**.

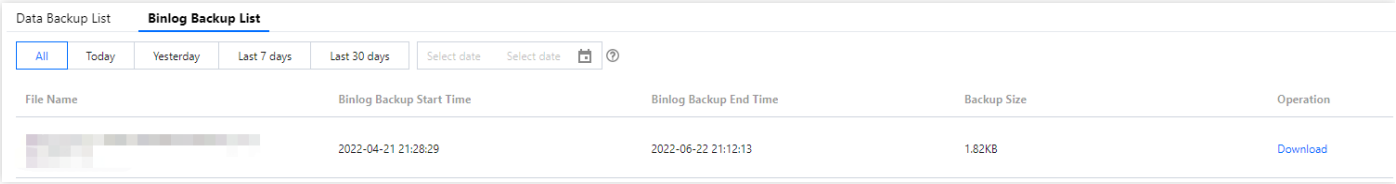
> Note：
>
> - After copying the download address, log in to a Linux CVM instance in the same VPC as the TencentDB instance as instructed in Customizing Linux CVM Configurations, and run the `wget` command for download over the private network.
> - The download address is valid for 12 hours, after which you will need to enter the download page again to get a new one.
> - The URL must be enclosed with quotation marks when wget 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.

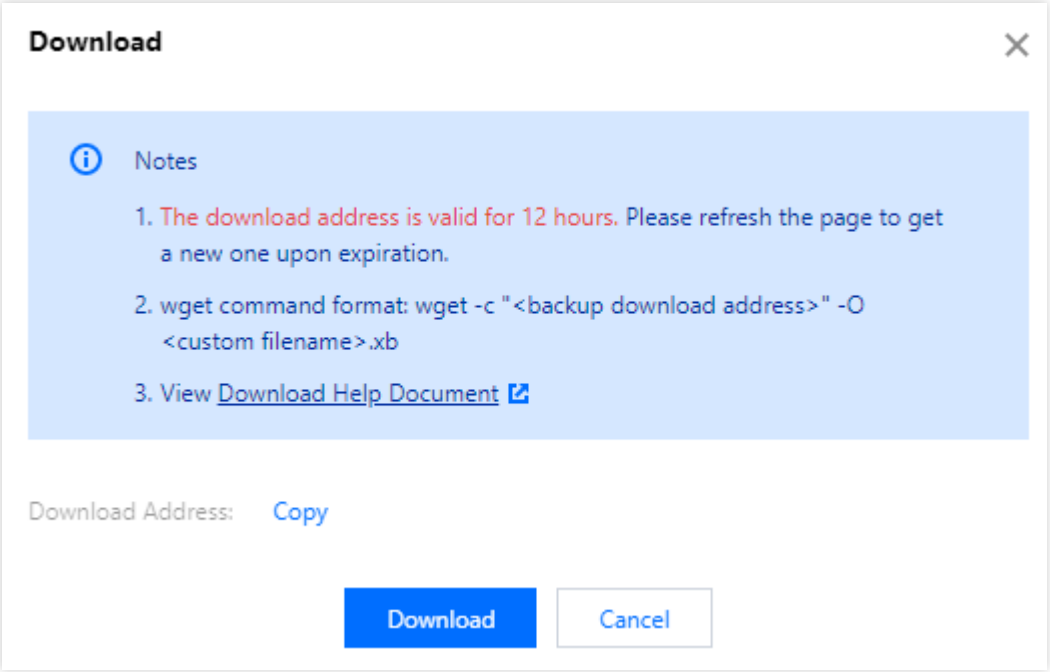
3. On the cluster management page, select **Backup Management** > **Binlog Backup List** to view the backup file.



4. In the **Operation** column of a binlog backup file, click **Download**.
5. In the pop-up window, click **Copy** to get the file download address for fast download over the private network by running the `wget` command, or directly click **Download**.

Note：

- After copying the download address, log in to a Linux CVM instance in the same VPC as the TencentDB instance as instructed in Customizing Linux CVM Configurations, and run the `wget` command for download over the private network.
- 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 wget 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

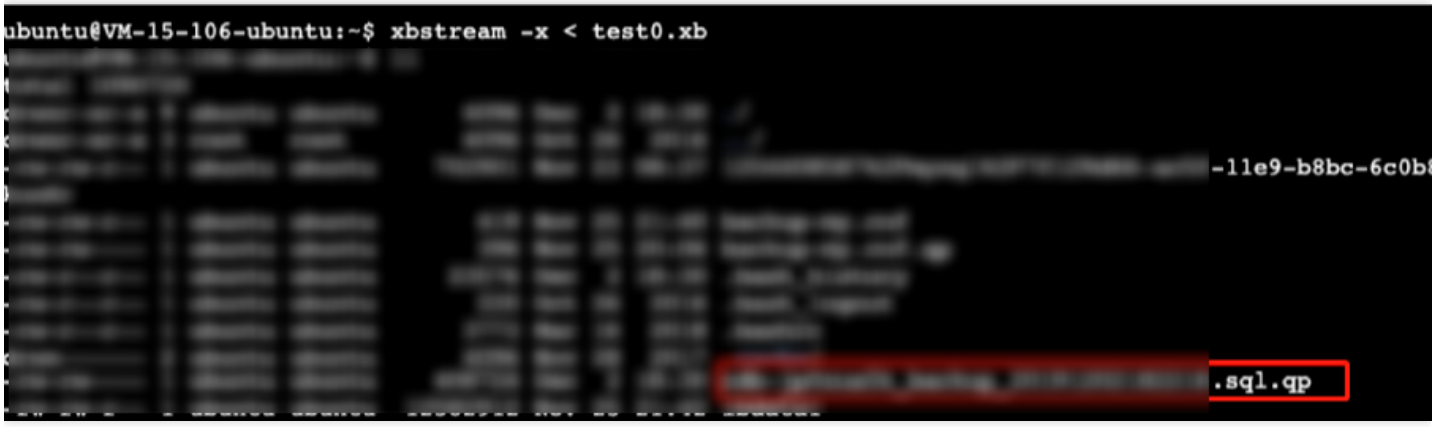
Unpack the backup file with xbstream.

> 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 2.4.

```
xbstream -x < test0.xb
```

> 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/10.0" http://www.quicklz.com/qpress-11-linux-x64.tar
```

Note：
If an error is displayed during the `wget` download, you can go to QuickLZ's official website 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

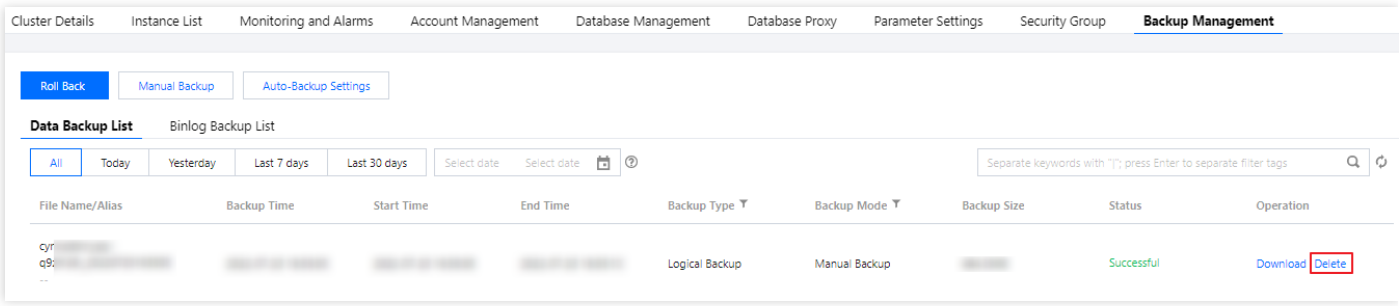
Last updated：2022-07-26 19:56:02

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.

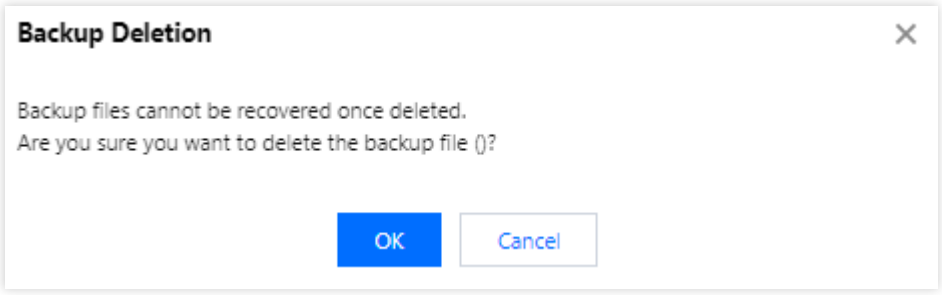
This document describes how to delete manual backups in the console.

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.

Restoring Data from Logical Backup

Last updated：2022-08-29 10:54:47

This document describes how to restore data from a logical backup file.

Overview

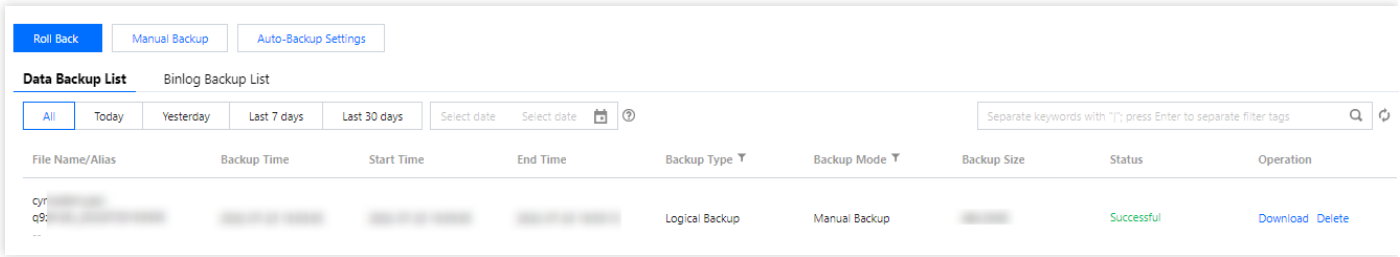
> Note：
> To save storage space, TDSQL-C for MySQL backup files will be compressed with qpress and then packed with xbstream offered by Percona.

TDSQL-C for MySQL supports logical backup. You can manually generate and download logical backup for the entire cluster or specified databases/tables. This document describes how to restore data from a logical backup file on Linux.

Directions

Step 1. Download the backup file

1. Log in to the TDSQL-C for MySQL console and click a cluster ID in the cluster list or **Manage** in the **Operation** column to enter 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 in the same VPC as the TencentDB instance as instructed in Customizing Linux CVM Configurations, and run the `wget` command for download over the private network.

> Note：

- You can also click **Download** to download it directly. However, this may take longer.
- wget command format: wget -c "" -O .xb

The `wget` command format is as follows:

```
wget -c "backup file download address" -O custom filename.xb
```

Step 2. Unpack the backup file

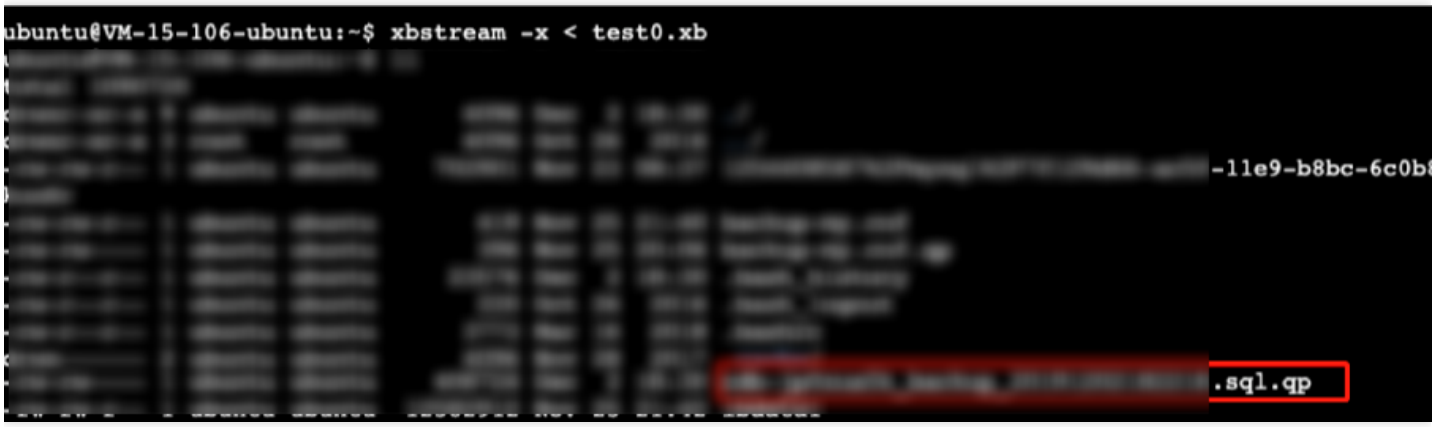
Unpack the backup file with xbstream.

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 2.4.

```
xbstream -x < test0.xb
```

Note：
Replace `test0.xb` with your backup file name.

The unpacking result is as shown below:



Step 3. Decompress 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/10.0" http://www.quicklz.com/qpress-11-linux-x64.tar
```

Note：
If an error is displayed during the `wget` download, you can go to QuickLZ's official website 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 .
```

Step 4. Import 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
< <.sql file compressed with qpress>
```

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

Operation	Use Case	Method
Prevent slow logs from being generated	Get suggestions from DBbrain for slow SQL statement optimization.	-
Set slow log parameters	Set the value of `long_query_time` to record slow logs.	Setting 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.

Migrating Data
Migrating with DTS

Last updated：2022-04-01 22:15:31

The data migration feature of DTS can be used to import data to TDSQL-C for MySQL. For more information.

Migrating with Command Line Tool

Last updated：2022-06-07 15:18:59

TDSQL-C for MySQL supports data migration through a command line tool.

Data Migration with 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 MySQL's official documentation.

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 Command Line Tool.

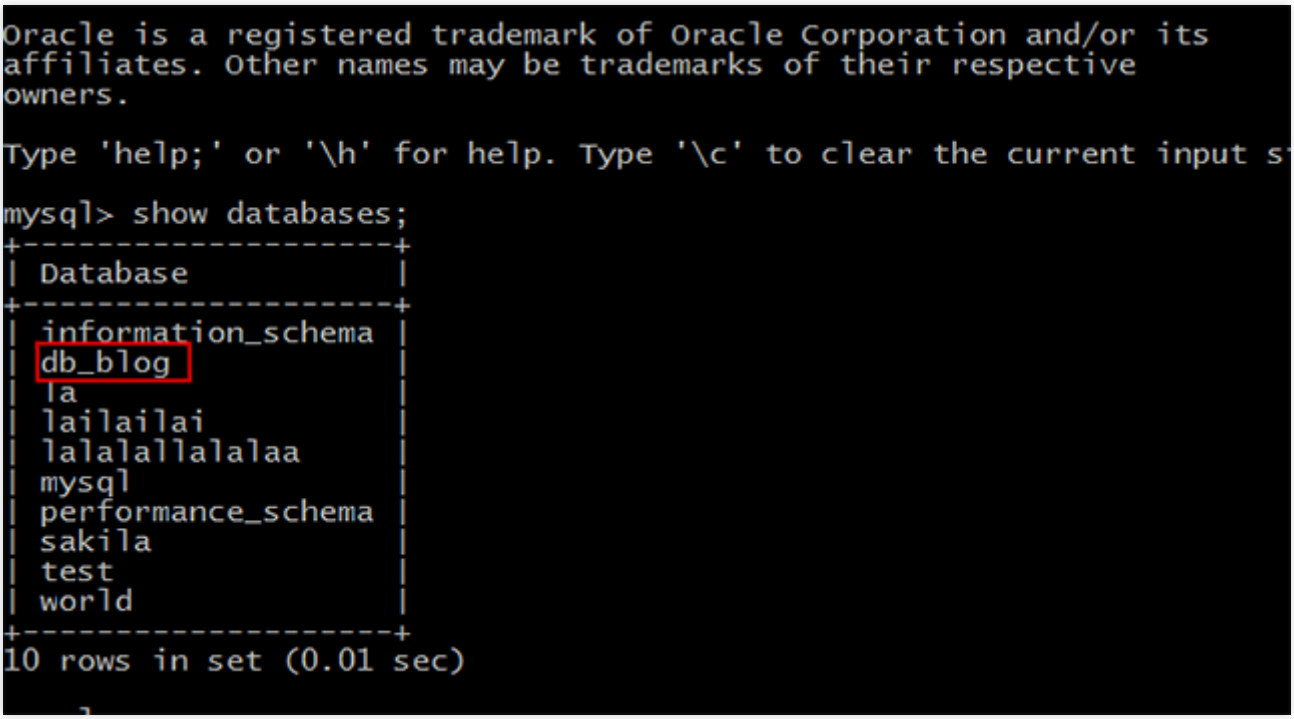
> Note：

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 by using the parameter

.

2. Enter the command prompt and restore the data with the MySQL command line tool.

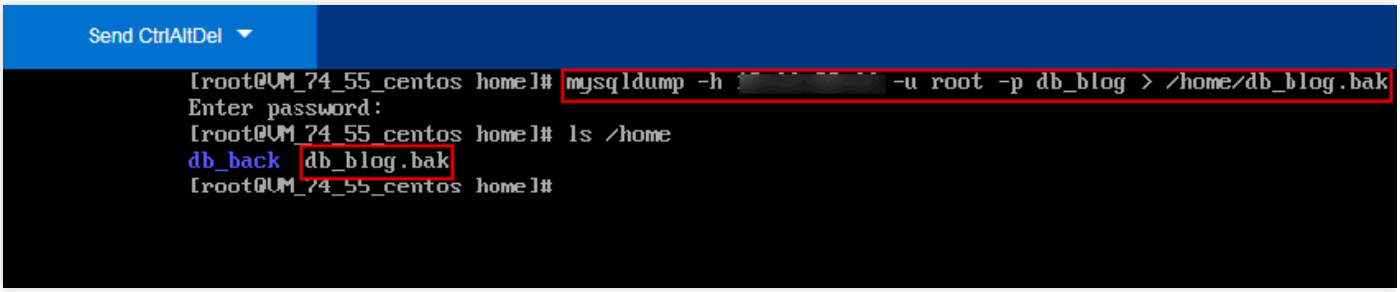
3. Log in to the MySQL database and you can see the backed up database has already been restored to the server.



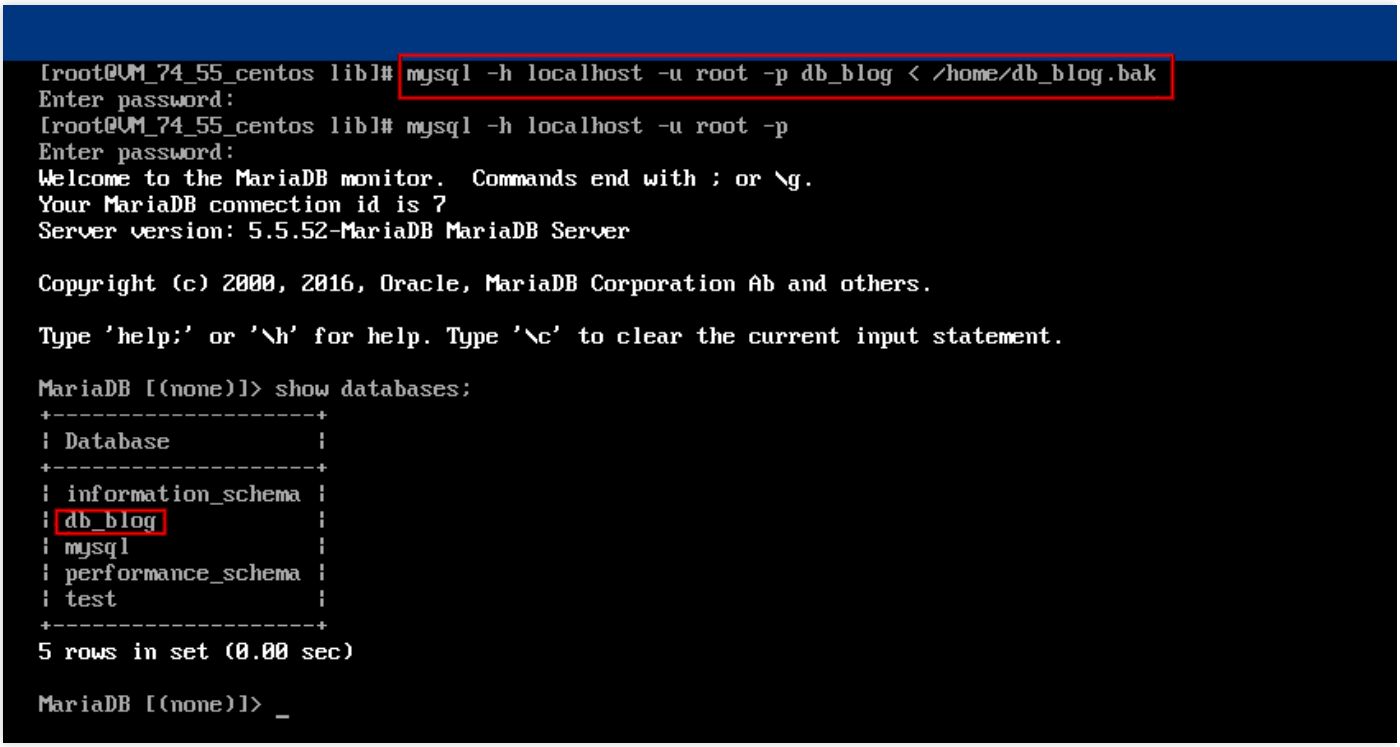
Migrating data on Linux CVM instance

For more information on how to access a database on a CVM instance, see Connecting to Cluster.

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 Character Set of Imported Data Files

1. If no character set is specified during data file import into the database, the one set 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.

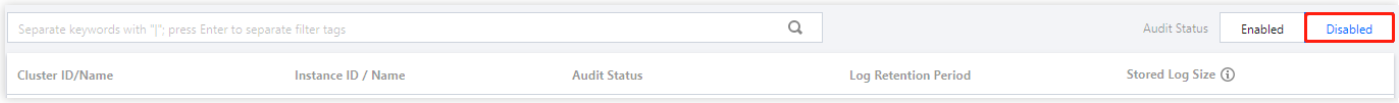
Database Security and Encryption
Database Audit
Enabling Audit in TDSQL-C for MySQL

Last updated：2022-06-14 16:03:12

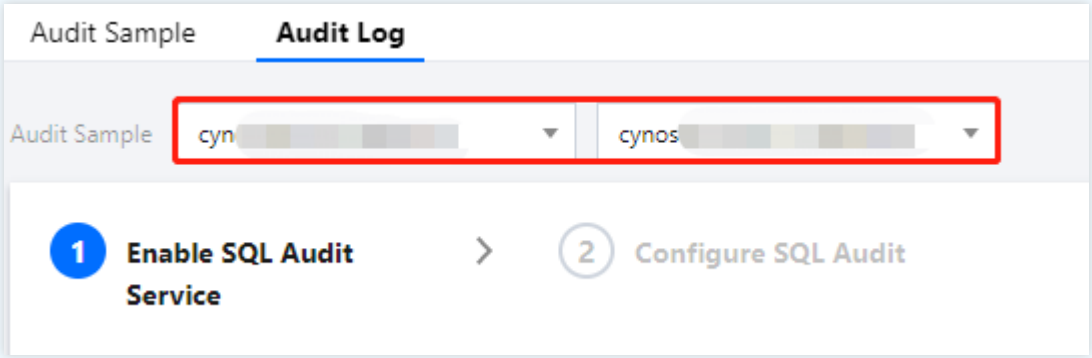
Tencent Cloud provides database audit capabilities for TDSQL-C for MySQL, which can record accesses to databases and executions of SQL statements to help you manage risks and improve the database security.

Enabling SQL Audit

1. Log in to the TDSQL-C for MySQL console, select **Database Audit** on the left sidebar, select a region at the top, click the **Audit Instance** tab, and click **Disabled** to filter instances whose audit is disabled.



Note：
Alternatively, in **Audit Instance** on the **Audit Log** tab, directly search for instances whose audit is disabled and then enable audit.



2. On the **Audit Instance** tab, click the ID of the target instance to enter the enablement page, select a log retention period, and click **Enable**.

Note：

- You can select 7 days, 30 days, 3 months, 6 months, 1 year, 3 years, and 5 years as the audit log retention period. You can also modify it in the console after enabling audit.
- In order to meet the security compliance requirements for the retention period of SQL logs, we recommend you select 180 days or above.

Viewing Audit Log

After enabling audit, you can view the corresponding SQL audit logs on the **Audit Log** tab.

Access and Authorization Overview

Last updated：2022-06-07 15:21:43

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:

- The risk of your key being compromised is high since multiple users are sharing it.
- Your users might introduce security risks from misoperations due to the lack of user access control.

Solution

You can avoid the problems above by allowing different users to manage different services through sub-accounts. By default, a sub-account does not have permissions to use Tencent Cloud services or resources. Therefore, you need to create a policy to grant different permissions to the sub-accounts.

Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access permissions to your Tencent Cloud resources. Using CAM, you can create, manage, and terminate users (groups), and control the specified Tencent Cloud resources that can be used by the specified user 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. For detailed directions, see Policy.

You can skip this section if you do not need to manage permissions to TDSQL-C for MySQL resources for sub-accounts. This will not affect your understanding and use of the other sections of the document.

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：

- We recommend you manage TDSQL-C for MySQL resources and authorize TDSQL-C for MySQL operations through CAM policies. Although the user experience does not change for existing users who are granted permissions by project, we do not recommend you continue to manage resources and authorize operations in a project-based manner.
- Effectiveness conditions cannot be set for TDSQL-C for MySQL for the time being.

Relevant Information	Link
Basic policy structure	Policy Syntax
Operation definition in a policy	TDSQL-C for MySQL Operations
Resource definition in a policy	TDSQL-C for MySQL Resource Path
Resource-Level permission	Resource-Level Permissions Supported by TDSQL-C for MySQL

Authorization Policy Syntax

Last updated：2022-04-01 22:05:41

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. It contains a permission or permission set of multiple 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.

TDSQL-C for MySQL Operations

In a CAM policy statement, you can specify any API operation from any service that supports CAM. APIs prefixed with `cynosdb:` should be used for TDSQL-C for MySQL, such as `cynosdb:DescribeClusters` or `cynosdb:ResetAccountPassword`.

To specify multiple operations in a single statement, separate them with commas as shown below:

```
"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 as shown below:

```
"action":["cynosdb:Describe*"]
```

If you want to specify all operations in TDSQL-C for MySQL, use the `*` wildcard as shown below:

```
"action":["cynosdb:*"]
```

TDSQL-C for MySQL 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/123xxx8`.
- **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 as shown below:

```
"resource":[ "qcs::cynosdb:bj:uin/123xxx8:instance/cynosdbmysql-123abc"]
```

You can also use the `*` wildcard to specify it for all clusters that belong to a specific account as shown below:

```
"resource":[ "qcs::cynosdb:bj:uin/123xxx8:instance/*"]
```

If you want to specify all resources or a specific API operation does not support resource-level permission control, you can use the `*` wildcard in the `resource` element as shown below:

```
"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：2022-04-01 21:59:42

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 that support resource-level permission, 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
TDSQL-C for MySQL 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：
> TDSQL-C for MySQL API operations not listed here do not support resource-level permissions. You can still authorize a user to perform such an API operation, but you must specify `*` as the resource element of the policy statement.

TDSQL-C for MySQL Cluster APIs

API Operation	Resource Path
DescribeBackupConfig	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`
DescribeBackupList	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`
DescribeRollbackTimeRange	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`
DescribeRollbackTimeValidity	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`

API Operation	Resource Path
ModifyBackupConfig	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ActivateCluster	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterDetail	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
IsolateCluster	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyClusterName	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyClusterProject	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
OfflineCluster	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DeleteAccounts	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeAccounts	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyAccountDescription	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ResetAccountPassword	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterInstanceGrps	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ActivateInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeInstanceDetail	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
IsolateInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId

API Operation	Resource Path
UpgradeInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyInstanceName	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
OfflineInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterAddr	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterNetService	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterParams	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterServerInfo	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeErrorLogs	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeMaintainPeriod	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeSlowLogs	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyClusterParam	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyMaintainPeriodConfig	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeDBSecurityGroups	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyDBInstanceSecurityGroups	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
CloseWan	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId

API Operation	Resource Path
OpenWan	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`
DescribeClusters	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`
DescribeInstances	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`
DescribeIsolatedInstances	`qcs::cynosdb:$region:$account:instanceId/*` `qcs::cynosdb:$region:$account:instanceId/$clusterId`

Security Group Management
Configuring Security Group

Last updated：2022-06-15 10:57:56

Overview

A security group 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 one or more TencentDB instances. 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 and are matched with each other within the security group based on rules. For specific rules and limitations, see Security Group Overview.

Note：

- TDSQL-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 does not have active outbound traffic, outbound rules are not applicable to TencentDB.
- TDSQL-C for MySQL security groups support primary and read-only instances.

Configuring Security Groups for TencentDB

Step 1. Create a security group

1. Log in to the CVM console.
2. Select **Security Group** on the left sidebar, select a region, and click **Create**.
3. In the pop-up dialog window, configure the following items, and click **OK**.

- **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 below:

Template	Description	Remarks
Open all ports	All ports are open. May present 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 internet. All ports are opened to the private network.	This template does not take effect for TencentDB.
Custom	You can create a security group and then add custom rules. For detailed directions, see "Step 2. Add a security group rule" below.	-

- **Name**: name of the security group.
- **Project**: select a project for easier management. By default, **DEFAULT PROJECT** is selected.
- **Notes**: a short description of the security group for easier management.

Step 2. Add a security group rule

1. On the Security Group page, click **Modify Rule** in the **Operation** column on the row of the security group for which to configure a rule.
2. On the security group rule page, click **Inbound rule** > **Add Rule**.
3. In the pop-up window, set the rule.

- **Type**: **Custom** is selected by default. You can also choose another system rule template. MySQL(3306) is recommended.
- **Source** or **Target**: traffic source (inbound rules) or target (outbound rules). You need to specify one of the following options:

Source or Target	Description
A single IPv4 address or an IPv4 range	In CIDR notation, such as `203.0.113.0`, `203.0.113.0/24` or `0.0.0.0/0`, where `0.0.0.0/0` indicates all IPv4 addresses will be matched.
A single IPv6 address or an IPv6 range	In CIDR notation, such as `FF05::B5`, `FF05:B5::/60`, `::/0` or `0::0/0`, where `::/0` or `0::0/0` indicates all IPv6 addresses will be matched.
ID of referenced security group. You can reference the ID of: ◦ Current security group ◦ Other security group	◦ To reference the current security group, enter the ID of security group associated with the CVM. ◦ You can also reference another security group in the same region and the same project by entering the security group ID.
Reference an IP address object or IP address group object in a parameter template.	-

- **Protocol Port**: enter the protocol type and port range or reference a protocol/port or protocol/port group in a parameter template.

  Note：
  To connect to a TDSQL-C for MySQL instance, its port must be opened. You can log in to the TDSQL-C console, click a cluster ID to enter the cluster details page, and view the port number.
  The security group rules displayed on the **Security Group** page in the TDSQL-C for MySQL console take effect for private and public (if enabled) network addresses of the TDSQL-C for MySQL instance.

- **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**.

Step 3. Configure a security group

A security group is an instance-level firewall provided by Tencent Cloud for controlling inbound traffic of TencentDB. You can associate a security group with an instance when purchasing it or later in the console.

Note：
Currently, security groups can be configured only for TDSQL-C for MySQL instances in **VPCs**.

1. Log in to the TDSQL-C console and click a cluster ID in the cluster list to enter the cluster management page.
2. On the cluster management page, select the **Security Group** tab and click **Configure Security Group**.
3. In the pop-up window, select the security group to be bound and click **OK**.

Importing Security Group Rules

1. On the Security Group page, click the ID/name of the target security group.
2. On the **Inbound rule** or **Outbound rule** tab, click **Import Rule**.
3. In the pop-up window, select an edited inbound/outbound rule template file and click **Import**.

Note：

As existing rules will be overwritten after importing, we recommend that you export the existing rules before importing new ones.

Cloning Security Groups

1. On the Security Group page, locate the target security group and click **More** > **Clone** in the **Operation** column.
2. In the pop-up window, select the target region and target project, enter the new security group name, and click **OK**. If the new security group needs to be associated with a CVM instance, do so by managing the CVM instances in the security group.

Deleting Security Groups

1. On the Security Group page, locate the security group to be deleted and click **More** > **Delete** in the **Operation** column.
2. Click **OK** in the pop-up window. If the current security group is associated with a CVM instance, it must be disassociated before it can be deleted.

Associating Instance with Security Group

Last updated：2022-06-07 15:57:19

A security group is an instance-level 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.
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.

Note：

- A TDSQL-C for MySQL instance can be associated with up to five security groups.
- You can configure different access addresses with different security groups, which will control only sources that access the current address.

Prerequisites

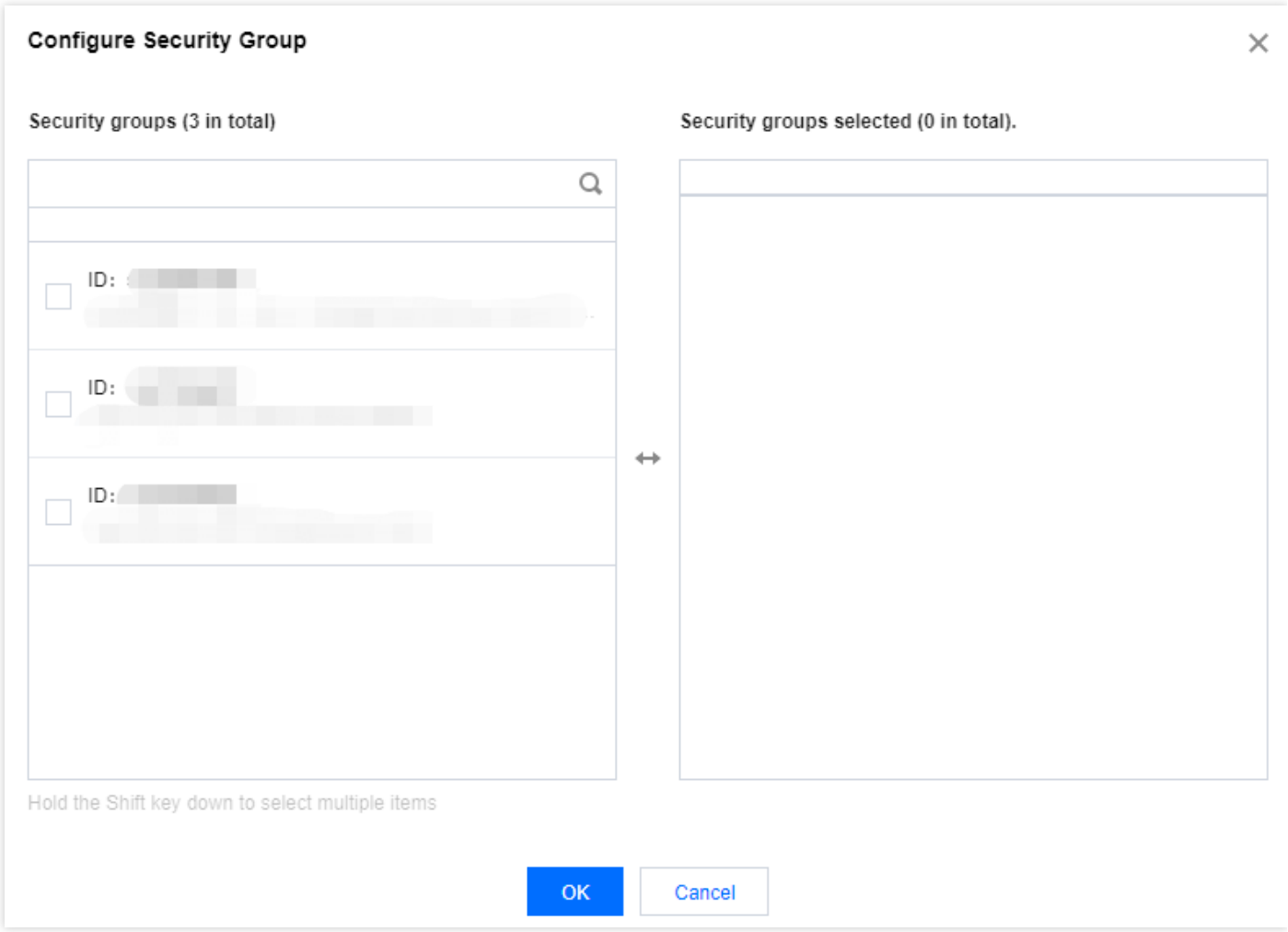
You have created security groups for TencentDB instances in the security group console. For more information, see Configuring Security Group.

Associating Read-Write Address/Read-Only Address with Security Group

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 **Security Group** tab and select the target address. Here:

- Private Read-Write Address for Cluster Connection: read-write access address of the database.
- Private Read-Only Address for Cluster Connection: read-only address of the database.

3. Click **Configure Security Group**. In the pop-up window, select the target security group and click **OK**.



Monitoring and Alarms
Monitoring Feature

Last updated：2022-04-01 21:27:49

To make it easier for you to view and stay up to date with how instances work, TDSQL-C for MySQL provides a wide variety of performance monitoring metrics and convenient monitoring features (custom view, time comparison, merged monitoring metrics, etc). You can log in to the TDSQL-C for MySQL console, click a cluster ID in the cluster list to enter the instance list, click an instance ID, and select **Instance Monitoring** to view them.

> Note：
> If the number of tables in a single instance exceeds one million, database monitoring may be affected. Make sure that the number of tables in a single instance is below one million.

Types of Instances for Monitoring

TDSQL-C for MySQL offers instance-level monitoring information, supports monitoring read-write and read-only instances, and provides each instance with a separate monitoring view for easy query.

Monitoring Granularity

TDSQL-C for MySQL has adopted an adaptive policy for monitoring granularity, which means that you cannot select a monitoring granularity as desired for the time being. The adaptive policy is as follows:

Time Span	Monitoring Granularity	Adaptation Description	Retention Period
(0h, 4h]	5 seconds	The time span is below 4 hours, and the monitoring granularity is 5 seconds	1 day
(4h, 2d]	1 minute	The time span is above 4 hours but below 2 days, and the monitoring granularity is 1 minute	15 days
(2d, 10d]	5 minutes	The time span is above 2 days but below 10 days, and the monitoring granularity is 5 minutes	31 days

Time Span	Monitoring Granularity	Adaptation Description	Retention Period
(10d, 30d]	1 hour	The time span is above 10 days but below 30 days, and the monitoring granularity is 1 hour	62 days

Note：
Currently, you can view the monitoring data of TDSQL-C for MySQL in the last 30 days.

Monitoring Metrics

CM provides the following monitoring metrics for TDSQL-C for MySQL instances in the instance dimension:

Note：
For more information on how to use TencentDB monitoring metrics, see TDSQL-C for MySQL Monitoring Metrics.

Metric Name	Unit
Traffic Received from Client	MB/s
Traffic Sent to Client	MB/s
DELETE Statements	Counts/sec
TPS	Counts/Sec
UPDATE Statements	Counts/sec
INSERT Statements	Counts/sec
Used Memory	MB
Max Connections	-
Cache Hit Ratio	%
Cache Hits	-

Metric Name	Unit
SELECT Statements	Counts/sec
QPS	Counts/sec
CPU Utilization	%
Used Storage Space	GB
Database Connections	-
Cache Hit Ratio (New)	%
Temp Tables Created	-
InnoDB Cache Utilization	%
Slow Queries	-
Running Threads	-
Rollbacks Performed in Storage Engine	-
Internal COMMIT Statements	-
Rows Deleted from InnoDB Tables	-
Rows Inserted to InnoDB Tables	-
Rows Updated in InnoDB Tables	-
Rows Read from InnoDB Tables	-
Redo Log Based Replication Lag of Secondary Instance	ms
Primary-Secondary LSN Difference During Redo Log Based Replication	Bytes
Replication Status of Secondary Instance	0-Yes, 1-No
InnoDB Logical Writes	-
InnoDB Logical Reads	-

Alarm Policies (Cloud Monitor)

Last updated：2022-02-11 16:58:45

This document describes how to create alarm policies and associate alarm objects in the Cloud Monitor console.

Overview

You can create alarm policies to trigger alarms and send alarm notifications when the Tencent Cloud service 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.
You can take appropriate precautionary or remedial measures in a timely manner when the alarm is triggered by changed product status. Therefore, properly created alarm policies can help you improve the robustness and reliability of your applications. For more information on alarms, see Creating Alarm Policy in Cloud Monitor.

To send an alarm for a specific status of a product, 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.

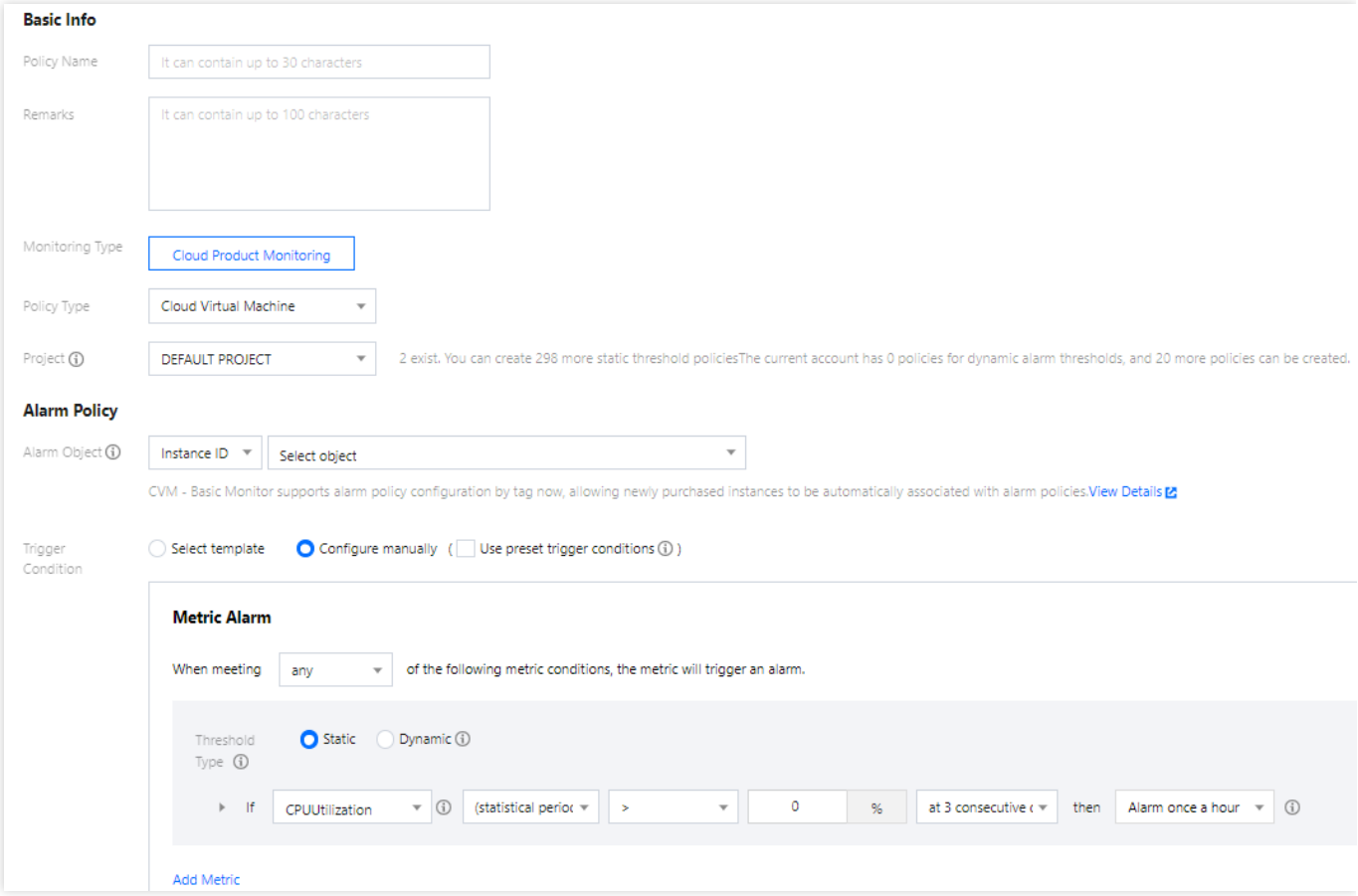
> Note：
> Make sure that you have set the default alarm recipient; otherwise, the default alarm policy of TencentDB won't be able to send notifications.

Directions

Creating alarm policy

1. Log in to the CM console and select **Alarm Configuration** > **Alarm Policy** on the left sidebar.
2. In the alarm policy list, click **Create**.
3. On the **Create Alarm Policy** page, set the policy name, policy type, alarm object, and trigger condition.
   - **Policy Type**: select TencentDB > TDSQL-C > MySQL.
   - **Alarm Object**: the instance to be associated with the policy alarm. You can locate the target instance by selecting the region where it is located or searching for its ID.

- **Trigger Condition**: an alarm trigger is a 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**: you can 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.



4. After confirming that everything is correct, click **Complete**.

Associating alarm objects

After the alarm policy is created, you can associate alarm objects with it. When an alarm object satisfies an alarm trigger condition, an alarm notification will be sent.

1. In the alarm policy list, click the name of an alarm policy to enter the alarm policy management page.
2. Click **Alarm Object** > **Add Object** on the alarm policy management page.
3. In the pop-up window, select the target alarm object and click **OK** to associate it with the alarm policy.

Alarm Notification (DBbrain)

Last updated：2022-07-26 19:56:02

This document describes how to view exception alarm messages from DBbrain in the console.

DBbrain's exception alarm notification service pushes instance exception alarm messages to you in real time, allowing you to conveniently and promptly discover database exceptions and diagnose problems.
All pushed exception alarm messages are displayed in the historical message list, so you can quickly view and locate previously pushed exception diagnosis problems.

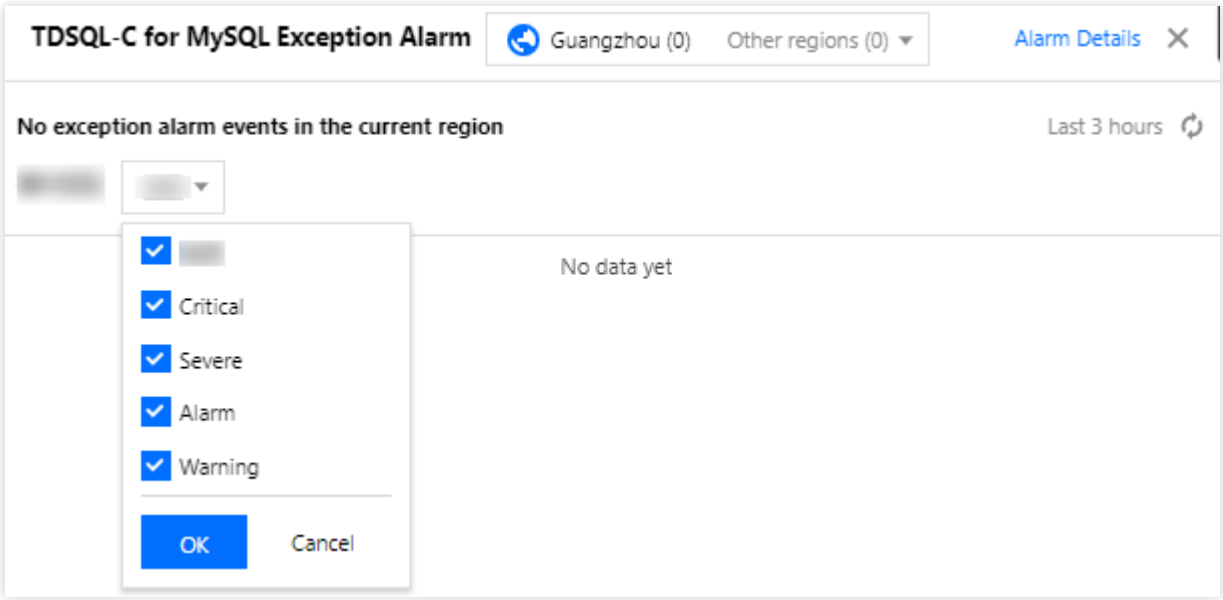
Viewing an Alarm

Option 1

Log in to the TDSQL-C for MySQL console. If an exception diagnosis problem occurs on an instance when you are in the console, a window will pop up in the top-right corner of the console in real time to push the exception alarm message notification, which contains the database instance information such as instance ID, instance name, diagnosis item, and start time, so you can quickly and conveniently stay on top of the running status of the database instance.

- You can click **View Exception Diagnosis Details** in the message notification to view the specific diagnosis details and optimization suggestion for the instance.
- If you check **No alarm again today** in the message notification, when an exception diagnosis problem occurs in a database instance under your account, no exception alarm messages will be pushed to you in a pop-up window.

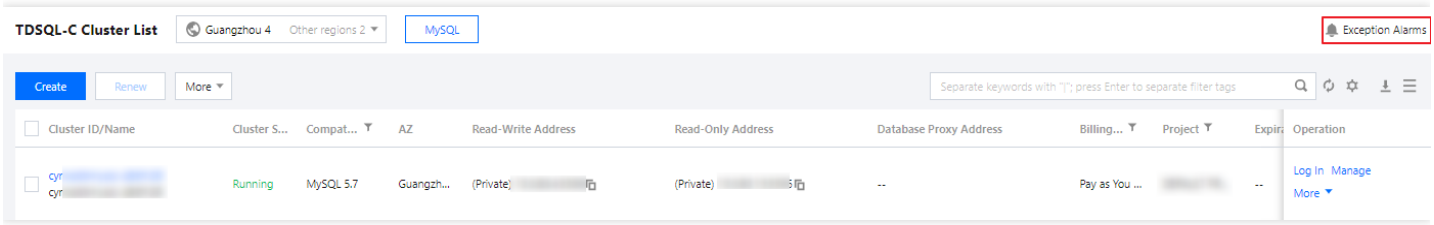
- **Event Level** allows you to filter alarm events by alarm level.



Option 2

Log in to the TDSQL-C for MySQL console, select **Instance List**, **Task List**, **Parameter Template**, or **Recycle Bin** on the left sidebar, and click **Exception Alarm** in the top-right corner to expand the list of historical exception alarm messages. The number of alarms generated in the instances under your account is displayed next to the button.

In the expanded list of historical exception alarm messages, you can view all pushed exception alarm messages. You can view them by region and click a message to view the diagnosis details of the exception alarm event.



Basic SQL Operations

Last updated：2022-02-11 15:51:33

This document describes common SQL commands.

For more information on SQL commands, including command parameters and restrictions, see MySQL 5.7 Reference Manual.

Querying Version

Method 1:

```
MySQL [(none)]> SELECT CYNOS_VERSION();
+-----------------------+
| CYNOS_VERSION() |
+-----------------------+
| 5.7.mysql_cynos.1.3.10 |
+-----------------------+
1 row in set (0.00 sec)
```

Method 2:

```
MySQL [(none)]> SELECT @@CYNOS_VERSION;
+-----------------------+
| @@CYNOS_VERSION |
+-----------------------+
| 5.7.mysql_cynos.1.3.10 |
+-----------------------+
1 row in set (0.00 sec)
```

Method 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	Example
Creates database and specifies character set	`create database db01 DEFAULT CHARACTER SET gbk COLLATE gbk_chinese_ci;`
Drops database	`drop database db01;`

Account Commands

Command	Example
Creates account	`CREATE USER 'username'@'host' IDENTIFIED BY 'password';`
Drops account	`DROP USER 'username'@'host';`
Grants permission	`GRANT SELECT ON db01.* TO 'username'@'host';`
Queries accounts in database	`SELECT user,host,password FROM mysql.user_view;` or `show grants for xxx`
Revokes permission	Revokes all permissions: `REVOKE ALL PRIVILEGES,GRANT OPTION FROM 'username'@'host';` Revokes specified permission: `REVOKE UPDATE ON *.* FROM 'username'@'host';`

Connecting to TDSQL-C for MySQL Through SCF

Last updated：2022-04-24 16:54:10

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.