

TDSQL for MySQL

Getting Started

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

 Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Getting Started

- Creating an Instance

- Initializing Instances

- Connecting to Instances

- Sharded Table Management

Getting Started

Creating an Instance

Last updated : 2021-03-02 17:36:00

This document describes how to create an instance in the TDSQL for MySQL Console.

Directions

1. Log in to the [TDSQL for MySQL Purchase Page](#), select a region, specify the AZs, network type, shard specifications, etc., and click **Buy Now**.
 - For more information, please see [Instance Architecture](#), [Shard Configuration](#), and [Pricing](#).
 - The networks of Tencent Cloud services in different regions cannot communicate with one another.
 - Once selected, a VPC cannot be changed. For more information on VPC operations, see [Managing VPC Instances](#).
2. Verify all the information is correct on the pop-up window and click **Buy Now** to purchase.
3. You will be returned to the TDSQL for MySQL instance list after purchase. Wait for the instance status to be created. When the status changes to **Uninitialized**, you can initialize the instance.

Initializing Instances

Last updated : 2021-04-08 09:49:12

After creating a TDSQL for MySQL instance, you need to initialize it before you can use it.

Directions

1. Log in to the [TDSQL for MySQL console](#), select a region, locate the uninitialized instance in the instance list, and click **More > Initialize** in the **Operation** column.
2. In the pop-up dialog box, select the configuration as needed, and click **OK**.
 - Supported character sets: select the character set supported by MySQL.
 - Case-sensitive table name: set whether the table name of database is case-sensitive.
 - Enable strong sync: enable strong sync to ensure data consistency at the secondary node in case the primary node fails. At least 2 nodes are required for this option to work.

Instance Initialization ✕

Supported character sets: UTF8 LATIN1 GBK UTF8MB4

Database importing may fail if the character set is improperly configured.

Case-sensitive table name:

Enable strong sync: Strong sync (downgradable) Async

If there is only one active node, the cluster automatically downgrade into async mode.

OK Cancel

3. Return to the instance list to locate the instance. When the status changes **Running**, the instance can be connected.

Instance Connection

After initialization, you can access the TDSQL for MySQL instance over both the private and public networks from a Windows or Linux CVM instance. For more information, please see [Connecting to Instances](#).

Connecting to Instances

Last updated : 2021-04-08 09:49:12

Connection Methods

You can connect to a TDSQL for MySQL instance in the following ways:

- **Private network connection:** a CVM instance can be used to connect to the private network address of a TencentDB instance. This method relies on the high-speed private network of Tencent Cloud and features low delay.
- The two instances must be under the same account and in the same [VPC](#) in the same region, or both in the classic network.
- The private network address is provided by TencentDB by default and can be viewed in the instance list or on the instance details page in the [TDSQL for MySQL console](#).

Note :

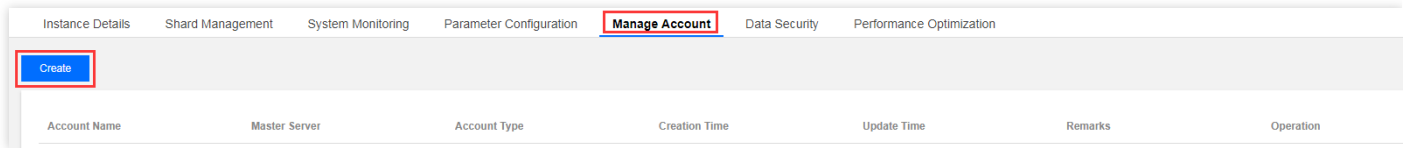
CVM and TencentDB instances in different VPCs (under the same or different accounts in the same or different regions) can be interconnected over private network through [Cloud Connect Network](#).

- **Public network connection:** if you cannot access the private network, you can connect to your TDSQL for MySQL instance at its public network address. The public network address needs to be [manually enabled](#). It can be viewed on the instance details page in the [TDSQL for MySQL console](#) and can be disabled if no longer needed.
- Enabling the public network address will expose your database services to the public network, which may lead to database intrusions or attacks. We recommend that you use the private network to connect to the database.
- Public network connection to TencentDB is suitable for development or auxiliary management of databases but not for business access in the production environment, as potentially uncontrollable factors may lead to unavailability of the public network connection, such as DDoS attacks and bursts of high-traffic access.

Preparations

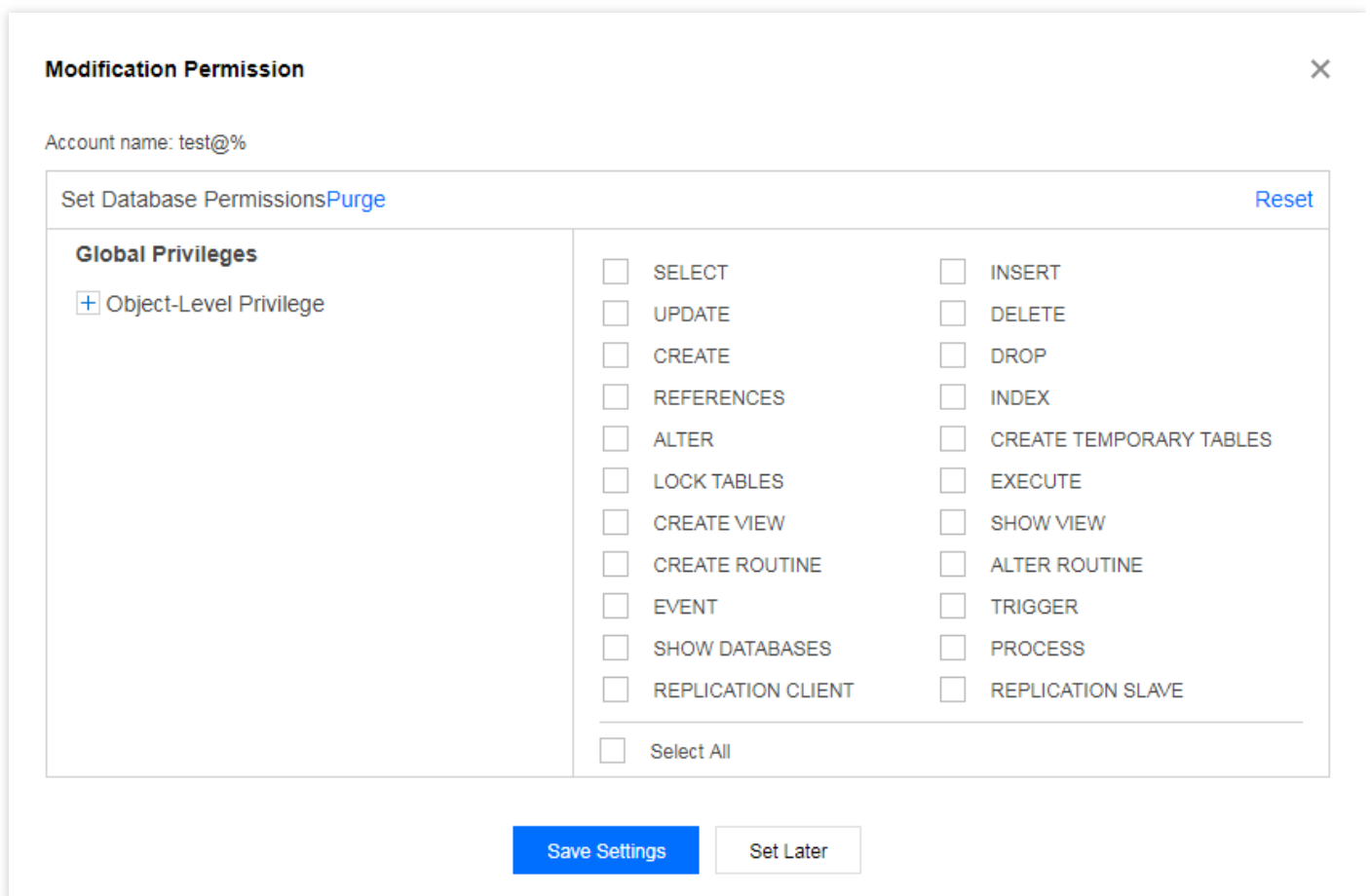
Creating an account

1. Log in to the [TDSQL for MySQL console](#). In the instance list, click an instance ID/name or **Manage** in the **Operation** column to access the instance management page.
2. Select the **Account Management** tab and click **Create**.



3. In the pop-up dialog box, enter the account name, host, password, and remarks. After confirming that everything is correct, click **OK**.
The host name is actually the network egress address. If needed, you can enter % to indicate that this instance is accessible to every IP.
4. In the permission modification dialog box, after assigning permissions based on needs, click **Save Settings** to complete the permission assignment. If you need to set permissions later, click **Set Later**.

The navigation bar on the left provides a graphical interface fully compatible with MySQL management. Permissions can be managed at the column level.



- Return to the account list, click **Modify Permissions** to modify user permissions, click **Clone Account** to completely copy the current account permissions to create a new account. Click **More** to reset the password and delete the account.

帐号名	主机	帐号类型	创建时间	更新时间	备注	操作
hao	%	常规帐号	2020-06-24 16:57:54	2020-06-24 16:57:54		修改权限 克隆帐号 更多 ▾

Enabling the public network address (optional)

- Log in to the [TDSQL for MySQL console](#), click an instance ID/name in the instance list to enter the instance details page, and click **Enable** next to **Public IP** in the **Basic Info** section.

Instance Details	Shard Management	System Monitoring	Parameter Configuration	Manage Account	Data Security	Performance Optimization
Basic Info						
Instance name: international-dcdbt-km62d7gb ✎			Instance ID: dcdbt-km62d7gb 🔑			
Running status: Disabling private network...			Instance type: Logic instance			
Instance version: Standard Edition (1 master-1 slave)			Region: South China (Guangzhou)			
Private address: [REDACTED]			Port: 3306			
Network: Default-VPC			Public address: [REDACTED] Close 🔑			
Project: DEFAULT PROJECT ✎			Character set: UTF8			
Configuration Info						
Database version: MariaDB 10.1.9			Configuration: 4 GB MEM, 20 GB storage space			
Backup and log space: 8 GB (50% of the instance capacity is provided for free)			Used/Total capacity: 0MB / 20GB			
Node count: 2			Creation time: 2019-09-19 19:43:20			
Expiration time: 2019-10-19 19:43:20 Renew			SQL engine layer version: 1.13.21-M-V2R501D002 Shard Details			

- After the successful enablement, you can view the public network address and port number next to **Public IP**. TDSQL for MySQL assigns a unique pair of public IP and port to each instance.

After creating an account and obtaining the public/private network address, you can connect to TDSQL for MySQL through third-party tools and program drivers.

- On Windows, connection methods of command line, client, and JDBC driver are taken as examples.
- On Linux, connection method of command line is taken as an example.

Connecting from a Windows Device

Connecting with Windows command lines

1. Open the Windows command line tool, and enter the following commands under the correct path of mysql.

```
mysql -h public/private IP -P port number -u username -p
Enter password: *****
```

2. If the following information is displayed, the database is connected successfully, and you can operate it.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

Connecting with the Windows client

1. Download a standard SQL client such as MySQL Workbench and SQLyog. Here we take SQLyog as an example.
2. Open SQLyog, select **File > New Connection**, enter host address, port, username and password, and click **Connect**.

- **MySQL Host Address:** enter the public/private IP of the instance obtained earlier.
 - **Username:** enter the account name created earlier.
 - **Password:** enter the password of the account. If you forgot the password, reset it in the [console](#).
 - **Port:** enter the port number of the instance obtained earlier.
3. The interface after successful connection is as shown below. You can operate the database in this interface.

Connecting with the Windows JDBC driver

TDSQL for MySQL supports connecting with program drivers. This document uses Java JDBC Driver for MySQL (Connector/J) as an example.

1. [Download](#) a JDBC jar package from the MySQL official website, and import it to the library referenced in Java.
2. Run the following code to call JDBC:

```
public static final String url = "public/private network address";
public static final String name = "com.mysql.jdbc.Driver"; // Call JDBC
public static final String user = "username";
public static final String password = "password";
//JDBC
Class.forName("com.mysql.jdbc.Driver");
Connection conn=DriverManager.getConnection("url, user, password");
```

```
//
conn.close();
```

3. After the connection is successful, you can perform other operations on the database.

Note :

Because TDSQL for MySQL needs to specify the shardkey when sharding and inserting data, these operations cannot be called with JDBC.

Connecting from a Linux Device

Connecting with Linux command lines

This document takes CentOS 7.2 64-bit on a CVM instance as an example. For more information on CVM instance purchase, please see [Purchasing Channels](#).

1. Log in to the Linux CVM, enter `yum install mysql`, and download and install the MySQL client from Tencent Cloud's image source with CentOS's package management software Yum.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.36.3.el7.x86_64 on an x86_64

UM_213_43_centos login: root
Password:
[root@UM_213_43_centos ~]# yum install mysql
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.52-1.el7 will be installed
--> Processing Dependency: mariadb-libs(x86-64) = 1:5.5.52-1.el7 for package: 1:mariadb-5.5.52-1.el7.x86_64
--> Running transaction check
--> Package mariadb-libs.x86_64 1:5.5.50-1.el7_2 will be updated
--> Package mariadb-libs.x86_64 1:5.5.52-1.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version                                Repository
=====
Installing:
mariadb                                x86_64                              1:5.5.52-1.el7                          os
Updating for dependencies:
mariadb-libs                            x86_64                              1:5.5.52-1.el7                          os

Transaction Summary
=====
Install 1 Package
Upgrade    ( 1 Dependent package)

Total download size: 9.5 M
Is this ok [y/d/N]:
```

2. When `complete` is displayed on the command line, the MySQL client is successfully installed.

3. Enter `mysql -h public/private IP -P port -u username -p` to connect to TDSQL for MySQL. Then, you can perform sharding.

Below uses `show databases;` as an example:

```
[root@UM_3_86_centos ~]# mysql -hcdcdbt ██████████ myqcloud.com -P12 -uexample1 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 493901845
Server version: 10.0.10-proxy Source distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> show databases;
+-----+
| Database           |
+-----+
| SysDB              |
| information_schema |
| mysql              |
| performance_schema |
| test               |
+-----+
5 rows in set (0.01 sec)

MySQL [(none)]> _
```

Sharded Table Management

Last updated : 2021-05-13 15:16:02

This document takes a sharded table as an example to describe some simple database operations in TDSQL for MySQL.

Creating a Table

- For more information about the differences between sharded, non-sharded, and broadcast tables, please see [Overview](#).
- For more information about the restrictions on shardkey, please see [Creating Tables](#).
- To create a sharded table, the shardkey needs to be specified. The sample code is as follows:

```
mysql> create table test1(id int primary key,name varchar(20),addr varchar(20))shardkey=id;
Query OK, 0 rows affected(0.15 sec)
```

Inserting Data

注意：

The shardkey must be included in an `INSERT` statement or else the operation will be denied.

Insert data into the table just created. The sample code is as follows:

```
mysql> insert into test1(id,name) VALUES(1,'test');
Query OK, 1 rows affected(0.08 sec)
mysql> insert into test3(name,addr) values('example','shenzhen');
ERROR 7013 (HY000): Proxy ERROR:get_shardkeys return error
```

Querying Data

注意：

When you query data, we recommend that you include the shardkey in the statement so that the request will be automatically routed to the corresponding shard according to the

distributed routes, achieving the highest efficiency; otherwise, TDSQL will automatically scan the entire table and then aggregate the results at the gateway, which compromises the efficiency.

The sample code for querying data is as follows:

```
mysql> select id from test1 where id=1;
```

Deleting Data

注意：

A `WHERE` clause must be included in a `DELETE` statement, and we recommend including the shardkey in the `WHERE` clause.

The sample code for deleting data is as follows:

```
mysql> delete from test1 where id=1;  
Query OK, 1 row affected (0.02 sec)
```